Improving Internal BGP Routing

Virginie Van den Schrieck

Thesis submitted in partial fulfillment of the requirements for the Degree of Doctor in Engineering Sciences

November 24, 2010

ICTEAM
Louvain School of Engineering
Université catholique de Louvain
Louvain-la-Neuve
Belgium

Thesis Committee:

Pierre **Dupont** (Chair)	UCL/ICTEAM, Belgium
Olivier **Bonaventure** (Advisor)	UCL/ICTEAM, Belgium
Marc **Lobelle**	UCL/ICTEAM, Belgium
Steve **Uhlig**	T-Labs, Germany
Renata **Teixeira**	LIP6, France
Amund **Kvalbein**	Simula Research Laboratory, Norway

Improving Internal BGP Routing

by Virginie Van den Schrieck

ICTEAM
Université catholique de Louvain
Place Sainte-Barbe, 2
1348 Louvain-la-Neuve
Belgium

This work was partially supported by the European Commission
within the TRILOGY project and by Cisco Research.

To my husband and children

Preamble

While initially designed as a research network, the Internet has become a large interconnection of networks all over the world. It is in constant evolution, both in its infrastructure and in its applications.

If it was first used mainly for data exchange with a best-effort service, the applications that run on top of the network are today more and more diverse, and have different requirements in terms of delay, bandwidth or reliability. Recent analyses show that video is reaching the top of online applications, and that the focus is migrating from connectivity to content [LMIJ09]. This is also reflected in the Internet hierarchy, as large Content Providers are now attracting a large part of the Internet traffic [LMIJ09]. New technologies such as online desktop applications and cloud computing will probably reinforce this trend.

The scalability of the system is also of great concern, with more than 750 millions hosts in early 2010 [Con10]. With mobile phones being more and more connected to the Internet, this trend is not expected to slow down in the near future. Next Generation Networks will have to support this increased number of hosts and amount of traffic while remaining able to fulfill the new performance requirements of their customers.

Increasing the resources of the current Internet infrastructure in parallel with its robustness and stability is thus an important challenge for the next years. This requires to make the physical network evolve, but also to improve routing, i.e. all the mechanisms needed to establish and use Internet paths. Improving global routing is however not easy to tackle, due to the decentralized structure of the Internet. While it was initially composed of a few networks that were able to cooperate quite easily, the Internet is now much more complex, and relationships between interconnected networks are driven by business. Collaboration is not possible in the absence of strong economic incentives, and the short-term benefits of investing in global routing evolution are not obvious.

In this thesis, we focus on the Border Gateway Protocol (BGP), the interdomain routing protocol that is used in the Internet to build and disseminate end-to-end paths across networks. The complexity of changing the way the Internet works is reflected on the difficulty of changing this protocol, as all routers of all networks must be modified to remain interoperable. BGP can only evolve through incremental deployment of new functionalities compatible with legacy routers, or through modifications restricted to individual ISPs.

Indeed, if the BGP protocol is used to exchange path information between networks, it is also used to propagate information received from neighboring networks among all internal routers of an ISP. Even though network operators have very limited ways to influence the routing of their external neighbors, they have a full control on the way BGP behaves inside their network. This opens a door for potential improvement : While it is not easy to change the BGP protocol globally, local modifications are more realistic. Improving the way BGP works internally should increase routing stability and efficiency inside ISP networks. But given the interconnections between ISPs, we can also expect that improvements inside an ISP will have a positive impact on the stability of its neighbors, and transitively, on the global Internet. The main purpose of this thesis is thus to **improve BGP locally** (i.e. inside ISPs) **with the objective of increasing the stability of BGP globally**.

We start with an analysis of the behavior of BGP inside Internet Service Providers, to study to which extent the internal part of BGP routing (iBGP) influences the whole interdomain routing system. Based on those results, we explore different solutions to improve this internal BGP routing with the objective of improving global Internet routing. The main contributions of the thesis are the following:

- We analyse the issues of internal BGP routing, based on a survey of the literature and on measurements with ISPs routing data. We show that BGP paths are poorly propagated inside ISP networks, and that this contributes to global Internet instability.
- We propose three mechanisms to improve internal BGP routing, while keeping the existing single path advertisement principle. The first mechanism improves the isolation of BGP in case of locally recoverable failure. The second provides a solution for fast recovery in case of failure. The third is a new internal BGP organization that is automatically configurable.
- Finally, we relax the single path advertisement constraint, and study the advertisement of multiple BGP paths among routers of the same ISPs. We analyse the cost/benefit tradeoffs between different sets of such paths, and developed a tool able to provide a quantitative evaluation of the deployment of this mechanism inside ISPs and on the global Internet. This tool relies on simulation to provide a set of metrics measuring the behavior of the network when multiple paths are advertised.

Road map

The thesis is organized in three parts. The first part of the thesis provides the background required, and explores the research challenges in BGP and iBGP routing. The first two chapters can be skipped by readers familiar with Internet Routing and BGP issues. The third chapter contains a survey of the literature about iBGP issues, as well as some contributions to measure the impact of those issues. As this

chapter formulates the problem statement of the thesis, it is not recommended to skip it, except if the reader is already expert in iBGP.

The second part of the thesis proposes three mechanisms for improving internal BGP routing, with the traditional single path advertisement. In the last part of the thesis, we focus on the advertisement of multiple paths between routers inside an ISP. Each of those two parts can be read independently of the others, provided that the reader has sufficient knowledge of the iBGP weaknesses presented in chapter 3.

Part I - BGP Analysis

In **Chapter 1**, we give an overview of the organization of the Internet and present the concept of routing, which consists in building paths across networks and routers such that packets can be exchanged between end-hosts. We then focus on the details of the BGP protocol. We also explain how an ISP can influence routing by manipulating BGP attributes depending on its own routing policies.

