University "Politehnica" of Bucharest

Faculty of Automatic Control and Computers

Vrije University of Amsterdam

Faculty of Sciences

Multi-path inter-domain routing: The impact on BGP's scalability, stability and resilience to link failures

---

Author:

Adriana Szekeres

Supervisors:

Benno Overeinder
NLnet Labs, Amsterdam

Guillaume Pierre
Dept. of Computer Science,
Vrije University of Amsterdam

August, 2011

Table of Contents

ABSTRACT

Boarder Gateway Protocol (BGP) is a critical part of the Internet, as it is the protocol that keeps the Autonomous Systems (ASes) connected. Despite the fact that it managed to scale to the current Internet's size, it also faces other problems, one of them being transient disconnectivity during convergence time. In the last years, efforts to solve this problem concluded with the proposal of multi-path routing protocols. As their name implies, these protocols are designed to explore more paths than BGP in the attempt to keep the ASes connected in case of link failures.

In this thesis we try to shed more light over the multi-path routing protocols by conducting experiments that show their behavior and impact on BGP. We focused on three multi-path protocols, i.e. R-BGP, YAMR and STAMP, and devised scenarios and experiments to show their impact on BGP's scalability, stability and resilience to link failures. Our results show that R-BGP outperforms the other two methods, being the only one that maintains continuous connectivity during convergence time and at the cost of the smallest number of extra BGP messages.

Keywords multi-path routing protocols, BGP, Internet topology, scalability, stability, resilience to failures

Chapter 1

Introduction

The Internet can be perceived as a network of networks. Each such network, referred to as an Autonomous System (AS), is managed independently from the others and presents a single, clearly defined routing policy to the Internet. ASes in today's Internet disseminate inter-domain routing information (reachability of networks) by the Boarding Gateway Protocol (BGP) [RLH06]. BGP is a path vector protocol, as it maintains path information that gets updated dynamically. Unlike most of the interior routing protocols, which periodically flood the network with all the topology information that they have, BGP sends incremental updates, i.e. only when a currently used path or policy has changed. Therefore, BGP achieves a greater degree of scalability.

The Internet has grown to such an extent that transient failures in backbone networks that previously impacted only a few scientists may now cause great financial loss and impact hundreds of thousands of end users. Being a critical part in Internet, BGP has been subjected to numerous studies that analyze its dynamics. It has been shown that during BGP convergence, triggered by a withdrawal or link failure, BGP faces temporary disconnectivity, even though a policy compliant path to the destination might still exist. The most relevant study in this area has been made by Labovitz and shows that the BGP convergence delay for isolated route withdrawals can be greater than 3 min in 30% of the cases and could be as high as 15 min [LABJ00]. They also found that packet loss rate can increase by 30 times and packet delay by 4 times during recovery. Although this is an old study (from year 2000), we believe that these problems still appear in today's Internet and could be even worse, as no solution has been adopted and the number of ASes has grown more than two times.

To understand the cause for such a high packet delay, consider an AS, AS_A, that learned several paths to the same destination, D, from several different neighbors, see Fig. 1.1 a) (a dashed line means an indirect path). AS_A chose the path through AS_B to forward to the other neighbors (as we explain in chapter 2, BGP allows only one path to be forwarded). When AS_E is disconnected from D, due to link failures, it will send withdrawals to its neighbors. Eventually, AS_A receives a withdrawal of the path to D from AS_B. AS_A removes the path received from AS_B and chooses another path to route packets on. Let this path be the one received from AS_C. After choosing the path from AS_C as the currently routing path, AS_A advertises it to all its neighbors (if the policy allows), Fig. 1.1 b). Recursively, the neighbors receiving this update will make changes

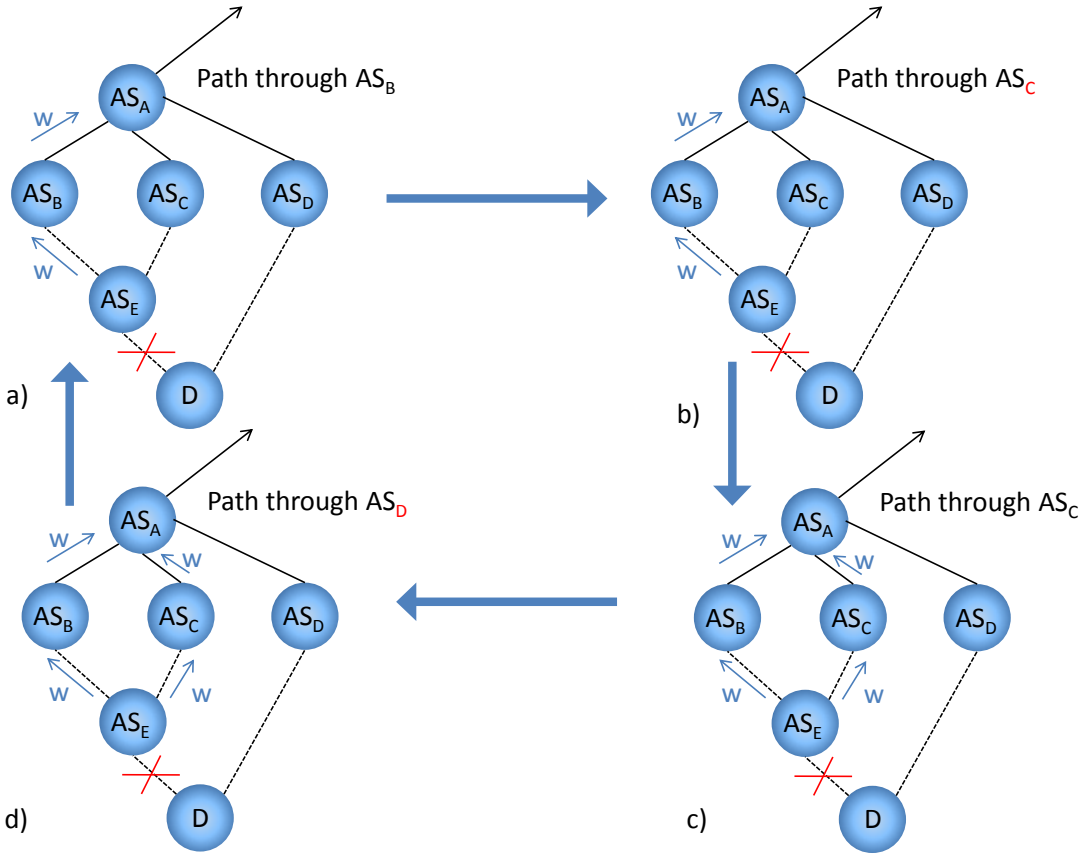

Fig. 1.1: Bad path exploration

in their routing tables and possibly will forward this path to their neighbors, and so on. Now, consider that the path received from AS_C has been affected by the same failure that caused the withdrawal received from AS_B (the path from AS_C to destination D, goes through AS_E). This means that AS_A will eventually receive a withdrawal of the path to D from AS_C also, Fig. 1.1 c). In conclusion, AS_A has chosen a bad alternative to replace the withdawn path and such, delayed the process of routing the packets on a valid path (and sending them into one or multiple loops). This is what could be called a *bad path exploration* after receiving a withdrawal. This situation could have been avoided if AS_A had chosen the path from AS_D from the beginning, Fig. 1.1 d).

Packet dropping happens when an AS doesn't know how to route the packet, i.e. it doesn't currently have any routing paths to the destination. This situation happens even if there exists another path to the destination, but the AS didn't learn about it. This is a consequence to the fact that BGP discovers only a very small fraction of the existing paths between any two given ASes, as only preffered paths are forwarded. Therefore, an AS must wait for an update containing a new path until it is able to route the packets again.

As a solution to the packet delay and temporary disconnectivity problems mentioned

above, in the recent years several papers proposed multi-path inter-domain routing. The basic idea of these proposals is to compute alternative paths, as disjoint as possible from the current path used by the BGP, thus adding a certain grade of flexibility to Internet routing. Therefore, in the case of a withdrawal or link failure, the ASes could still remain connected by immediately switching to an alternate path, without waiting for the announcement of a new path.

Although multi-path inter-domain routing might be the solution to transient disconectivity during BGP convergence, using it to solve this problem could do more damage than expected. Currently, there are over 30,000 ASes in the Internet and their number continues to grow. At this size, BGP faces serious scalability and stability problems. To deal with the scalability problems, various solutions have been proposed. One very effecient solution was the Minimum Route Advertisement Interval (MRAI) (section 9.2.1 of [RLH06]). This mechanism allows a BGP speaker to announce routes about a certain destination (a prefix) to its peers no more frequently than once per MRAI time interval. Another solution was Route Flap Damping (RFD), [VCG98], which, however proved to do more bad than good [MGVK02]. BGP's scalability is still a problem and researchers are still looking for improvements. A very recent proposal is PED (Path Exploration Damping) [HRA10]. Also, to deal with the stability problems, an Internet-Draft has been published [LG07].

The necessity of these *patches* shows the fragility of BGP. Even one slight modification to the protocol might have a critical impact on the whole Internet, as BGP is the one that keeps the ASes connected. As a consequence, any new proposal that introduces modifications to BGP must be carefully studied and tested in an environment as close as possible to the reality. At this point it is not clear how these multi-paths methods will impact BGP, i.e. how they will behave in the real Internet. Therefore, in this project we study the impact of multi-path inter-domain routing methods on the scalability and performance of BGP.

The multi-path inter-domain routing proposals mentioned above, can be classified in two categories:

- Protocols constructed on top of BGP (e.g. MIRO [XR06], R-BGP [KKKM07], STAMP [LGGZ08], YAMR [GDGS10]): these methods introduce slight modifications to BGP and usually compute only one alternate disjoint path (except for YAMR, which, for each link in the primary path, computes an alternate path that doesn't contain that link);
- Protocols which propose an entirely new routing protocol (e.g. Pathlet routing [GGSS09], Path Splicing [MEFV08]): these methods introduce new routing architectures and provide ASes with many path segments that can be combined to form a complete path to a destination.

In this project, we focus on the first category and study the first three (and most recent) proposals from this category, i.e. R-BGP, STAMP, YAMR. We chose not to study MIRO

as, even though it finds alternative paths, it does not achieve the same goal as the others, i.e. ensuring connectivity during the convergence time. It just permits an AS to make requests for alternative paths when it is not satisfied with the ones it already has. Also, we chose not to focus on the methods from the second category because they assume the replacement of BGP and we believe that it is more useful and urgent for now to study the methods that require the least change to BGP.

For each of the three methods, we will highlight:

- The impact on BGP's scalability.

  It is obvious that the proposed methods introduce some overhead to the basic exterior routing protocol. However, we do not know exactly to what extent this will affect its scalability. These experiments will answer the following questions: *How many new updates will be introduced? How much will the routing table size grow?.*

- The impact on BGP's stability.

  The proposed multi-path methods could also have some impact on the convergence time. This is related to the the MRAI Timers and the increase in the number of updated. In this project we will also study this aspect.

- The effectiveness of the method; impact on resilience to failures.

  Some of the proposed methods compute only one alternative path, while others more. We will study what happens when one link fails. We will also study the quality of the alternate paths discovered, i.e. are they the best alternate paths that could have been chosen, in terms of disjointness/length/policy-compliance? In short, these experiments will answer the following question: *To what extent does the method offer resilience to one link failures and what is the quality of the alternative paths it found?.*

All the experiments were conducted using CAIDA topologies collected between 2004 and 2010. Prior to the above mentioned experiments, we conducted a study to characterize the topologies in terms of their path diversity and disjointness. This is important as it will show the connectivity of the ASes which has great impact on the effectiveness of these methods. We cannot keep two ASes connected as long as there is only one path that connects them and some link on it fails.