Chapter 2 is a survey of the research challenges related to BGP routing. The main concern is the scalability, but BGP also suffers from convergence issues which either delay the recovery in case of failure, or create oscillations in the routing system.

In **Chapter 3**, we survey the issues at the level of internal BGP routing. We present measurements based on routing information from three ISPs to support this analysis. We show that internal BGP routing suffers from the same issues as global BGP routing, i.e. scalability concerns and convergence inconsistencies. But the main issue is that paths are not sufficiently propagated among routers and this contributes to the Internet instability. We measure this lack of diversity inside the three ISPs, and show that this issue is not negligible.

Part II - Improving iBGP

Based on the observations of the first part of the thesis, we tackle the challenge of improving internal BGP (iBGP) routing. We start by listing in **Chapter 4** the requirements for a good iBGP system. We survey the existing proposals for improving iBGP, and provide a taxonomy of those solutions based on the listed requirements.

Then, in **Chapter 5**, we propose a first solution to improve iBGP. This solution aims at improving network isolation in case of failure, when internal BGP is not able to provide a local recovery. This solution consists in overcoming the lack of alternate paths by advertising the existence of such paths among routers. This mechanism has the advantage of being incrementally deployable inside the ISP.

In **Chapter 6**, we focus on fast recovery from failure. The objective is to minimize the packet loss in case of failure by combining a new FIB organization with a mechanism providing alternate paths on selected routers. This solution is well suited to protect important customers. The BGP part of the mechanism can be

automatically configured. This auto-configuration scheme also forms the basis for the new iBGP organization presented in **Chapter 7**.

Part III - BGP Add-Paths

While the solutions presented in the second part of the thesis were designed with the objective of minimizing the changes to the existing BGP, in the last part, we focus on a modification of the BGP protocol allowing to advertise multiple paths between iBGP routers [WRC09a]. This modification is called *Add-Paths* [WRC09a]. Even though such change is likely to increase the resources needed for iBGP, it has several advantages and can be used to fulfill several of the requirements listed in Chapter 4. It is currently undergoing standardization within the Internet Engineering Task Force (IETF).

As only the encoding of the advertisement of additional BGP messages in iBGP has been described in the specification of Add-Paths, the selection of those additional messages is yet to be defined. We provide in **Chapter 8** a qualitative analysis of different possible sets of paths, based on several possible application of Add-Paths. We also study the convergence properties of those sets of paths.

In **Chapter 9**, we present a software tool that we developed to provide quantitative analyses of the deployment of Add-Paths inside ISPs through simulation. Based on this tool, we extend the analysis of the previous chapter with simulation results on synthetic topologies. We also analyse the gain in isolation that can be obtained when propagating the proper set of paths in iBGP. This analysis shows that deploying Add-Paths in iBGP can have a positive impact on the global interdomain churn.

Finally, we conclude the thesis by extending our taxonomy of iBGP solutions with the mechanisms proposed in this thesis, and by presenting future perspectives on this subject.

Bibliographic Notes

Most of the work presented in this thesis has been previously published in conference proceedings and journals, or presented in workshops. The list of related publications is shown hereafter:

- V. Van den Schrieck, P. Francois, S. Tandel and O. Bonaventure, *Let BGP speakers configure their iBGP sessions on their own*. October 2006. Position Paper, Wired2006 Workshop, Atlanta
- V. Van den Schrieck, *Automating iBGP organization in large IP networks*. Proc. ACM CoNEXT Student Workshop, p.41, New York, USA, december 2007
- P. Mérindol, V. Van den Schrieck, B. Donnet, O. Bonaventure and J-J. Pansiot, *Quantifying ASes Multiconnectivity using Multicast Information*. Proc.

ACM USENIX Internet Measurement Conference (IMC), p.370-376, November 2009

- V. Van den Schrieck, P. Francois, C. Pelsser and O. Bonaventure, *Preventing the Unnecessary Propagation of BGP Withdraws*. Proceedings of IFIP Networking, p.496-508, 2009

- B. Quoitin, V. Van den Schrieck, P. François and O. Bonaventure, *IGen: Generation of Router-level Internet Topologies through Network Design Heuristics*. Proceedings of the 21st International Teletraffic Congress, p.1-8, September 2009

- V. Van den Schrieck, P. Francois and O. Bonaventure, *BGP Add-Paths: The Scaling/Performance Tradeoffs*. IEEE Journal on Selected Areas in Communications, 28(8):1299 - 1307, oct. 2010.

Several contributions were also submitted to the IETF. Here is the list of corresponding Internet-Drafts :

- V. Van den Schrieck and O. Bonaventure, *Routing oscillations using BGP multiple paths advertisement*. Internet draft, June 2007. draft-vandenschrieck-bgp-add-paths-oscillations-00.txt.

- V. Van den Schrieck and P. Francois, *Analysis of paths selection modes for Add-Paths*. Internet draft draft-vvds-add-paths-analysis-00, July 2009

- J. Uttaro,V. Van den Schrieck, P. Francois, R. Fragassi, A. Simpson and P. Mohapatra, *Best Practices for Advertisement of Multiple Paths in BGP*. Internet draft draft-uttaro-idr-add-paths-guidelines-03.txt, October 2010

Acknowledgments

The fulfillment of this thesis would not have been possible without the contributions of many persons. I wish to thank all of them, and I apologize in advance to those not listed here.

First of all, I am sincerely grateful to my advisor, Olivier Bonaventure. Olivier is a very mindful advisor, with an impressive knowledge of networking and routing. I really appreciate his pragmatic view of research, and his concern for exploring solutions "that really work in the real world". Throughout all those years, he provided me a lot of support, ideas and help. I would like to thank him for giving me the opportunity to work with him and explore the world of BGP, as well as for its understanding and encouragements, especially when family stuffs kept me away from research for a while. And, also... I wish to thank him for encouraging me to enroll in the volleyball team of the departement. I had a lot of fun learning and playing volley every week.

I am also really thankful to the members of my thesis committee, Steve Uhlig, Marc Lobelle, Amund Kvalbein, Renata Teixeira and Pierre Dupont. Thank you all for having taken time to review my thesis, and for the interesting discussions during the private defense. You gave me a lot of interesting feedback and valuable comments for improving my manuscript.

To my great pleasure, my research was far to be a lonely task, thanks to my colleagues of the IP Networking Lab. Among them, I owe a special thanks to Pierre Francois, with whom I had a lot of interesting research discussions, preferably in front of a coffee or a milkshake. Pierre is a passionate researcher, never giving up any idea, and curious about everything. I had a great time working on Add-Paths selection modes with him, and am very grateful for all the job he is doing for promoting our joint work all around the world.

I am also sincerely thankful to Bruno Quoitin, who is a skilled developer as well as an expert in BGP. I enjoyed working with C-BGP and iGen, whose codes are particularly clear and easy to understand. In addition to having benefited from its tools, I was also very lucky to be able to discuss and have feedback from him about my work.

Cristel Pelsser has been of great support to me as well, and was always eager to provide valuable feedback about my research. Even though we started working on similar subjects after she left our team for going to Japan, Cristel has always been available for discussions and for collaboration. I would like to thank her for all the

valuable remarks, reviews and comments she has provided about my papers.

Pascal Merindol and Benoît Donnet gave me the opportunity to explore with them the potentiality of the mrinfo tool developed by Jean-Jacques Pansiot. Laurent Vanbever is working on topics similar to mine, and we had the opportunity to share some very interesting discussions. Cédric Delaunois has developed the Ghitle AS-level topology generator, which has been really useful to me several times during my thesis.

My other colleagues, former or current, even when working on different topics, also contributed to this thesis indirectly, by sharing their knowledge about programming languages and interesting tools that facilitated my everyday work. Thanks to Sébastien Tandel, Steve Uhlig, Damien Leroy, Sébastien Barré, Damien Saucez, Grégory Detal, Christoph Paasch and Simon van der Linden. I also wish to thank the whole INGI team, academic, administrative and technic, for all their help during those years at the UCL.

Other persons outside UCL also contributed to this thesis. Clarence Filsfils from Cisco gave me the opportunity to work on PIC and provided very interesting feedback on my research. I am also very grateful to all people who allowed me to work on real ISPs routing data.

But of course, this thesis would not exist without the love and support of my family, and especially without my parents, who gave me the taste for science and mathematics. Thanks also to my brother Jean-Christophe for sharing his thesis experience with me. I owe also much to my grand parents, whose advices and opinions are particularly important for me.

I have a very special thought for my son, Arthur. His birth was a joyful interruption in my research, and gave me renewed motivation when I came back to work. He learned me to rethink my priorities, and to optimize my working time. He is an amazing little boy, demanding a lot, but giving so much in return.

I also wish to mention the tiny little person within me, that had to support a few periods of stress during the end of this thesis. His presence was very comforting, and I am really impatient to welcome him next spring.

Last, but not least, I am particularly grateful to my husband, Geoffrey, for his continuous love and support. He is always encouraging and ready to hear and comment about my research issues so different from his. When I was writing the manuscript, he often relieved me from the little daily troubles, including, but not restricted to, being always ready to wake up several times at night for our son, and spending a lot of time with him at bedtime. Thank you for all that, I hope to be able to do the same for you soon.

Virginie Van den Schrieck
November 24, 2010

Contents

List of Figures

List of Tables

Part I

Background

Chapter 1

Introduction

1.1 Organization of the Internet

Today's Internet is part of the daily life of lots of people in developed countries. Instantaneous written or spoken communications are common, and the Web is the largest available source of information in the society. Shopping, gaming and file sharing are also widespread activities on the Internet, which has become a huge economical system. More recently, the 2009 Internet Observatory report reveals that content distribution networks, online applications and video content as YouTube are pushing the Internet into new commercial, security and engineering challenges [LMIJ09]. While the Internet is probably seen by most as a sort of blackbox to which they have access thanks to their web browser, it is actually a huge interconnection of devices all over the world, interconnection that spreads at different levels. Those devices have to communicate with each other to allow the communication of the end-host. In this chapter, we present the structure of the Internet, and explain the mechanisms used to establish communication paths between end users.

1.1.1 Physical end-to-end connectivity

An end-user is often connected to a **Local Area Network** (LAN), which allows him to communicate with other devices in the same LAN. It is for example the case in a house with several computers, or in a small enterprise. A LAN might be connected with other LANs close to them. This is typically the case in an enterprise having different departements or services. Dedicated devices called switches will often be used to interconnect those LANs. But if anyone wants to communicate with other devices, outside the house or outside the enterprise, he will have to rely on a service provider, which is able to interconnect its different customers. However, in this scheme, an end-user cannot communicate with another host that is not customer of the same service provider. This is why service providers connect with each other and agree to carry the traffic between their customers. Service providers

offering connectivity to the Internet to their customers thanks to their interconnections with others services providers are called **Internet Service Providers** (ISPs)