In conclusion, our main research question is:

> *What will happen if we deploy multi-path routing methods to the current Internet; how will they affect BGP's scalability, stability and resilience to link failures?*

As the proposed methods are relatively new, to our knowledge, no previous study that tries to answer our question has been made. We believe this comparative study is important as it will show the strengths and weaknesses of the methods, when deployed on the same (as close as possible to the current Internet's architecture) topologies. It will give insight into their effectiveness against multiple types of failures and their impact on AS-level routing.

Chapter 2

Background

This chapter provides the necessary background to understand the work done in this thesis and interpret the obtained results. We first describe the Border Gateway Protocol, as all the methods we chose to analyze are built on top of it. We also highlight its problems, as they make for an important part of our study, i.e. how do multi-path methods affect BGP's problems. After describing BGP, we present the multi-path routing methods, particularly the three methods that we implemented and studied. Then we describe BGP simulation techniques and, in particular the one that we use to run our experiments. At the end we briefly present the topologies used in such BGP studies.

2.1 Border Gateway Protocol

Today's Internet is basically a large computer network that links together smaller networks to each other. Each such smaller network, called an Autonomous System (AS), is, logically, a connected group of one or more Internet Protocol (IP) routing prefixes[1] under the control of one or more network operators that presents a single, clearly defined routing policy to the Internet [HB96]. Physically, the core of an AS is a connected group of routers that exchange routing information through the so called Interior Gateway Protocols (IGPs). Even when multiple IGPs and metrics are used inside an AS, the administration of the AS appears to other ASes to have a single coherent interior routing plan and presents a consistent picture of what networks are reachable through it [HB96].

The various IGPs, such as Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and Intermediate System-to-Intermediate System (IS-IS), are distributed routing protocols that basically come in two flavours: *distance vector* and *link-state* routing protocols. The term *distance vector* refers to the fact that each router computes a vector containing the distance and direction[2] to each destination prefix, and periodically advertises it to its neighbors. In contrast to distance vector protocols in which nodes share their routing tables (the distance vectors), in link-state protocols nodes share only connectivity information, *link*

[1] A routing prefix is basically a prefix of a normal IP address, used to uniquely identify a certain network in the Internet.

[2] Direction is simply the next hop to the destination.

state, that help them to create a connectivity graph, based on which they independently compute the best paths to each prefix destination.

Distance vector IGPs work well at the size of an AS. However, in the current Internet there are over 35000 ASes. Running a distance vector IGP at the scale of the Internet would not be possible as the routing tables and the number of messages required by the protocol would explode. Although link-state algorithms have traditionally provided better routing scalability, which allows them to be used in bigger and more complex topologies, they still should be restricted to interior routing. Link-state protocols by themselves cannot provide a global connectivity solution required for Internet inter-domain routing. In very large networks and in case of route oscillation caused by link instabilities, link-state retransmission and recomputation will become too large for any single router to handle. Therefore, other routing protocols have been devised to run only between ASes and not between all the routers in the Internet. These protocols are called Exterior Gateway Protocols (EGPs). Running such a protocol at the AS-level is possible as an AS doesn't care what happens inside of another AS, it doesn't need to know about such specific routes, i.e. internal AS routes. It only needs to know the path to the AS containing the destination prefix.

The Exterior Gateway Protocol uses Autonomous System Numbers (ASNs). An ASN is represented by a unique 2-byte or, recently introduced due to ASN pool exhaustion, 4-byte identifier associated with an AS. ASNs are assigned in blocks by the Internet Assigned Numbers Authority (IANA), [Aut11], to Regional Internet Registries (RIRs). Recursively, RIRs assign ASN from their IANA allocated blocks within its designated area. More information on ASNs and their allocation can be found in [Hus06]. The EGP currently deployed in the Internet and used by all ASes is Boarder Gateway Protocol (BGP). BGP is a *path vector* routing protocol, a class of *distance vector* protocol discussed above. The difference is that in *path vector* protocols the node maintains a vector of the entire paths to each prefix destinations, not only the distances.

The components with which BGP works are: address prefixes and ASNs. Every prefix has an originating AS, known as the *Origin AS* from which reachability for the prefix is propagated across the inter-domain space. When an AS receives a path to a prefix, it stores it in its routing table and, if it is the best path that it received so far, it signs the update by prepending its ASN to the path and forwards it to its neighbors. An example of how network reachability is propagated into the intra-domain space is shown in Fig. 2.1, where AS1 is the Origin AS for the address prefix 186.0.2.0/24. After receiving the update from AS1, AS2 stores path [1] to prefix 186.0.2.0/24 into its routing table, prepends its ASN to this path and sends an update message to AS5. AS3 does the same as AS2 but sends the update message to AS4. As can be noted in the picture, AS5 receives two BGP advertisements for this prefix. One has the AS path [4, 3, 1], and the other has the AS path [2, 1]. AS5 will choose the best of the two routes, let that be route [2, 1], and advertises it to AS6. The left-most number in the AS path list is the ASN of the adjacent AS from which the address prefix advertisement was received. The sequence of numbers indicates the sequence of ASs though which this

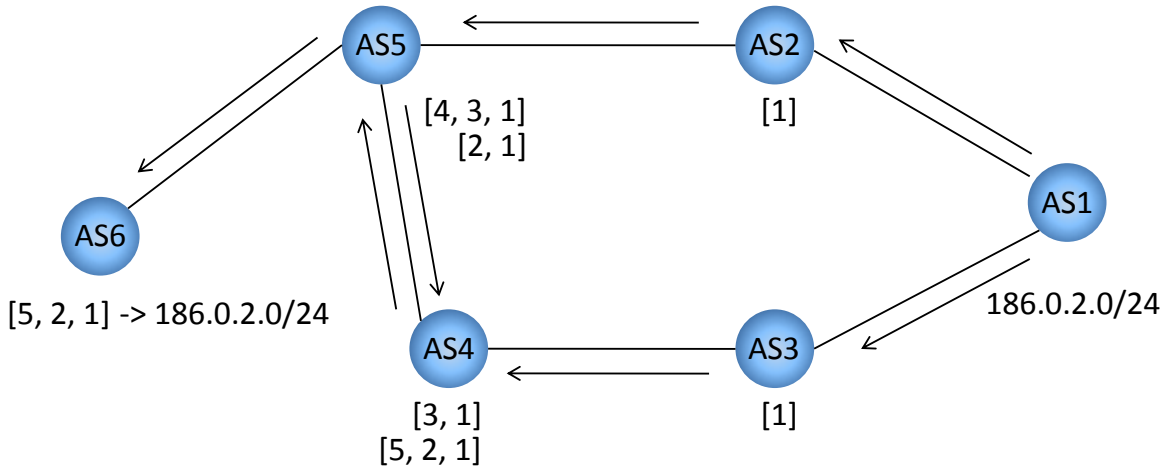

Fig. 2.1: BGP network rechability propagation

update was propagated. The right-most, or final ASN, is the AS number of the Origin AS. A withdrawal of a route is propagated in a similar manner. When an AS receives a route withdrawal, it will remove it from it tables, choose another route if it has one, and forward the new route or also a withdrawal to its neighbors.

The key feature of BGP is that it allows each AS to choose its own administrative policy in selecting and propagating routes to its neighbors. Routes are selected and propagated by taking into consideration the **relations** (commercial agreements) with the neighbors and the other **policies** of the AS, such as routes learned from a customer are preferred over those learned from provider or peer. The relations between two neighbor ASes can be classified into: *peer-peer*, *consumer-provider* and *sibling-sibling*. A customer pays its provider to transit its traffic to the rest of the Internet. However, a customer does not transit traffic between two of its providers. An AS transits the traffic from its peers to all its customers free of charge. A pair of siblings offer connectivity information to each other. How these relations affect the propagation of routes can be summarized into Fig. 2.2. For example, if an AS learned a route from a peering AS, it will not export it to any of its providers but it will export it to its customers.

Advertize routes to		provider	customer	peer	sibling
Learned from	provider	no	yes	no	yes
	customer	yes	yes	yes	yes
	peer	no	yes	no	yes
	sibling	yes	yes	yes	yes
Own routes		yes	yes	yes	yes

Fig. 2.2: BGP routes propagation rules

In her work on inferring AS relations, Gao demonstrated an interesting theorem, which

shows how the routes found by BGP look like [Gao01]. The theorem states that if every AS followed the above route propagation rules, then all the routes found by BGP would be *valley-free*. The valley-free property states that once a path traversed a provider-to-customer or peer-to-peer edge in the AS connectivity graph, that path cannot traverse a customer-to-provider or peer-to-peer edge. Therefore, a valley-free path can be described by one of the following patterns: **uphill**: a sequence of edges that are either customer-to-provider or sibling-to-sibling edges, **downhill**: a sequence of edges that are either provider-to-customer or sibling-to-sibling edges, **an uphill path followed by a downhill path**, **an uphill path followed by a peer-to-peer edge**, **a peer-to-peer edge followed by a downhill path** or **an uphill path followed by a peer-to-peer edge, which is followed by a downhill path**.

Many following BGP related proposals, like the multi-path routing methods we are studying, assume that routes are *valley-free*. However, an AS might choose, for example, to also export all its routes to a certain provider, although such cases rarely happen.

2.1.1 BGP's scalability and stability problems

Although BGP is a very simple protocol, running it at a such large scale raises scalability and stability problems. BGP's scalability is affected by the number of updates that are sent between ASes and the number of entries in the routing tables, whereas BGP's stability is affected by path exploration during BGP's convergence and also by some anomalies (e.g. routing loops that appear due to misconfigurations or bugs). In this project we will focus only on path exploration as this is directly affected by the multi-path methods.

BGP's scalability problem has always been carefully studied and many solutions have been proposed to alleviate it. A solution for the increasing number in updates has been the introduction of Minimum Route Advertisement Interval (MRAI timers). An AS can receive multiple routes to the same prefix, from different neighbors, at different times. Therefore, it can choose to propagate suboptimal routes before receiving the best one. It is shown that if the AS was permitted to forward the updates right away, the number of updates and the convergence time would increase considerably [Pre01]. Therefore, MRAI timers control how often an AS is allowed to send BGP updates and, thus, delay the decision of choosing which path to forward. MRAI timers have also a great impact on BGP's stability, as the AS will have more chances to forward only the optimal route, and such it won't create any other unnecessary *waves*. The causes of BGP's routing table growth have been studied, [BGT04], and some solutions have been adopted, such as Classless Inter-Domain Routing (CIDR) and route aggregation/summarization. However, routing tables continue to grow exponentially as more and more ASes choose to have more than one provider (multi-homing).