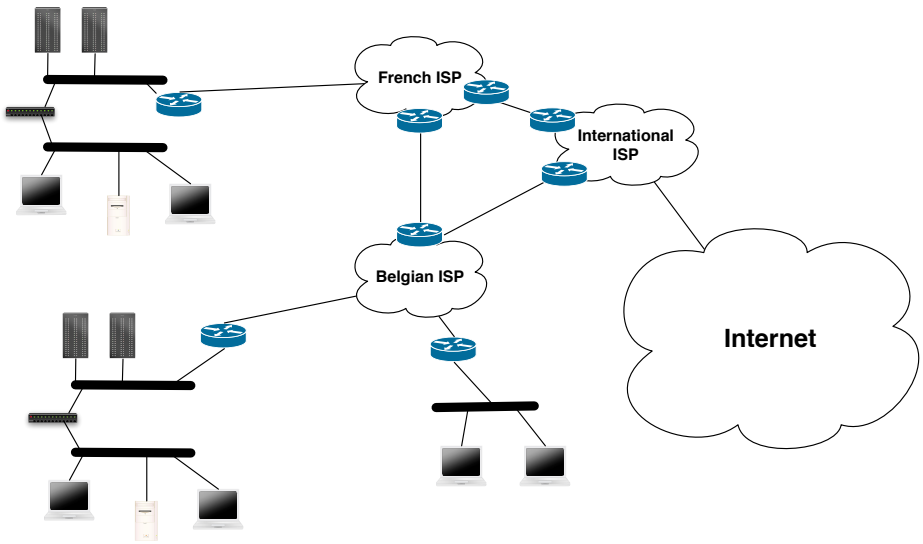


Figure 1.1: End-to-end connectivity

1.1.2 Connectivity between ISPs

Interconnections between service providers are usually established at Internet eXchange Points (IXPs), which are locations with a specific physical infrastructure (routers, switches) allowing networks to exchange their traffic with each other. Using such dedicated IXPs reduce the cost of having to set up specific connections with each new peer. The costs of maintaining an IXP is shared between all IXPs participants. IXPs can be found in every large city worldwide.

Service providers to which end-users connect might be local providers, and their infrastructure is often geographically limited. They are able to establish connections with service providers present at the same IXPs as them, but they can't directly reach every destination in the Internet. Those small service providers typically buy connectivity to the rest of the world from larger service providers, that are present worldwide and connect with lots of networks. Typically, a national ISP won't be able to communicate with other national ISPs that are not located in the same region or continent. They will thus ask international ISPs to take care of their traffic with distant destinations. In the example of figure 1.1, the French and Belgian ISPs peer with each other, but rely on an International ISP to reach the rest of the Internet.

Two types of peering relationships are often encountered between ISPs. Those relationships are also called **business relationships**[Gao01]. They can be Shared-

cost peering (P2P), where two ISPs agree to exchange traffic to their customers, or Customer-to-Provider (C2P) peering, where an ISP buys connectivity from a larger provider. The provider allows him to reach all the destinations that it is able to reach itself.

The Internet is thus a hierarchy of ISPs: small, regional ISPs that depend mostly on providers for their connectivity are the bottom of the hierarchy. Larger ISPs that are providers for lots of other ISPs and rely on shared-cost peering for global connectivity are the top of the hierarchy, i.e. they don't have any provider themselves. They are often called **Tier-1 ISPs**

1.1.3 Routing

Interconnecting routers and ISPs is a necessary step for a working Internet, but it is not sufficient. Paths between end-users exist, but there is, at this point, no map to find which of those paths is to be followed to connect two devices. In the Internet, the information exchanged between two devices are split into packets, and each packet is sent through the network independently, hop by hop, following the rules of the **Internet Protocol (IP)**. At each step, an intermediary device (a router) checks the identifier of the destination (IP address) in each packet, and should forward it to the neighboring router that will lead the packet to its destination. Thus, each router needs to know to which neighboring router (called **nexthop**) it must forward packets for each possible destination. The process of computing the path for each destination, i.e. the nexthop router, is called **routing**.

IP prefixes

Routing protocols allow routers to compute their **Routing Table**, which contains the list of reachable destinations and the corresponding nexthop. A routing table does not contain IP addresses directly, but rather sets of IP addresses, to keep the data structure scalable. Those set of IP addresses are represented by their common prefix, called **IP prefix**. An IP prefix is represented by an IP address contained in the prefix, and the length of the prefix itself. For example, the IP prefix $3.0.0.0/8$ contains all addresses between $3.0.0.1$ and $3.255.255.255$. The length of that prefix is 8 bits, and it represents 2^{32-8} IP addresses. Upon reception of a packet for a destination, the router performs a lookup in its routing table, and searches the longest IP prefix that contains the IP address of the destination. This algorithm is called **Longest Prefix Match**. As an example, a router has two prefixes in its Routing Table, $3.0.0.0/8$ with nexthop $R1$ and $3.1.0.0/16$ with nexthop $R2$. The router receives a packet to destination $3.1.2.4$. Both prefixes match that IP address, but the second one is longer, so the packet is forwarded to $R2$. Thus, destinations that routing protocols consider are not IP addresses, but IP prefixes.

Intradomain routing

Routing is performed at two levels. At the intradomain level, the Interior Gateway Protocols (IGP) are responsible to establish the paths between the devices inside the network. IGP protocols are usually link-state protocols, which means that routers exchange information about the physical links between each other and the prefixes of local destinations. Then, based on this information, each router builds a representation of the network and computes the shortest paths to each destination.

Interdomain routing

When ISPs establish connections with each other, they have to exchange their reachable prefixes in order to exchange packets for those destinations. Routing protocol dedicated at exchanging routing information between different ISPs are called **Interdomain Routing protocols**.

At the origin, the Internet was an interconnection of networks using TCP/IP to exchange data with each other. Reachability information was exchanged between networks via the External Gateway Protocol (EGP)[Mil84]. However, as EGP was backbone-centered and limited to tree-like networks (its behavior was undefined with loops in the network). As such, it was not designed to support a growing Internet, with Autonomous Systems becoming more and more complex, independent and competing.