BGP's stability is also a serious and constantly monitored problem. An Internet-draft was submitted in 2007 analyzing BGP's stability problems and proposing several so-

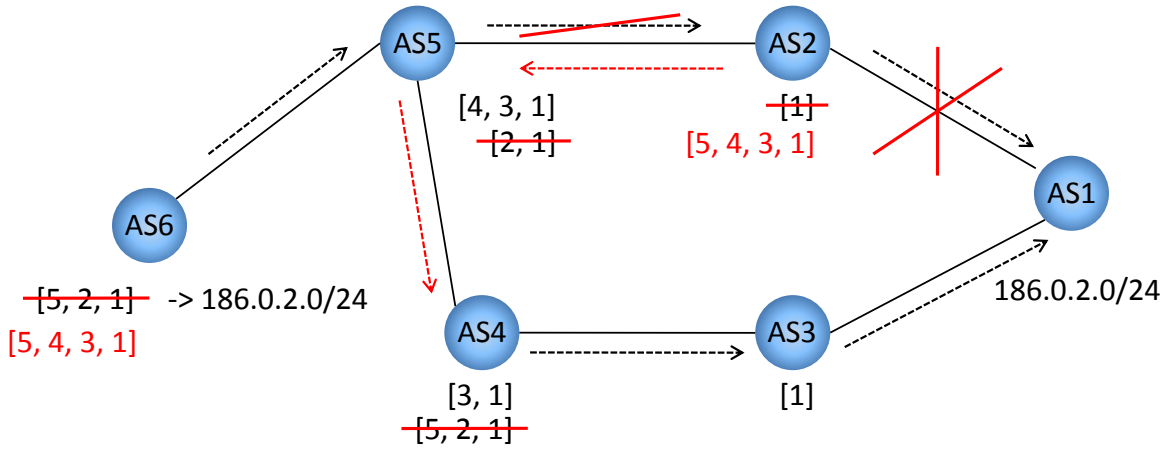

Fig. 2.3: BGP transient disconnectivity problem

lutions [LG07]. The newest proposed solution to BGP's stability and also scalability problem is Path Exploration Damping (PED), which delays update messages which would announce a route with a same-length or longer AS Path than the previously announced route for the same prefix for a period of time, called the Path Exploration Damping Interval (PEDI) [HRA10]. As the MRAI timers, this technique would also suppress unnecessary BGP updates.

2.2 Multi-path routing methods

In the last few years, multi-path routing mechanisms have been proposed as a solution to the disconnectivity problem that appears during BGP's convergence triggered by a link failure/withdrawal. Ideally, in case of link failure BGP would immediately redirect traffic on paths that do not contain that failed link. However, in reality, it can happen that the traffic cannot be redirected due to a shortage of alternate paths so packets are dropped until an alternate path is advertised and BGP re-converges. To understand how such a situation can appear, consider the example in Fig. 2.3, where the dashed arrows show how the packets flow into this topology. Also, the routing tables for each AS are shown. For example, AS5 has two paths to prefix 186.0.2.0/24 into its routing table, but it currently uses path [2, 1].

When the link between AS1 and AS2 fails, as marked in the figure with a red cross, AS2 will send a withdrawal to AS5, informing it that the path previously advertised ([2, 1]) is no more available. AS5 will remove path [2, 1] from its routing table and will switch to the alternate path [4, 3, 1]. AS5 will then send an update to AS2 and AS6 announcing this new path that it is using. AS2 will start using this new learned path and we can say that BGP has re-converged. In BGP, an AS is permitted to forward only the path that it currently uses to route the packets, thus, at the beginning, AS2 knew no alternate path to AS1. Therefore, starting form the moment when the link between

AS1 and AS2 failed to the moment when AS2 received the new path, AS2 dropped all the self-initiated messages destined to AS1 as well as those routed through it to AS1.

The disconnectivity problem shown in Fig. 2.3 could have been avoided if AS2 had known about the alternate path to AS1 from the beginning. Therefore, the ambitious goal that the multi-path methods are trying to achieve is: any AS, A, should be able to continuously have reachability information about an advertised prefix as long as there is a path in the AS connectivity graph between A and the AS that advertised the prefix. In other words, if an AS has a policy compliant path both before an event and after BGP has re-converged, then it should not be disconnected at any time during the convergence time.

2.2.1 Resilient BGP (R-BGP)

The idea behind R-BGP is to use *failover* paths [KKKM07]. A failover path is computed and forwarded before a link fails, instead of waiting for a link failure in order to begin the path exploration, as it is the case in the current BGP. Although this idea is simple in principle, the solution should consider BGP's scalability and stability problems presented above. For example, a very simple solution would be to let ASes advertise not only the best paths but all the other paths they learned, as failover paths. Of course this will solve the problem, but it will greatly affect BGP's scalability. Therefore, R-BGP tries to solve the following challenges: select and disseminate failover paths that constitute continuous reachability information without much overhead; prevent the formation of transient loops during convergence; determine when BGP has re-converged to stop using failover paths.

To solve the first challenge, R-BGP selects and advertises only a few failover paths, i.e. one path per prefix per neighbor, the same as BGP. The failover paths are strategically disseminated, meaning that an AS advertises a failoverpath only to the neighbour through which it is routing. For example, in Fig. 2.1, AS5 would have advertised path [5, 4, 3, 1] as a failover path to AS2, as its current best path is [2, 1], thus AS2 being the AS through which it is routing and AS4 would have advertised path [4, 5, 2, 1] as a failover path to AS3. Also, in R-BGP the failover paths are chosen to be as disjoint as possible from the current path used. These paths will intuitively protect the most against link failures. The authors of R-BGP claim that it is not necessary for each AS to know a failover path for every link that can fail and, in fact, it suffices if each AS is responsible only for the link immediately downstream[3] of it.

Transient loops can appear during BGP's convergence, whether R-BGP is used or not. To solve this problem, R-BGP uses *Root Cause Information* (RCI). RCI has been previously proposed to reduce the convergence time and number of messages, by modifying the BGP update packet to contain information about the failed link. R-BGP uses RCI to prevent the formation of transient loops during convergence time. However, using

[3]On which it is routing

RCI to eliminate affected paths before receiving a proper withdrawal could generate another problem: an AS could be left without a path even though it will be advertised a new one. To solve this problem, R-BGP lets the ASes use the old primary paths when left without any alternate path.

The last challenge that must be solved is to know when an AS should stop using the old primary path or the failover path. To solve this issue, R-BGP uses the following mechanism: an AS stops forwarding the traffic along old primary paths or failover paths when explicit withdrawals have been received from all neighbors; an AS delays sending a withdrawal to a neighbor until it is sure it will not offer this neighbor a valley-free path at convergence time; an AS knows it will not offer a valley-free path to a non-customer once it has heard withdrawals or advertisements from all customers, additionally it knows it will not offer a valley-free path to a customer once it has heard withdrawals or non valley-free paths from all neighbors.

2.2.2 SelecTive Announcement Multi-Process protocol (STAMP)

The idea behind STAMP is to run in each AS several BGP instances that will discover *complementary* paths [LGGZ08]. Two paths are complementary if they are not affected by the same set of network events. For two paths to be complementary it is sufficient that they satisfy the following property: *node disjointness*, i.e. the two paths do not contain the same AS, except for the source and destination. For example, in Fig. 2.4 paths [5, 2, 1] and [5, 4, 3, 2, 1] are complementary. Requesting full node disjointness might limit the BGP process in choosing and disseminating paths. However, the authors of STAMP claim that full node disjointness is not necessary for the paths to be complementary, not affected by the same network event. Assuming the valley-free property, the paths should ensure node disjointness only for the downhill portion. This assumption is verified by proving the following lemma: a route withdrawal event in the uphill portion of an AS path to a destination does not produce transient routing loops or failures during BGP convergence.

In [LGGZ08], the authors describe STAMP for two BGP processes, red and blue, that run in parallel. The red process accepts only those paths received from red processes running on its neighbors (red paths) while the blue process accepts paths only from the blue processes running on its neighbors (blue paths). STAMP's goal is to ensure that the red and blue paths are downhill node disjoint. To achieve this goal, STAMP selectively announces standard BGP discovered paths, thus controlling their dissemination. There are three rules an AS must follow to propagate the paths:

- if the Origin AS is multi-homed, it selects a subset of its providers to which it advertises its prefixes only through the red process while to the rest it advertises its prefixes only through the blue process; if the Origin AS is single-homed, this

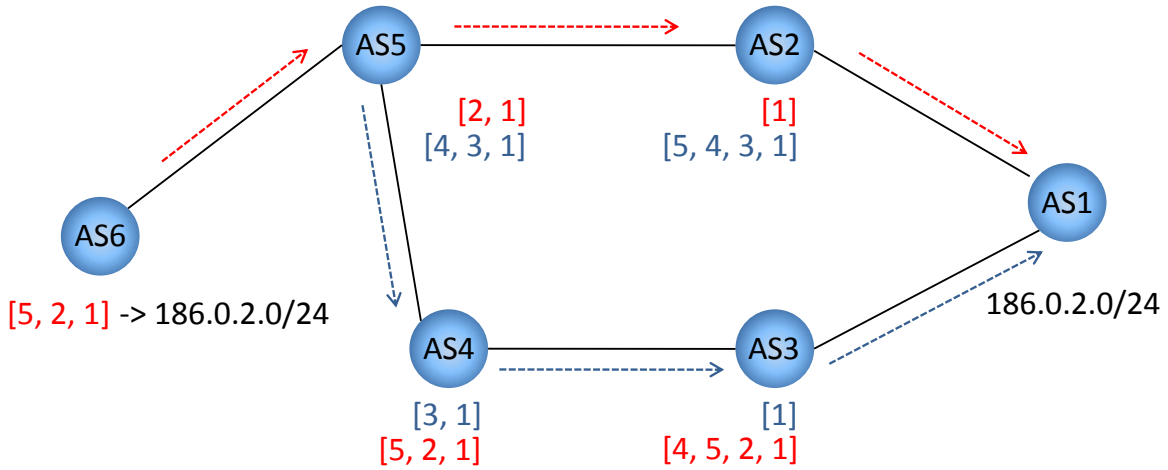

Fig. 2.4: STAMP path dissemination

split is performed at its first direct/indirect provider that is multi-homed. This ensures that red and blue paths are as downhill node disjoint as possible.

- an AS that is not the Origin AS and neither an AS at which the splitting must be performed, must announce either red or blue paths to its providers. Otherwise, the red and blue paths would not be node disjoint as it will share this AS.
- path announcements to peers and customers are not selective. In other words, an AS will announce its best red path as well as its best blue path to its customers and peers.

To see how STAMP works, consider the example in Fig. 2.4, in which the left AS of an edge is the provider for the other end of the edge (e.g. AS2 and AS3 are providers for AS1, AS5 is the provider for AS2 and AS4 and so on). AS1 announces prefix 186.0.2.0/24. Being a multi-homed Origin AS, AS1 announces a red path to AS2 and a blue path to AS3. Having only one color paths, AS2 and AS3 have no other choice than to preserve their color and forward the paths to their providers. AS5 will receive paths from both blue and red process, thus it will have to choose a color and forward it to its providers (in this case, only AS6). Let us assume the best path for it is the red one, [2, 1]. Next, AS5 will send a red and blue update, if necessary, to all its customers. AS4 will receive the red path [5, 2, 1] while AS2 will receive the blue path [5, 4, 3, 1]. As a last step, the red process on AS4 will send the red path [4, 5, 2 , 1] to its customer, AS3.

2.2.3 Yet Another Multi-path Routing protocol (YAMR)

YAMR is the most recent from the three multi-path routing protocols that we chose to analyze. The idea of YAMR is to try to protect the primary path by advertising additional paths that avoid the links contained in the primary path. Each such alternative

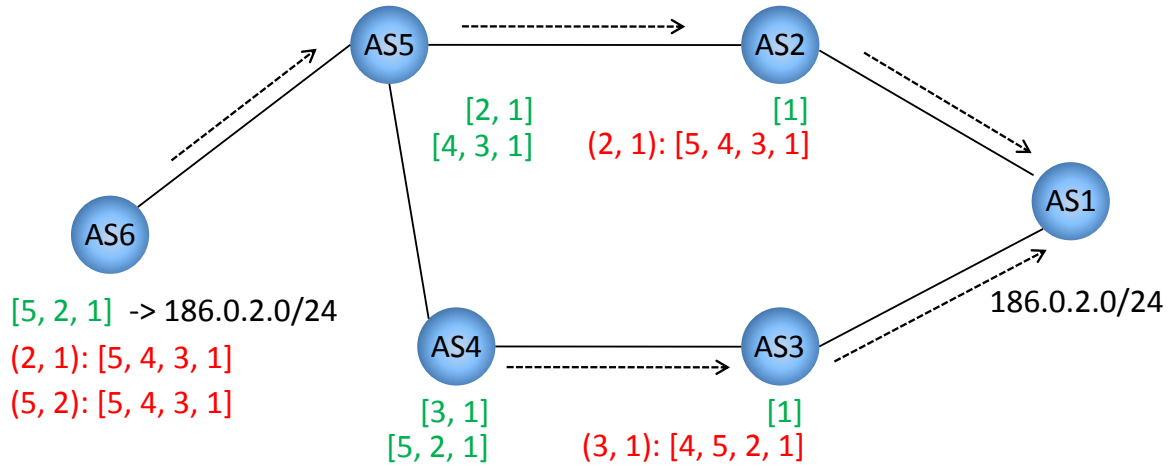

Fig. 2.5: YAMR path dissemination

path is identified by a label, corresponding to the link that the path avoids. Therefore, the forwarding table of the AS will contain the primary path and, for each link in the primary path, one additional path that avoids that link. The protocol is given in Algorithm. 1, where A refers to the AS on which the protocol runs, U_p is the set of primary paths received from neighbours, p_p is the primary path, p_L is a L-labeled path (a path that does not contain label/link L), U_p^L is the set of primary paths that do not contain label/link L, U_L is the set of L-labeled paths received from neighbors and $best_A$ is a function selecting the best from a set of paths, according to AS A's policies.

Protocol 1 YAMR path selecting procedure

```
/* Select the primary path */
p_p ← best_A(U_p)
for link L in p_p do
    /* Select the L labeled path */
    p_L ← best_A(U_p^L ⋃ U_L)
end for
```

To see how the algorithm works, consider the example in Fig. 2.5, in which, as stated before, the left AS of an edge is the provider for the other end of the edge (e.g. AS2 and AS3 are providers for AS1, AS5 is the provider for AS2 and AS4 and so on). When AS5 receives the paths from AS4 and AS2, it selects path [2, 1] as the primary path and then it chooses path [4, 3, 1] as the labeled path for both labels (2, 1) and (5, 2). Note that AS2 can't use path [5, 4, 3, 1] for label (5, 2), because path [5, 4, 3, 1] would be in fact path [2, 5, 4, 3, 1] which indeed contains link (5, 2).

2.3 Testing and analyzing changes to BGP

BGP is in principle a very simple protocol; fundamentally, BGP is a peer-to-peer protocol in which its peers *gossip* about network reachability to keep their routing tables up to date. Its complexity lies in the fact that it is run at a very high scale and that is when the problems start to show. Testing and analyzing a proposed feature for BGP should be performed at the same scale at which the current BGP is working. Ideally, a new BGP feature should be tested on the real Internet. However, this is not an option because it requires the implementation of the enhanced BGP to be imposed on each router — as BGP is a critical component in the Internet, this will never be done until it is certain that the new feature will work as it is supposed to and will not have any bad consequences. Therefore, other methods to test BGP and BGP changes must be used. The first one is using an analytical model of BGP. While analytical methods can provide useful insight into the protocol operation by showing, for example, bounds on number of messages and convergence delay, they are simplistic and do not capture the complexity and flexibility of the protocol. For example, the analytical method has been succesfully used to prove that BGP does not converge under certain policy configurations in [GW99]. The second method, which is also used to perform our analysis, is simulation and is described in the following section.

2.3.1 Simulation of BGP

Modeling BGP

Simulation has long been the preferred method in studying BGP's behaviour. Although simulating a small network is easy, building a simulator that can simulate the whole Internet is not a trivial task. In the last decade, researchers focused on ways of building efficient large scale simulators. In [HK03] the authors present the first steps towards building a large scale BGP simulation environment. In [DR06] the authors present BGP++, a BGP simulator that takes into consideration the abstraction-scalability tradeoff: a higher layer of abstraction, i.e. a less detailed protocol, makes for a more scalable simulator. However, ignoring important details of the protocol could affect the quality of the simulations, therefore the model of BGP used should be carefully designed.

Internet topologies

Besides the level of abstraction used to model the protocol, the accuracy of the results is also influenced by the topologies used by the simulator. Note that we use the term *topology* to refer to the AS connectivity graph together with the relations between ASes. Intensive research has been done additionally in obtaining accurate Internet topologies. There are mainly two approaches to achieving this goal: building a topology generator or inferring a topology from real BGP data (BGP updates or routing tables).

A topology generator has the advantage of permitting the customization of the topology (for example, it can generate a topology in which there are no tier-2 ASes or in which all ASes are multi-homed). Building a topology generator requires an extensive study of the real Internet topology characteristics as the generated graph should exhibit specific Internet characteristics. Most of the previously proposed topology generators do not annotate the connectivity graph with AS relations, however, recently, policy-aware topology generators have been proposed [EKD08, HFKC08].

The second method to obtain an Internet topology is by inferring it from real BGP updates and routing tables. Real BGP data is collected at several public sources, such as Route Views, [oO11], RIPE Routing Service, [Ser11], and CAIDA, [CAI11a], by using BGP monitors. A BGP monitor is an AS that does not announce prefixes or forwards routes, it just records the routes it receives. Using BGP monitors is a passive method that offers limited experimental setup, therefore, BGP beacons have been introduced [MBGR03]. A BGP beacon is a well known and documented prefix that can be injected into the Internet. The advantage is that it permits data analysis when the input is known.

Chapter 3

Approach and techniques

This chapter describes the methodology we used to answer our research questions. We start by analyzing how thoroughly the evaluation of the proposed multi-path methods have been done, as this will give insight into the methodology that was used to evaluate these proposals. Then we will present our approach to evaluate these methods, the criteria and the metrics we used in order to answer our research questions. At the end we will briefly present the tools we used and some notes on the implementation.

3.1 Current evaluation of multi-path routing

The multi-path routing proposals that we study are quite new (starting from 2007), and have not yet been evaluated very thoroughly. In this section we will describe and analyze, in turn, how the testing and evaluation have been done for each of the three multi-path algorithms that we chose to analyze in this thesis. For each of the methods we will first describe the experimental environment and then we will analyze what did the experiments want to measure, i.e. the criteria used to evaluate the methods, and what was the metodology used to carry on these measurements.

3.1.1 R-BGP

To evaluate R-BGP, the authors used their own BGP simulator, which permitted the simulation of a 24,142-ASes connectivity graph. Their simulator implements the basic functions of BGP, i.e. sending and receiving update (announcement/withdrawal) messages and full BGP decision process, and detailed message timing, including MRAI timers. The AS connectivity graph was generated from BGP updates recorded at Route Views, [oO11], and the AS relationships were inferred using the algorithm in [DKF+07].

All experiments were performed on three variants of R-BGP that differ only by which failover path is elected to be advertised: Most-Disjoint Failover Path — the AS picks the most disjoint path from its primary path to advertise as the failover path; Most Disjoint Policy Compliant Failover Path — same as the previous variant but, in addition, the failover path should be policy compliant; Second Most Preferred Failover Path — the

AS advertises its second best path as the failover path. The criteria used to evaluate R-BGP were:

- scalability: This measures the overhead introduced by the method and shows the impact on BGP's scalability. The experiments were conducted as follows. For each dual-homed AS in the topology an experiment was conducted in which one of its links was withdrawn and then the number of messages sent on each link was computed. These experiments might give some insight into what happens during a withdrawal, however it might be useful to also see what happens during a prefix advertisment, as we won't make use of RCI. Also, it should be useful to study how many messages are received by an AS, and not only on one link as these messages are propagated also inside the AS and we suspect that a very large number of message will be send towards the Tier-1 ASes. In conclusion, the metric used was: *the average number of messages sent on each link during convergence time triggered by a link failure at a dual-homed AS.*

- stability: The authors did an experiment to measure the convergence time, computed as the interval between the moment of the failure, to the moment of the last update received at an AS. The same scenario has been used, for each of the dual-homed ASes, one of its links was withdrawn and then the convergenge time was computed. In conclusion, the metric used was: *the convergence time triggered by a link failure at a dual-homed AS*

- resilience to link failures: These experiments show how effective the method really is, compared to the standard BGP. For R-BGP, the authors studied mainly two scenarios, one for *edge links*[1] and one for *core links*[2]. In the first scenario, for each of the dual-homed ASes, it is run a simulation in which one of its links is withdrawn. The results are analyzed to see how many of the ASes that know a path to the dual-homed AS after BGP has re-converged have experienced transient disconnectivity during the convergence time. This is a reasonable scenario that also gives insight into how effective is the multi-homing technique. In the second scenario, a simulation is run in which a core link is withdrawn. The results are analyzed to see how many AS pairs that were connected before the failure by a path containing that core link and are also connected after BGP re-converged, have experienced transient disconnectivity during convergence time. Additionally, simultaneous link failures have been studied: failure of both primary and failover paths (the first failed link is chosen as in the first scenario and the second is chosen randomly from the failover paths used to complement the primary path); changing failover path during failure. In conclusion, the common metric used in all the above scenarios was: *the fraction of ASes that experience transient disconnectivity* and was applied for several different scenarios.

[1] A link that connects a stub AS to the Internet
[2] A link between two non-stub ASes

In conclusion, R-BGP evaluation was done following this criteria: scalability, stability and resilience to link failures. In all the experiments, the comparison was done only between BGP and variants of R-BGP.

3.1.2 STAMP

To evaluate STAMP, the authors also used real BGP updates collected from Route Views, [oO11], to generate the connectivity graph, however they used an older algorithm to infer the AS relations [Gao01]. They also used their own event driven simulator with which they simulated about 26,000 ASes. The processing and transmissions delays are modeled by a random value between 10ms and 20ms. Also, the MRAI timer is per peer and is equal to 30 seconds multiplied by a random value between 0.75 and 1.

Besides standard BGP and different heuristics applied to STAMP, STAMP's performance is also evaluated against R-BGP. The criteria used to evaluate STAMP are:

- resilience to link failures: To see how effective the method is, the authors used several metrics. First, specific only to STAMP, they computed *the probabilities that ASes have both a red and a blue path.* Then, similar to the methodology used to evaluate R-BGP, they used the metric *the fraction of ASes that experience transient problems* in three scenarios: single link failure, in which a link of a multi-homed AS was withdrawn; multiple link failures, in which two links are withdrawn. Two cases are considered: first, the two links are connected to the same multi-homed AS, and second, the links are not connected to the same multi-homed AS, but the second link is connected to an indirect provider; single node (AS) failure, in which all the links attached to an AS are withdrawn.
- incremental development: Deploying an enhanced BGP at all ASes at the same time might be very difficult to achieve, therefore, the proposed method should be *compatible* with the current BGP, i.e. even if only a fraction of the ASes run the modified version of BGP, everything must still function accordingly. The following scenario was simulated to prove that STAMP can be incrementaly deployed and also to give insight into the performance of STAMP when incrementally deploying it: deploy STAMP only at Tier-1 ASes. The metric used to measure STAMP's performance in this scenario was *the fraction of ASes to which each Tier-1 AS has two downhill node disjoint paths*

In conclusion, to evaluate STAMP, the authors used the following criteria: resilience to failures and incremental deployment. The impact on BGP's scalability and stability was not analyzed. However, STAMP was also evaluated against previous methods, specifically R-BGP.