The Border Gateway Protocol (BGP) was thus proposed as a new interdomain routing protocol, emerging from a cafeteria discussion between Yakov Rekther, Len Bosack and Kirk Lougheed during an IETF meeting, in January 1989 [WMS04]. The first draft of the protocol specification was written on three napkins, but as time elapsed, four versions of the protocol were successively proposed. New features were added to the protocol, such as Classless InterDomain Routing (CIDR), Route Reflection and Confederation or BGP/MPLS VPN support [WMS04]. Today, BGP4[RLH06] is still used as the de-facto InterDomain Routing Protocol in the Internet.

1.2 The Border Gateway Protocol

The goal of BGP is to provide a way to exchange reachability information between Autonomous Systems. RFC 4271 [RLH06] defines an Autonomous System as follows *"An autonomous system is a set of routers under a single technical administration, using an interior gateway protocol (IGP) and common metrics to determine how to route packets within the AS, and using an inter-AS routing protocol to determine how to route packets to other ASes."* We can summarizes this definition by stressing that an AS appears as a single coherent entity under a single technical administration to the other ASes. Most of the time, an Internet Service Provider will act as a single Autonomous System.

BGP is a Path Vector protocol in which reachability information is represented by IP prefixes that aggregate IP adresses of sets of destinations. The BGP Path vector is the AS-Path, which is the list of ASes that form the path to the destination prefix.

Each prefix is normally originated by an AS that is able to directly reach the IP addresses belonging to that prefix. It will thus advertise a BGP message (**BGP Update**) containing a path to that prefix to its BGP neighbors, which in turn propagate this information to their neighbors such that the prefix becomes globally reachable. If several paths are available per prefix, only one of them is advertised to the neighbors. When a router receives a packet to an IP address belonging to a prefix that it learned via BGP, it transmits that packet to the neighboring router from which it received the corresponding BGP Update message. This neighboring router is called the Next-Hop router. The propagation of the prefix reachability information is constrained by a simple rule, designed to prevent BGP Update messages to loop in the network: When a router belonging to an Autonomous System forwards a BGP Update message to another Autonomous System, it will add its identifier, called AS Number (ASN) into the AS-Path data structure of the BGP message. Then, upon reception of a BGP Update message from another AS, a router will look into the AS-Path and discard the BGP Update if it contains its own AS Number, revealing a routing loop.

In the example of figure 1.2, the Belgian ISP announces a BGP Update with the prefix of its customer to its BGP neighbors AS20 and AS30, with only its own AS number (AS10) in the AS Path. The neighbors add the prefix, along with the router from which it was received, into their routing table such that they know on which link they have to forward the packets to the IP adresses contained in the prefix. Then, they also advertise the prefix to their own neighbors, this time appending their respective AS number into the AS Path. AS10 discards the Update messages received from AS20 and AS30 because its own AS number is in the AS Path.

When a prefix is no longer reachable, BGP routers will tell their neighbors that they cannot reach the prefix anymore by advertising another sort of BGP message, called **BGP Withdraw**. A BGP Withdraw only contains a prefix that was previously advertised by the router, and that is no longer reachable via this router. In the example of figure 1.2, if the link between the Belgian ISP and the $2.0.0.0/24$ network fails, a BGP Withdraw will be sent, in a similar fashion as the BGP Update it cancels.

BGP uses incremental Updates. That means that, after having exchanged all their prefixes upon session establishment, BGP routers only need to send BGP Updates to each other if a path towards a prefix changes. The new BGP Update implicitly replace the previous BGP message for the same prefix. As no periodical refresh of BGP advertisement are needed by the protocol, BGP needs to be run upon a reliable transport protocol. For this reason, BGP runs on top of a TCP connection.

Figure 1.2 presents an example of incremental Update in the following scenario. The international ISP has two paths to the prefix $2.0.0.0/24$, one via AS10

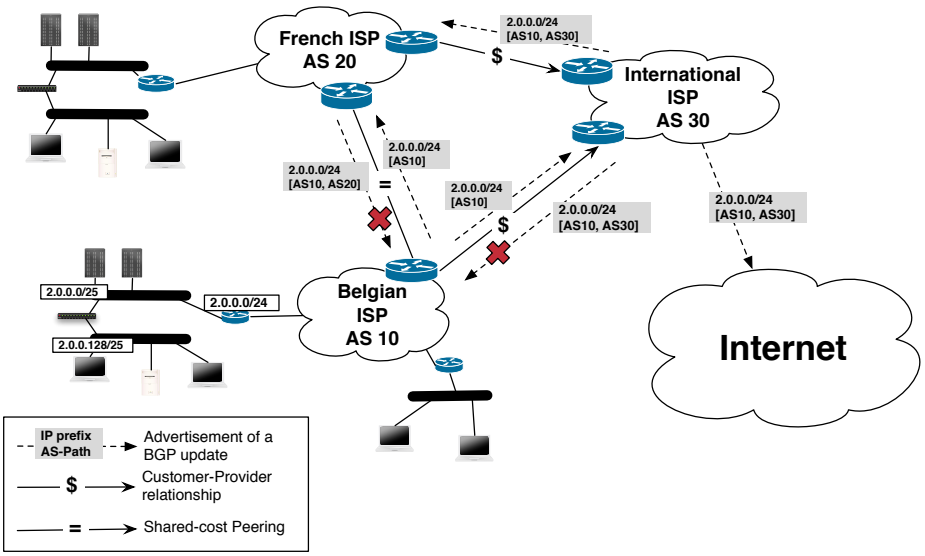
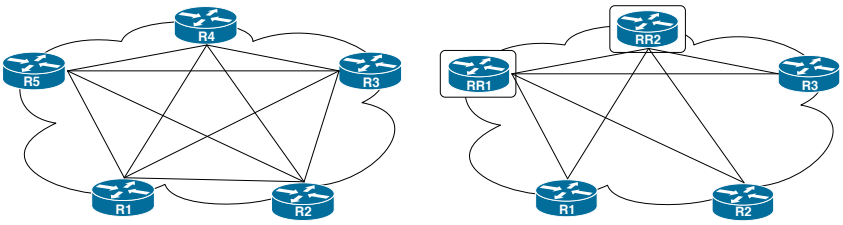


Figure 1.2: Exchange of BGP Updates

and one via AS20, then AS10. Initially, it prefers the path via AS10 over the path via AS 20, and advertises prefix $2.0.0.0/24$ with AS Path [AS10 AS30] to the rest of the Internet. If, for some reason, AS30 changes its mind and prefers the path via AS20 over the path via AS10, it will advertise a new BGP Update with AS Path [AS10 AS20 AS30] to the Internet. This path will implicitly replace the previous one.