3.1.3 YAMR

To evaluate YAMR, another self-implemented event-driven simulator was used. The simulator supported the important features of BGP, like MRAI timers (with average value of 30 seconds), router processing delay and message propagation delay. They generated annotated topologies of sizes from 500 to 5,000 ASes using [DKVR09].

All the experiments were performed on standard BGP, HBGP[3], YPC and YAMR. Also, for all the experiments the following scenario was used: a multi-homed stub AS announces a prefix; after the network converges, a link connected to that AS is withdrawn and the network is left to re-converge. The same scenario is played for each of the multi-homed stub ASes and each of their links. The following criteria was used to evaluate YAMR:

- scalability: The authors plotted a CDF showing the number of messages following a link event. They also plotted a graph which shows how the number of messages varies with the size of the network.
- stability: To measure the impact on stability, the authors used the same metric, i.e. *the convergence time triggered by a link failure*, however they didn't consider only the dual-homed edged ASes, but the multi-homed edged ASes.
- resilience to link failures: The metric used to study the effectiveness of YAMR is the same as the one used in the previous methods, i.e. *the fraction of ASes that experience transient problems during convergence.*

In conclusion, to evaluate YAMR, the authors used the following criteria: impact on scalability, stability and resilience to failures. However, their experiments were done on very small, self-generated topologies.

3.2 Our approach

In the previous section we described the methodology that was used to evaluate the multi-path routing proposals. Resilience to failure has been studied for each of the three proposals, however, comparison with other methods has been carried out only in one of the proposals, i.e. STAMP. We will perform a thorough comparison between all three methods.

Although the impact on scalability has been studied in two out of three methods, i.e. R-BGP and YAMR, it wasn't studied for all *interesting* scenarios. The scenario in which the impact on scalability was studied was the one in which a link connected to a dual/multi-homed was withdrawn, thus the number of messages were computed

[3]Hiding mechanisms applied to standard BGP

only during convergence time, triggered by a withdrawal, when all the paths have been previously advertised, which constitutes an advantage. However, another *interesting* scenario is when a prefix is advertised. In this scenario the protocols won't use helpful features as in the withdrawal scenario, i.e. the RCI in R-BGP and hiding techniques (at least not to the same extent) in YAMR, therefore we expect the two methods to introduce more overhead.

Impact on stability has also been studied in two out of three methods, i.e. R-BGP and YAMR, and the metric used was the convergence time. However, this too has been measured only for the scenario for which the impact on scalability has been studied.

To perform our experiments we used already annotated Internet topologies from CAIDA for years 2004 to 2010 [CAI11b]. The graphs were annotated using the inference algorithm from [DKF+07]. As BGP simulator we used BGPSim [Woj08], a high-scale BGP simulator, capable of simulate up to 60,000 ASes. We will describe the experimental setup in more details in the next section.

The criteria, metrics and scenarios we used to analyze the three multi-path methods are:

- **scalability**: We studied the impact on BGP's scalability both for the scenario in which a link is withdrawn as well as the scenario in which a prefix is advertised. The metrics we used are: *the number of BGP update messages sent during convergence on each link* and *the number of messages per AS*. It is important to also see the total number of messages per AS as the messages are propagated also inside the AS, i.e. from an edge router to another.
- **stability**: To study the impact on BGP's stability we used the same metric that was used for R-BGP and YAMR, i.e. *the convergence time*, however, we computed it for both the scenarios mentioned above.
- **resilience to link failures**: To give insight into how the methods really work, we used the following scenario: advertise a prefix and let the protocol converge. After this first scenario, we measured *the node-disjointness of the paths from the primary path at each AS* and *the number of alternative paths found by the methods at each AS*. Even though these metrics are not decisive in showing the effectiveness the method, it shows the differences between them. For example, R-BGP will find paths for a lot less ASes than STAMP, but at important ASes. To study the effectiveness of the methods, we used the same scenarios as in R-BGP, for *edge links*.

3.3 Tools used and implementation details

To evaluate the multi-path protocols that we described above we used BGPsim [Woj08] and Internet topologies from CAIDA [CAI11b].

3.3.1 BGPsim

BGPsim is a highly scalable BGP simulator, designed to run on the DAS-3/DAS-4 clusters, [Ams11a, Ams11b], using 32 to 79 computing nodes. BGPsim can simulate tens of thousands of ASes (we used it to simulate up to 33508 ASes), as it uses a high level of AS abstraction. The simulator has been validated by comparing its results with real data collected from real beacons [MBGR03].

In BGPsim an AS is modeled as a single BGP speaking router which has a forwarding table that stores the routes used for packet forwarding, and a table in which it stores all the received paths, which are not necessarily used for packet forwarding. BGPsim also implements the MRAI timers, the per-neighbour variant. It also implements AS policies and exporting rules. For simplification, we considered that each AS uses the same policies and follows the exporting rules described in a previous chapter.

BGPsim is implemented in Java and is structured in several Java packages, all having the prefix *nl.nlnetlabs.bgpsym01.*, which, for simplicity, will be omitted from the packages' names in the next paragraphs. BGPsim has been mainly built as a proof of concept, and such, it lacks some attributes that every simulator should have, the most important one being the possibility of being extended. Therefore, to implement the multi-path protocols in BGPsim we had to heavily modify some important BGPsim packages, the most important being shown in Fig. 3.1: *cache*, *route* and *route.output*.

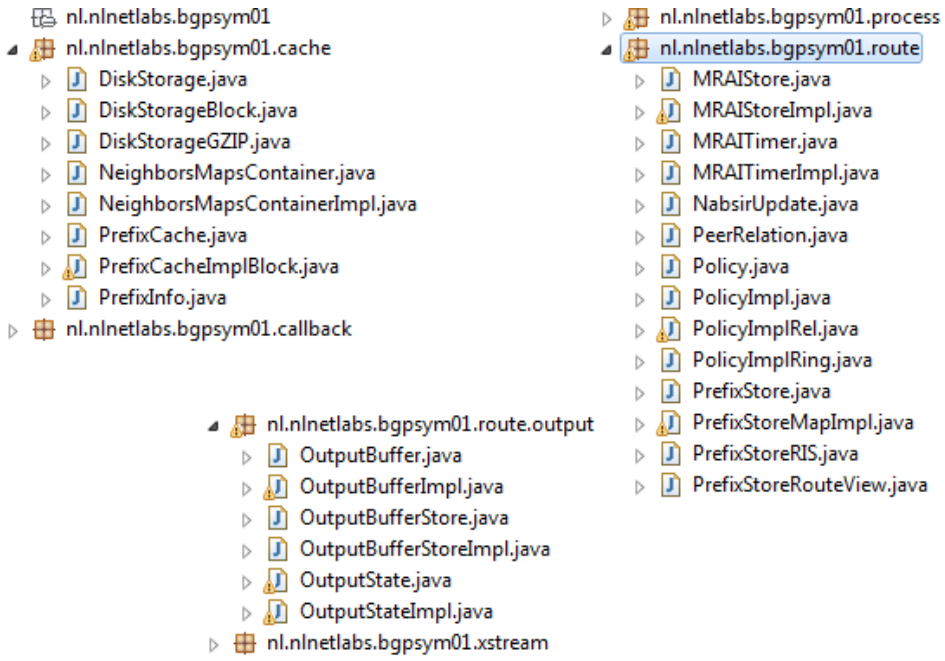

Fig. 3.1: BGPsim important packages

The main function of the *cache* package is to provide classes which store the routing tables. The most important classes in this package are *PrefixInfo* and *PrefixCacheIm-*

plBlock. PrefixInfo stores the routes received from neighbours, for a certain prefix. PrefixCacheImplBlock basically maps a prefix to a PrefixInfo. Therefore, when receiving an update for a certain prefix, class PrefixCacheImplBlock is used to retrieve the PrefixInfo for that prefix and consequently all the routes received for that particular prefix.

Package *route* mainly deals with everything that has to do with routing decisions. The most important class in this package is *PrefixStoreMapImpl*, which implements two intuitive methods, *prefixRemove* and *prefixReceived*. Therefore, PrefixStoreMapImpl is used to decide what happens when an update is received, i.e. if the preferred route changed, what paths must be announced to/withdrawn from neighbors, etc.

If in package *route* the routing decision are taken, it is in package *route.output* where these decisions are applied. Basically there is an announcements buffer and a withdrawals buffer in which the announcements/withdrawals that must be sent to neighbours are stored by the *route* package and actually processed by the *route.output* package. All the classes in this package are important but special attention should be given to class *OutputStateImpl* as it treats the package deferring mechanisms, induced by the MRAI timers. Basically this class stores the actual state of the forwarding table, i.e. which routes have actually been sent to the neighbours. Therefore, the forwarding table in the *route* package might not be up to date, but class *OutputStateImpl* deals with this situation by knowing which paths have been actually sent to corresponding neighbours.

To implement the multi-path protocols we extensively modified all the three packages described above. For each of the three methods we needed specific forwarding tables (e.g. for R-BGP we also needed an entry for the failover path, for STAMP we needed two entries for the two preferred paths, one for each of the two BGP processes, but with one being more preferred than the other to be sent to the providers, and for YAMR we needed additional entries for each link in the preferred path) and specific routing decisions.

3.3.2 CAIDA topologies

CAIDA topologies are basically a snapshot of the Internet, derived from real BGP updates, collected at BGP monitors. A BGP monitor is a passive BGP speaking device, i.e. it only listens to updates but never sends any. From the data collected at several BGP monitors, the Internet graph can be derived. CAIDA graphs are derived from RouteViews, [oO11], BGP table snapshots taken at 8-hour intervals over a 5-day period. However, having just the graph is not enough for applying the AS policies and export routes. Therefore, the next step is to apply an algorithm for inferring the relations between ASes. To infer the relations between ASes, CAIDA used the algorithm proposed in [DKF+07]. Therefore, the general procedure for creating a file in the CAIDA dataset is as follows:

- Extract all AS links from RouteViews snapshots.

- Infer customer-provider relationships, and annotate AS links.
- Infer peer-to-peer relationships, and annotate AS links, possibly overriding customer-provider relationships inferred in step 2.
- Heuristically fix suspicious looking inferred relationships (e.g., a low-degree AS acting as provider to a high-degree AS).
- Infer sibling ASes (that is, ASes belonging to the same organization) from WHOIS, and annotate AS links, possibly overriding previous relationship annotations.

It is important to note that the inferred topologies are not exactly the same as in reality. A truly accurate picture of the Internet topology would require collection of data from every AS, while CAIDA's inferred topologies are limited to the measurement points publicly available at Route Views. Also, the AS relations inferring algorithm is not perfect as it applies heuristics to guess what is the relations between the ASes.

Chapter 4

Evaluation

This chapter presents the scenarios that we used to evaluate the multi-path routing protocols, as well as the results that we obtained. First, we describe the experimental setup, second, we evaluate the impact on BGP's scalability, third, we evaluate to what extent the methods achieve their goal, i.e. the impact on BGP's resilience to link failures, and finally, we evaluate the impact on BGP's scalability.

4.1 Experimental setup

In this section we study the characteristics of the topologies we used. As we mentioned before, we used CAIDA topologies to perform our experiments. We chose CAIDA topologies as they are inferred from real BGP updates, which is very important for our evaluation. We also chose to perform our experiments on topologies from various years to better observe the impact on scalability. As the Internet follows a certain trend in the evolution of its topology (e.g. more ASes appear, more and more ASes go multi-homed, some ASes grow to a superior tier, etc.), using topologies from various years will give insight into what may happen in the future.

To describe the topologies that we used, we first counted the number of ASes that form the topologies from 2004 to 2010 (see Fig. 4.1). It can be noted that the Internet grows consistently from one year to another; it almost doubled during the last 7 years.

As the number of multi-paths that can form on a given topology mostly depend on the degree of multi-homing (they also depend on the relations between ASes and the export policies), we also counted the number of the edged (which have no customers) dual-homed ASes (see Fig. 4.1). We are particularly interested in the edged dual-homed ASes as they are involved in most of our tests, as it will be seen in the scenarios we used.

Topology	2004	2005	2006	2007	2008	2009	2010
# of ASes	16874	18740	21202	24013	26960	30610	33508
# of dual-homed ASes	7283	7952	9017	9800	10627	12075	13028

Fig. 4.1: Number of ASes in the inferred topologies

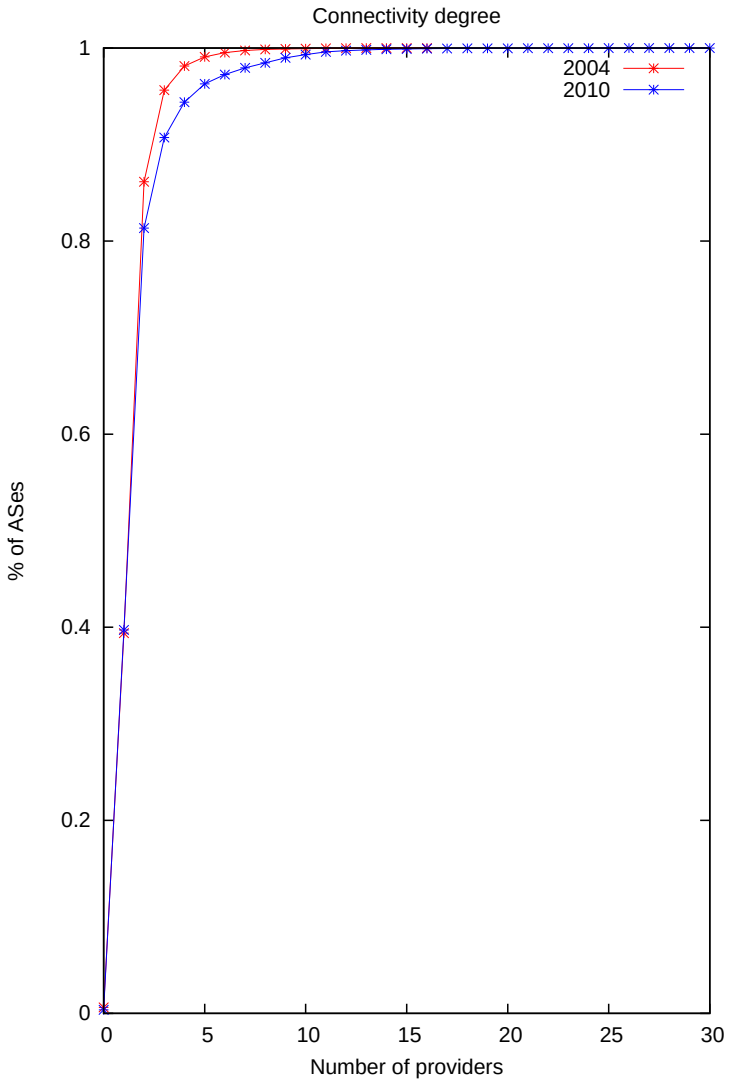

Fig. 4.2: Connectivity degree

To get a full image on the degree of connectivity of the Internet, the graph in Fig. 4.2 plots the cumulative distribution function (CDF) showing the number of providers per AS. To keep the graph comprehensible, we plotted the CDFs for only two topologies, the one from year 2004 and the one from year 2010, respectively. However, it is sufficient to observe that more and more ASes choose to have multiple providers. This will definitely make the Internet more connected and robust, however it will not guarantee to solve the transient disconnectivity problem presented at the beginning at the thesis, although it will give a certain guarantee that an alternate path will be found at the end of the convergence time.

Another interesting aspect to study about the topologies is the disjointness of the paths discovered by BGP, as well as the multi-path protocols. Fig. 4.3 shows the CDF of the

maximum node disjointness of the paths found by BGP and the multi-path protocols (it basically show the path diversity). By node disjointness between two paths, P1 and P2, we refer to the number of ASes that appear in P1 but do not appear in P2. The maximum node disjointness is in fact the maximum node disjointness between the preferred path and all the other alternate paths found by the protocols. Again, we plotted the graph for two topologies, the one from 2004 (in the left of the figure) and the one from 2010 (in the right of the figure). As it was exepcted, YAMR found alternate paths at almost every AS, many of them being 3 node disjoint. STAMP also found alternate paths at almost every AS but most of them are 2 node disjoint. Recall that STAMP only tries to find downhill node disjoint paths, as the authors proved that this is a sufficient requirement to ensure continuous connectivity during the convergence time triggered by a link failure. The differences between BGP and R-BGP are very small. Recall that R-BGP only advertises strategically chosen alternative paths.

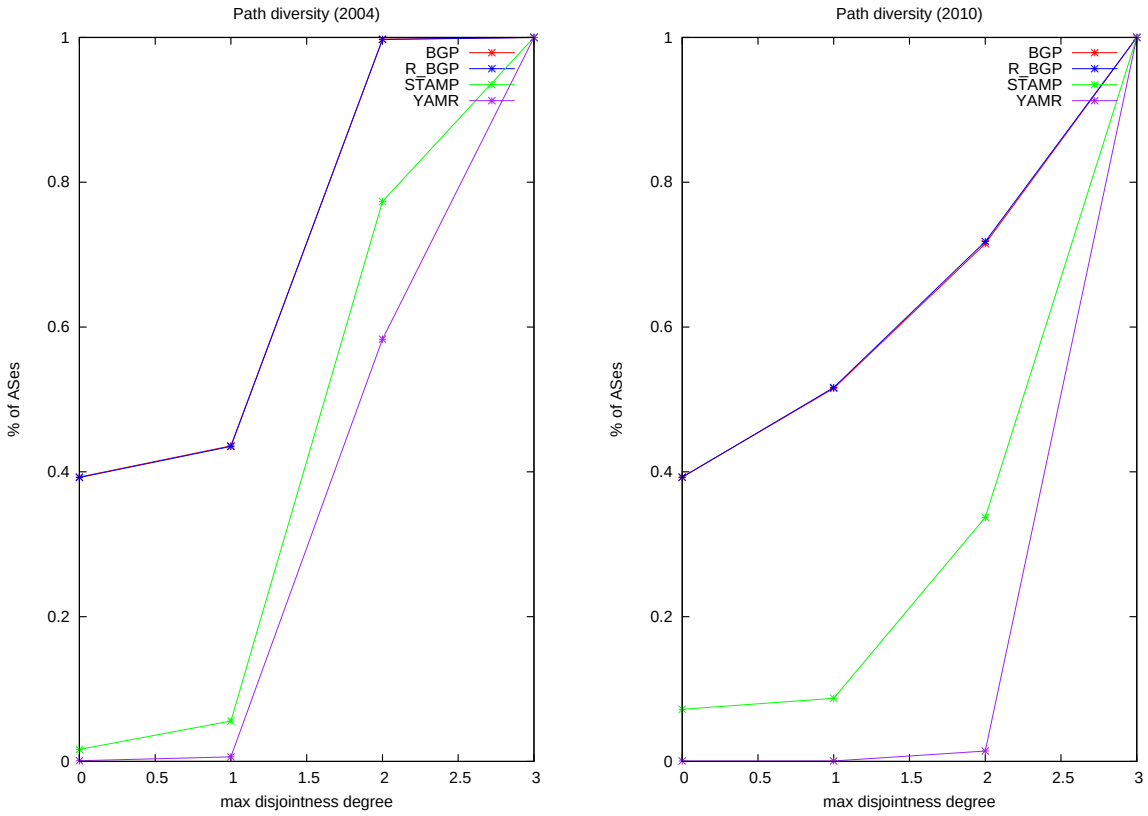

Fig. 4.3: Path diversity

4.2 Impact on BGP's scalability

To study the impact on BGP's scalability we used the following scenario: we let a dual-homed edged AS announce a prefix and then count the number of messages that were sent in the network. We chose the edged dual-homed ASes as they constitute the majority of the ASes in a topology (as it can be noted in Fig. 4.1), therefore being responsible for the majority of the events happening in the topology. Ideally we should have studied each possible event (i.e. announcement/withdrawal at a core AS, announcement/withdrawal at an edged AS, etc.) but time didn't permit so we chose to focus on the most probable events. We repeated the experiment for 100 edged dual-homed ASes. The results are plotted in Fig. 4.4.

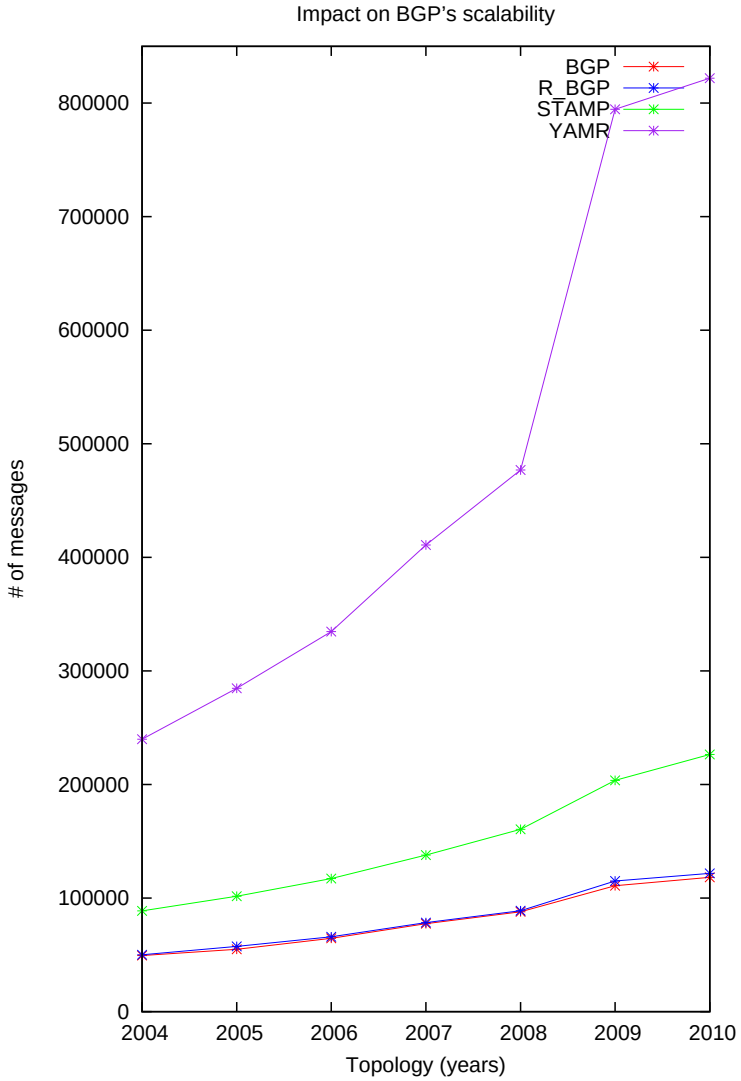

Fig. 4.4: Impact on BGP's scalability

As expected, the YAMR protocol generates the most number of messages as it sends a message for the preferred route, as well as messages for each link in the preferred route for which it knows an alternative path that avoids it. Also, during the convergence time, ASes frequently change their preferred route. With YAMR, the ASes that change their preferred route, must also change the alternative paths, generating consistently more withdrawals and announcements that BGP. However, YAMR's propose some *hiding techniques* to alleviate this problem, which, in their tests, seem to be very efficient, but they will certainly increase the complexity of the protocol.