1.2.1 iBGP organizations

An Autonomous System is usually composed of several routers, most of them being connected to one or several BGP routers from neighboring ASes. In order to maintain a consistent routing state inside the autonomous system, the routers of the AS need to exchange the BGP paths they receive from their external neighbors. This is why there will also be BGP sessions between them. A BGP session between routers belonging to the same Autonomous System is called **internal BGP (iBGP) session**, while a BGP session between routers belonging to different autonomous systems is called **external BGP (eBGP) session**. When a BGP Update message is advertised over an iBGP session, the AS number of the local AS is not appended to the AS Path. It is appended only when the BGP Update is advertised outside the AS, over an eBGP session. Routers in an AS can establish their iBGP sessions in different ways. Classical iBGP organization are Full Mesh [RLH06], Route Reflection [BCC00] and confederation [TMS01]. In this thesis, we will only focus on the first two organizations, as they are the most widely used.



(a) Full-Mesh of iBGP sessions (b) Route Reflection

Figure 1.3: iBGP organizations

Full-Mesh

The **Full-Mesh** is the simplest organization of iBGP sessions, where all BGP routers of an AS have an iBGP sessions with each other, as shown in figure 1.3(a). As all routers receive all paths from all iBGP peers, there is no need for a router to re-advertise any iBGP-received path over an iBGP session. Only eBGP-received paths are advertised over iBGP sessions.

With a Full-Mesh, there will thus be $\frac{n.(n-1)}{2}$ iBGP sessions in the AS. While this can be reasonable in small domains, larger ISPs will need a more scalable organization.

Route Reflection

Route Reflection [BCC00] was introduced as a more scalable iBGP organization. The idea behind Route Reflection is that some routers are elected as **Route Reflectors** and will be responsible for providing BGP paths to other routers, called their **clients**. In figure 1.3(b), there are two Route Reflectors, RR1 and RR2, and the three other routers are their clients. In this organization, the Route Reflectors are allowed to re-advertise iBGP-received paths over iBGP sessions, following those two rules:

1. Advertise all paths to client BGP neighbors
2. Advertise paths from client BGP neighbors to non-client BGP neighbors

A typical implementation of Route Reflection is to choose two routers as Route Reflectors inside a PoP Point of Presence, i.e. physical location at which an ISP provides connectivity to customers, and to connect all BGP routers of the PoP to both Route Reflectors [BCC00]. The example of figure 1.3(b) respects this design rule, in the case of a single-PoP network. This redundancy is needed for robustness reasons, in case of failure of one Route Reflector. All Route Reflectors of the AS are then connected with a Full-Mesh of iBGP sessions. In larger network, a hierarchy of two levels of Route Reflectors can also be used, with some Route Reflectors being themselves clients of other Route Reflectors.

iBGP will be studied in more details in the next chapters of the thesis.

Path attribute	Description
AS Path	List of AS Numbers
Nexthop	IP Address of the router that should be used as nexthop for the destinations of this BGP Update
LOCAL_PREF	Integer
MED	Integer (optional attribute)
Communities	Set of 32-bits values (optional)

Table 1.1: Main BGP attributes

1	Highest Local-Preference
2	Shortest AS-Path
3	Lower MED (only between paths from same neighbor AS)
4	eBGP over iBGP
5	Closest nexthop
6	Tie-Break

Table 1.2: BGP Decision Process

1.2.2 BGP Best Path selection

Due to the way Autonomous Systems are connected to each other, it may happen that routers receive several BGP paths to a given prefix from their neighbors. Only one of those paths is generally used to forward traffic, thus a choice needs to be done between all paths to select the **Best Path**. Only the best path will be subsequently advertised to BGP neighbors, as it represents the path normally followed by packets transiting through that router.

An AS has full control of the best path selections that its routers will perform on the set of available paths. The operators will thus configure the routers such that the policies of the AS can be enforced. In practice, this will be implemented by changing the **attributes** of a BGP path that are taken into account by the rules that a router applies for choosing a path. The attributes of a BGP path are listed and described in table 1.2.2.

The set of rules applied on the BGP attributes and used to select the best path of the router is called the **BGP Decision Process**. The rules are applied successively on candidate paths, until a single candidate remains. The last remaining path is elected as the Best Path.

Table 1.2.2 lists the rules of the decision process in the order in which they are applied. In the next paragraphs, we detail each of them more precisely.

Local Preference

The first attribute that is considered by the Decision Process in the **Local Preference** attribute. It is a numeric value that translates the preference of an operator for a given path. This attribute is local to the AS, which means that no local preference is transmitted over eBGP sessions.

Shortest AS-Path

The second rule of the Decision Process compares the **length of the AS-Paths** of the candidate paths. Shortest AS-Paths are preferred, because they can be seen as an indication of the shortest path to the destination. This attribute is global, as the AS-Path is transmitted across domains, each AS adding its own AS number once or several times at its end.