R_BGP generates the least number of messages. This is explained by the fact that R_BGP does not try to find an alternative path for each AS in the topology, instead it tries to protect the paths to each destination, by puting alternative/failover paths in strategic points. In the next section we will see how this strategy performs.

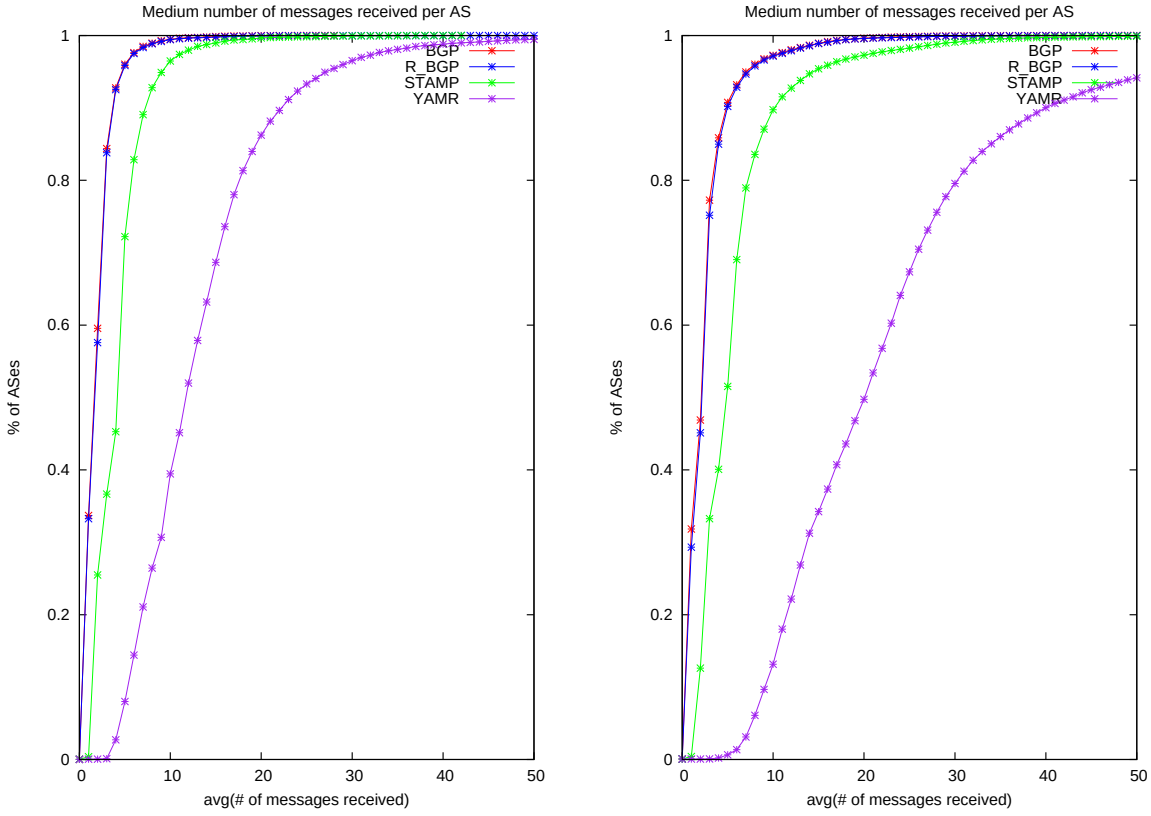

Fig. 4.5: Impact on BGP's scalability

In the middle is STAMP. It generates almost double the number of messages generated by BGP. This result clearly shows the idea behind STAMP, i.e. using two instances of BGP. The number of messages is under double the number of messages generated by BGP because only one instance of BGP running on the AS sends announcements to providers, when something changes, whereas both instances send announcements to the

customers and peers.

Fig. 4.4 gives a global image of what happens during the convergence time after a prefix is introduced into the network. Fig. 4.5 rather shows what happens locally at each AS, i.e. the number of messages it receives. We consider this to be equally important to study as it will show wether the current deployment of an AS will handle the increase in the number of messages it receives. In the left we plotted the CDF of the average (measured from the 100 experiments that we did at 100 different edged dual-homed ASes) number of messages received per AS for the 2004 topology and in the right it is the same graph, but for the 2010 topology. The increase of in the number of messages from the 2004 topology to the 2010 topology is obvious for each of the protocols. For example, if in the 2004 topology 80% of ASes received less than 20 messages for YAMR, while in 2010 only 40% of ASes received under 20 messages.

4.3 Impact on BGP's resilience to failures

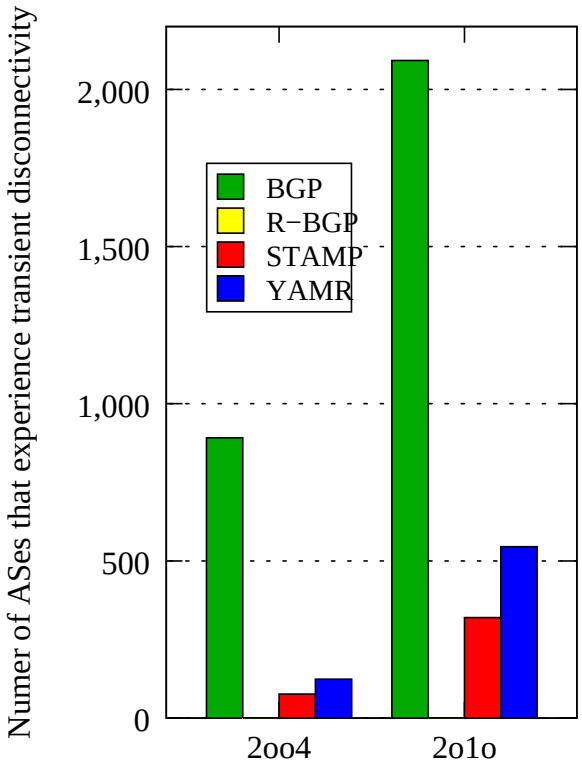

Fig. 4.6: Impact on BGP's resilience to one link failure

In this section we present the results that show to what extent the proposed multi-path protocols achieve their goal, i.e. remove the transient disconnectivity problem.

The scenario that we used in order to measure the resilience to failures is the following: let a dual-homed edged AS advertise a prefix, wait until the network has converged, fail one of its provider links and count the number of ASes that experience transient disconnectivity. We consider that an AS experience transient disconnectivity when it remains with no routes or, in the case of R-BGP, as the ASes continue to use the primary path, the AS remains with no alternate paths and also the primary path is not valid (there is no valid fail-over path at any of the ASes along the primary path - this means that even if the AS uses the old primary path, its messages will still be dropped).

We ran the scenario described above for 500 dual-homed edged ASes. However, in the majority of the cases, BGP performed well (with 0 ASes experiencing transient

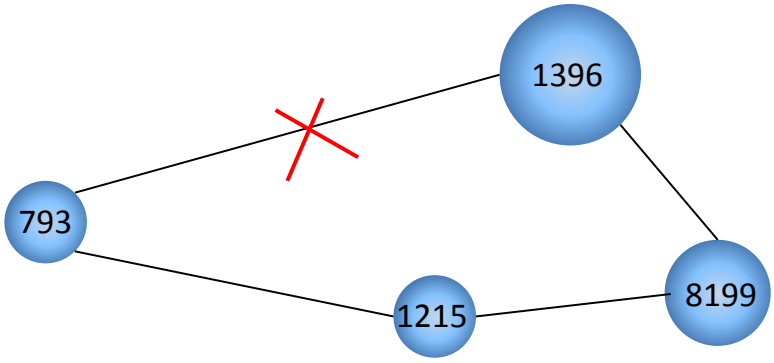

Fig. 4.7: Problem with YAMR

disconnectivity - this implies that in most of the cases, a route through the other provider gets advertised to the provider to which the link was dropped). Consequently, we limited our experiments to 20 ASes for which BGP performed the worse. The results can be seen in Fig. 4.6, which shows a graph for 2004 and one for 2010.

The results show that R-BGP performed the best, with no disconnectivity at all. And, surprisingly, YAMR didn't perform that well, despite the fact that it finds the most alternate paths. To explain the results obtained in case of YAMR, we looked in the routing tables and found that there is an extreme case in which YAMR can perform really bad. The problem is that, although YAMR finds so many alternate paths, when an AS remains with no primary path, it also withdraws all the alternate paths sent to its neighbours. It is possible that the withdrawals are propagated faster than the new announcements (because the MRAI timer is set only on announcements) thus leaving the neighboring ASes disconnected. In Fig. 4.7 we show an example of this extreme case, taken from the routing tables of the topology of year 2004. The dual-homed edged AS considered is AS793, which is linked to providers AS1396 and AS1215. AS1396 is a Tier-1 AS, with over 1000 customers. With YAMR, the primary path at AS1396 for the prefix announced by AS793 would be [793] and all the paths going through AS1215 will be labeled paths. When link (793, 1396) goes down, AS1396 can't choose any other primary path that avoids the fallen link because all the other paths, through AS1215, have been received as labeled paths. Therefore, it withdraws the primary, as well as all the labeled paths from its neighbours, leaving them disconnected until it receives a primary path that goes through AS1215.

4.4 Impact on BGP's stability

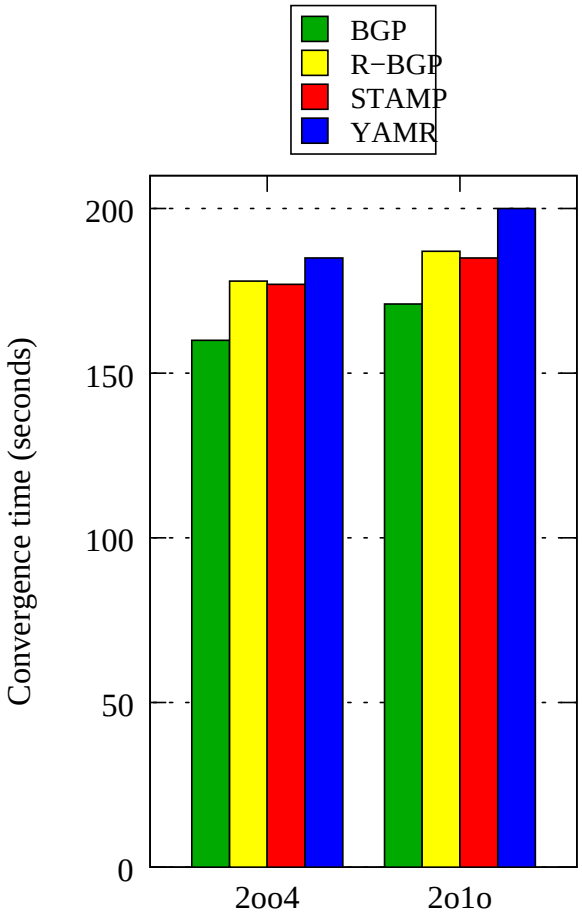

Fig. 4.8: Impact on BGP's stability (convergence time)

To study the impact on stability, we conducted the following experiment: announce a prefix and measure the convergence time (the time after which no more messages are sent into the network). We repeated the experiment for 100 different ASes. The results are plotted in Fig. 4.8, for two topologies, the one from year 2004 (left) and the one from year 2010 (right).

As can be noted in the graph, there is not a very important difference between the results obtained for the 2004 topology and those obtained for the 2010 topology: there is only a few seconds extra delay for the topology from year 2010. As expected, YAMR obtains the biggest delay, with 20-30 seconds over the convergence time obtained by BGP. The convergence times obtained by the other two multi-path methods are almost equal.

Note that the times do not necessary match the ones that would be obtained in reality. Even though the simulator introduces real time delays for network communications and the MRAI timers, other factors might influence the obtained times. For example, the

tests were run on 16 nodes from the DAS-4 cluster, which means that, to simulate over 30,000 ASes, almost 2,000 threads ran on the same machine. This could induce a few seconds delays.

Also, note that the experiment was done on single link failure scenarios. Therefore, the results are not globally valid. We return to this topic in Future Work, chapter 5. We also should mention that R-BGP has a know flaw, i.e. when multiple links of the same AS go down (this is a reasonable scenario, as there exist ASes that use a single router to connect to multiple ASes), other ASes can indeed experience transient disconnectivity.

Chapter 5

Conclusions

The aim of this project is to build insight into the multi-path routing mechanisms. As they are relatively new, these mechanisms have not been studied thoroughly and to our knowledge, there is no comparative study that shows the behavioral differences between these protocols and their impact on BGP. As we showed in chapter 3, although tests have been made to evaluate the methods, they were not primarily focused on showing the impact on BGP (some tests were indeed done in this direction, but not for all the methods). Moreover, there is almost no comparison at all, each method being implemented on a different simulator. Also, some of the tests were performed on very small, self-generated topologies, which is not sufficient to show how these protocols perform at the scale of the current Internet.

5.1 Impact of our work

We believe that our work sheds some more light into the behavior of multi-path routing mechanisms and how they will affect BGP. Our results will be useful to researchers, as they show the behavioral differences between the methods and how do various modifications affect BGP. Therefore, our results can be used to further improve the multi-path protocols. Also, our work will be useful once these methods will start to be taken into consideration as a real solution to the transient disconnectivity problem, as it also gives a comparison between the various multi-path mechanism.

Our studies on the inferred topologies from CAIDA show that the Internet is very well connected, with more and more ASes choosing to have multiple providers. This aspect is very favorable as it will provide many disjoint paths that can be explored. However, BGP was not designed to propagate multiple paths, and therefore, it doesn't take advantage to the fullest of the current Internet topology.

Our experiments show that introducing the multi-path methods in the current Internet will not have a great impact on BGP's stability, i.e. the convergence time. It will delay the convergence time with less than 30 seconds. Another aspect worth noting is that the convergence time increased very slowly from year 2004 to 2010. This means that the size of the topology doesn't have a great impact on the convergence time. Instead, what influences the most is the diameter of the topology, which seemed to have remained

almost constant. An explanation could be that the ASes that get bigger, level-up in the Tier hierarchy.

Although the impact on BGP's stability is small, the impact on BGP's scalability is substantial for STAMP and YAMR. Surprisingly, for R-BGP the number of messages sent into the network is comparable with the number of messages sent by BGP. However, STAMP sends almost double the number of messages sent by BGP while YAMR sends up to five times the number of messages sent by BGP. We cannot really quantify what these numbers mean, i.e. how big two or five times the number of messages sent by BGP really is, however, we believe that the results will be useful to the network operators.

We also studied the impact on BGP's resilience to failures. These experiments show if the methods really meet their goal, i.e. continuous network reachability during convergence time. Our results show that R-BGP achieves the best results, whereas, surprisingly, YAMR doesn't obtain a perfect score. Even though R-BGP sends the least number of messages, it manages to obtain the best results, as it strategically places the failover paths, i.e. only at important ASes. The conclusion for future research is to focus more on strategies that try to protect the most used paths, and not all the ASes. Note however that our experiments were only done for one-link failure scenarios. It is also important to see what happens when multiple links go down at the same time.

5.2 Future work

To have a complete view on the multi-path routing protocols, it is important to see what happens in each possible scenario. Because of the lack of time, we only focused on what we believe to be the most important and frequent scenarios that can happen in the Internet. However, there are still other scenarios that should be studied, among which the most important one is multiple simultaneous link failures.

Also, because of the lack of time, we only focused on a specific category of multi-path routing protocols, i.e. the multi-path protocols built on top of BGP, and on few methods from this category. However, there are other categories and multi-path methods that can be added to the study. We should also mention that we didn't have time to implement the hiding techniques proposed for the YAMR protocol. These should decresease the impact on BGP's scalability and stability, but at the cost of complexity.

Bibliography

[Ams11a] VU Amsterdam. DAS-3 clusters. http://www.cs.vu.nl/das3/, July 2011. 22

[Ams11b] VU Amsterdam. DAS-4 clusters. http://www.cs.vu.nl/das4/clusters.shtml, July 2011. 22

[Aut11] Internet Assigned Numbers Authority. http://www.iana.org/numbers/, July 2011. 7

[BGT04] Tian Bu, Lixin Gao, and Don Towsley. On characterizing bgp routing table growth. *Comput. Netw.*, 45:45–54, May 2004. 9

[CAI11a] CAIDA. Data. http://www.caida.org/data/, May 2011. 16

[CAI11b] CAIDA. The CAIDA AS Relationships Dataset, years 2004-2010. http://www.caida.org/data/active/as-relationships/, May 2011. 21

[DKF+07] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc claffy, and George Riley. As relationships: inference and validation. *SIGCOMM Comput. Commun. Rev.*, 37:29–40, January 2007. 17, 21, 23

[DKVR09] Xenofontas Dimitropoulos, Dmitri Krioukov, Amin Vahdat, and George Riley. Graph annotations in modeling complex network topologies. *ACM Trans. Model. Comput. Simul.*, 19:17:1–17:29, November 2009. 20

[DR06] Xenofontas A. Dimitropoulos and George F. Riley. Efficient large-scale bgp simulations. *Comput. Netw.*, 50:2013–2027, August 2006. 15

[EKD08] Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. On the scalability of bgp: the roles of topology growth and update rate-limiting. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 8:1–8:12, New York, NY, USA, 2008. ACM. 16

[Gao01] L. Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. Netw.*, 9:733–745, December 2001. 9, 19

[GDGS10] Igor Ganichev, Bin Dai, P. Brighten Godfrey, and Scott Shenker. Yamr: yet another multipath routing protocol. *SIGCOMM Comput. Commun. Rev.*, 40:13–19, October 2010. 3

[GGSS09] P. Brighten Godfrey, Igor Ganichev, Scott Shenker, and Ion Stoica. Pathlet routing. *SIGCOMM Comput. Commun. Rev.*, 39:111–122, August 2009. 3

[GW99] Timothy G. Griffin and Gordon Wilfong. An analysis of bgp convergence properties. *SIGCOMM Comput. Commun. Rev.*, 29:277–288, August 1999. 15

[HB96] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS), RFC 1930. http://www.ietf.org/rfc/rfc1930.txt, January 1996. 6

[HFKC08] Yihua He, Michalis Faloutsos, Srikanth V. Krishnamurthy, and Marek Chrobak. Policy-aware topologies for efficient inter-domain routing evaluations. In *IEEE INFOCOM*, pages 2342–2350, 2008. 16

[HK03] Fang Hao and Pramod Koppol. An internet scale simulation setup for bgp. *SIGCOMM Comput. Commun. Rev.*, 33:43–57, July 2003. 15

[HRA10] Geoff Huston, Mattia Rossi, and Grenville Armitage. A technique for reducing bgp update announcements through path exploration damping. *IEEE J.Sel. A. Commun.*, 28:1271–1286, October 2010. 3, 10

[Hus06] G. Huston. Exploring Autonomous System Numbers, The Internet Protocol Journal - Volume 9, Number 1. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-1/autonomous_system_numbers.html, March 2006. 7

[KKKM07] Nate Kushman, Srikanth Kandula, Dina Katabi, and Bruce M. Maggs. R-BGP: staying connected In a connected world. In *Proceedings of the 4th USENIX conference on Networked Systems Design & Implementation*, NSDI'07, pages 25–25, Berkeley, CA, USA, 2007. USENIX Association. 3, 11

[LABJ00] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. *SIGCOMM Comput. Commun. Rev.*, 30:175–187, August 2000. 1

[LG07] T. Li and G.Huston. BGP Stability Improvements, Internet-Draft. http://tools.ietf.org/html/draft-li-bgp-stability-01, June 2007. 3, 10

[LGGZ08] Yong Liao, Lixin Gao, Roch Guerin, and Zhi-Li Zhang. Reliable interdomain routing through multiple complementary routing processes. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 68:1–68:6, New York, NY, USA, 2008. ACM. 3, 12

[MBGR03] Z. Morley Mao, Randy Bush, Timothy G. Griffin, and Matthew Roughan. Bgp beacons. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, IMC '03, pages 1–14, New York, NY, USA, 2003. ACM. 16, 22

[MEFV08] Murtaza Motiwala, Megan Elmore, Nick Feamster, and Santosh Vempala. Path splicing. *SIGCOMM Comput. Commun. Rev.*, 38:27–38, August 2008. 3

[MGVK02] Zhuoqing Morley Mao, Ramesh Govindan, George Varghese, and Randy H. Katz. Route flap damping exacerbates internet routing convergence. *SIGCOMM Comput. Commun. Rev.*, 32:221–233, August 2002. 3

[oO11] University of Oregon. Route Views Project. http://www.routeviews.org/, May 2011. 16, 17, 19, 23

[Pre01] B. Premore. An experimental analysis of bgp convergence time. In *Proceedings of the Ninth International Conference on Network Protocols*, pages 53–, Washington, DC, USA, 2001. IEEE Computer Society. 9

[RLH06] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4), RFC 4271. http://www.ietf.org/rfc/rfc4271.txt, January 2006. 1, 3

[Ser11] RIPE Routin Service. RIS Raw Data. http://www.ripe.net/data-tools/stats/ris/ris-raw-data, May 2011. 16

[VCG98] C. Villamizar, R. Chandra, and R. Govindan. BGP Route Flap Damping, RFC 2439 (Proposed Standard). http://www.ietf.org/rfc/rfc2439.txt, November 1998. 3

[Woj08] Maciej Wojciechowski. Border gateway protocol modeling and simulation. Master's thesis, University of Warsaw and Vrije University of Amsterdam, July 2008. 21

[XR06] Wen Xu and Jennifer Rexford. Miro: multi-path interdomain routing. *SIGCOMM Comput. Commun. Rev.*, 36:171–182, August 2006. 3