Multi-Exit Discriminator

The next attribute that influences the choice of the best path is the **Multi-Exit Discriminator (MED)**. This attribute is typically configured in a path by an AS wishing to influence the best path selection of its neighbor when it advertises that path over several eBGP sessions. The AS will set a lower MED value on the BGP Update message(s) advertised over the eBGP link(s) that it wants to be used by the neighbor. Upon reception of those paths, if the preceding rules did not remove any of those paths from consideration, the neighbor will compare the MED values of the paths and choose the lower one. As the MED value is typically used by an AS on the paths advertised to a neighbor, this attribute is not comparable between paths advertised by different ASes. For example, if a router receives three paths with MED values, two from AS A (MED values of 5 and 10) and one for ASB (MED value of 10), it will compare the paths from ASA and remove the one with MED 10. After applying this rule, the two remaining paths are thus the one from ASA with MED value of 5, and the one from AS B with MED value of 10. The decision process needs to apply subsequent rules to elect the best path.

The particularity of MED being applied on a per-neighbor basis violates the rule of independent ranking mentioned in [GW02a]Thus, the BGP decision process cannot be considered as a total ordering of the paths to a prefix, because of the MED attribute.

eBGP over iBGP

Up to this rule, the BGP decision process is global or AS-Wide, i.e. the outcome is dependent on BGP attributes and not on the location of the router performing the choice. The fourth rule takes into account the BGP session on which the path was received by the router: The paths received over an eBGP session are preferred over the paths advertised over an iBGP session. The goal of this rule is known as the **Hot Potato** principle [TSGV04]: In order to minimize the traffic crossing

its infrastructure, an AS will try to send the traffic outside its domain as soon as possible. For a router, this means forwarding preferably the packets on an eBGP link.

IGP cost

When the remaining candidate paths are all received over iBGP sessions, Hot Potato is enforced by the following rule: A router will prefer the paths with the closest nexthop in terms of IGP distance. The transit cost of packets forwarded to this destination is minimized, as the cost of the path followed through the AS is the cheapest.

Tie-breaking rules

When all candidates paths are equivalent from all the above rules' viewpoint, the router needs to use a tie-break to select its best path. The identifier of the neighbors that advertised each path is thus used to this end.

1.3 BGP policies

The Routing Policies of an AS are the set of rules defining the way paths are selected and disseminated to the neighbors. Typically, an AS will prefer to forward traffic to some neighbor instead of others, or prefer not to transit traffic from given neighbors for some destinations.

In BGP, the implementation of those policies relies on two mechanism. First, the rules of the decision process are dependent on the attributes of BGP paths. Thus, by tweaking those attributes, the operator can influence the best path selection. They have indeed the opportunity to configure the routers such that they modify the BGP attributes of the paths they receive or advertise, by using inbound and outbound filters. Such filters can also be used to control the prefixes that are advertised to each neighboring router.

Inbound and outbound filters are defined on a per-session basis, i.e. they are applied on all BGP messages received from a neighbor or sent to a neighbor. The granularity can also be specified, such that a filter is applied only on a subset of the prefixes received on a session. The actions performed by a filter are of two types. First, a filter can define wether a BGP message can be accepted from a peer as a candidate best path or not (inbound filter), or if a message can be sent to the corresponding peer or not (outbound filter). If the message is accepted by the filter, a change in the BGP attributes can be applied. Modifiable attributes are typically the Local-Preference, the AS-Path (with prepending) and the Multi-Exit Discriminator.

Figure 1.4 summarizes the behavior of a BGP router. It shows that BGP messages received from peers cross the inbound filters related to that BGP session. They are then stored in a database storing the paths received from that peer. This

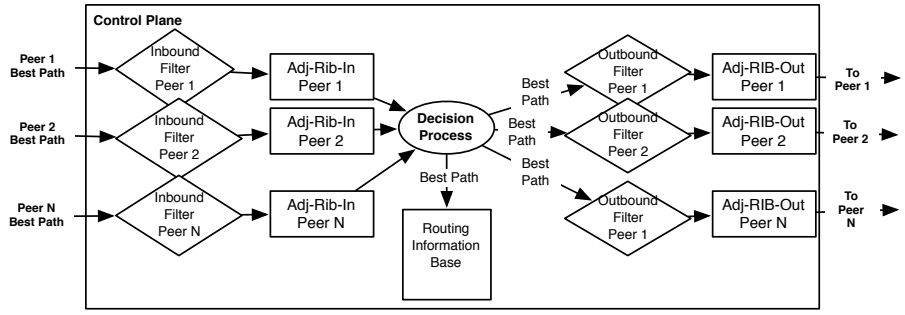


Figure 1.4: Control-plane of a BGP router

inbound database is called Adj-RIB-In. All paths stored in the Adj-RIB-Ins are used by the decision process to compute the best path. This best path is stored in the Routing Information Base (Rib), and will be installed in the Forwarding Information Base (FIB) of the forwarding plane, i.e. the effective routing table used for packet forwarding. The best path is also propagated to the peers, after traversal of the outbound filters. The paths sent to each peer are stored in the corresponding Adj-RIB-Out database.

Sometimes, an operator may want to implement an outbound policy that depends on the properties of the session on which the path was originally learnt. Those properties are for example the type of peering, or the location of the peering link. BGP communities [CTL96] can be used to identify the properties of a BGP path. A community is simply a 32 bits transitive attribute which does not participate in the decision process, but that can be used in a filter to select BGP paths on which a specific action must be applied. Thus, a community is a sort of tag with different possible meanings. Some have a standardized function, such as the $NO-EXPORT$ community that is appended to BGP messages sent to a peer AS, asking him not to advertise this message to its own neighbors. But for most of them, the operator has the opportunity to use them for any purpose. A review of the different usages of BGP communities can be found in [BQ03, DB08]

1.3.1 Business relationships enforcement

We explained earlier that Autonomous Systems agree to carry each other's traffic under certain conditions. Typical agreements are Shared-Cost peering and Customer-Provider relationship. For Customer-Provider relationship, a transit AS allows a customer to use its infrastructure to reach all the destinations it knows, and will accept all traffic destined to the customer. In the example of figure 1.5, AS 20 is the customer of AS 40 and sends its traffic through AS 40 to reach all the Internet. AS 40 is itself the customer of AS 2 and AS 3. Thus, AS 40 wants traffic to flow towards its client because it is paid for that, but will try to limit the traffic sent to its

providers, because it has to pay for it. The only traffic it will allow to be forwarded to its providers is the traffic coming from or going to itself or its customer. The implementation of such a policy can be be enforced thanks to communities. A community specifying the origin of a path such as $ORIGIN\_CUSTOMER$ (resp $ORIGIN\_PROVIDER$) is tagged by the router receiving a path on some eBGP session with a customer (resp. provider). Outbound filters are configured such that BGP paths with the $ORIGIN\_CUSTOMER$ are advertised to all neighbors, but BGP paths with $ORIGIN\_PROVIDER$ are only advertised to customers. In the example, the paths from AS2 and AS3 are advertised only to AS20, while the paths of AS20 are advertised to all neighbors.

In a shared cost peering, each AS agrees to carry the traffic from the customers of the other Shared-Cost peer and destined to its own customers. In terms of BGP messages dissemination, this means that each AS will allow the BGP prefixes received from a shared-cost peer to be advertised to its customers, but not to other shared-cost peers or providers. In the example of figure 1.5, AS 40 wants to implement a Shared-Cost Policy with AS 30. That means that it wants to allow traffic to flow between AS 10 and AS 20, which are the customers of both ASes, but not, for example, between AS 30 and AS 3, because AS 3 is a provider and it has to pay for the traffic going to this AS. Thus, AS 40 will only advertise the paths from its shared cost peer AS 30 to its customer AS 20. Upon reception of a BPG message from AS 30, router $R1$ will tag a community identifying the type of neighbor that advertised the route, say, in this case, $ORIGIN\_PEER$. Then, it re-advertise the BGP message to other iBGP routers. If that prefix is selected as best by the border routers, they will consider their outbound filter, and see if there is a rule corresponding to the $ORIGIN\_PEER$ community. For all routers except $R4$, this rule will specify that the path must not be advertised, such that only customer AS 20 will finally receive that BGP path.

Besides filtering advertisements to neighbors depending on the business relationship, an AS will also prefer some paths over others. As it is cheaper to send traffic through a customer than through a provider or a peer, when there are several paths available to a destination, the one going through a customer will be preferred. Similarly, paths via peers are preferred over paths via providers. The Local Preference attribute is well suited to implement such preferences, as it is sufficient to configure inbound filters to set a high Local-Preference on paths from customer, a medium Local-Preference on paths from a peer and a low Local-Preference on paths from a provider. The decision process will then prefer paths with the highest Local Preference.

1.4 Traffic engineering

Traffic engineering consists in influencing routing such that the traffic is distributed across the links in such a way that the cost is minimized. Typically, operators try to spread the traffic depending on link capacities or delays. In BGP, traffic

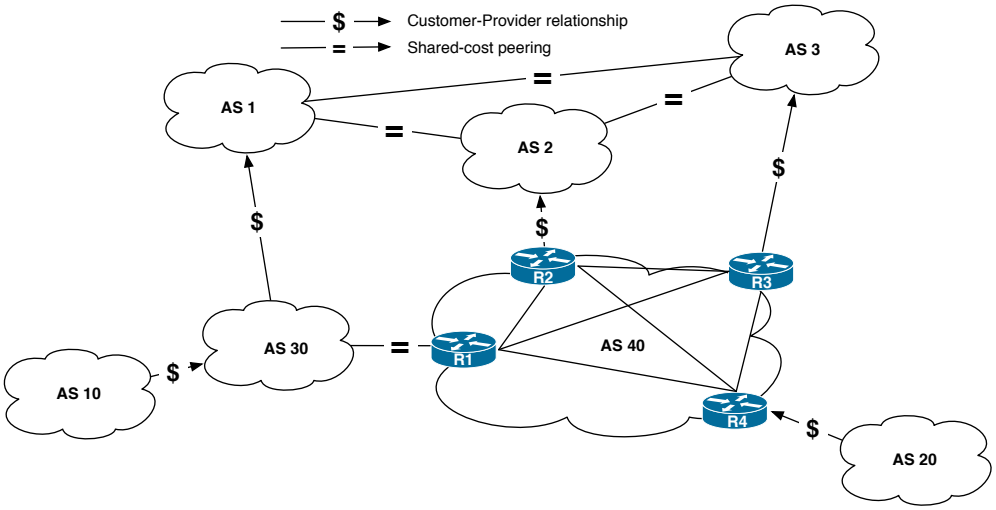


Figure 1.5: Business relationships

engineering can be performed in two directions: for the incoming traffic and for the outgoing traffic [QUP+03] [FBR03].

1.4.1 Outgoing traffic engineering

Outgoing traffic engineering in BGP consists in influencing the choice an egress router for the traffic destined to some destinations. The choice of the egress router is done by the BGP decision process, and as such, an operator wishing to engineer its outgoing traffic will influence this decision process. First, the Local-Preference can be used to prefer some paths over others. For example, if a prefix is reachable via a high bandwidth link and a low-bandwidth link, the first one can be set a higher Local-Preference than the second one. This method is binary: all routers will use the link with the highest Local-Preference, and the low bandwidth link will not be used except in case of failure of the primary. A second method can be used when there are several possible exit points for a given destination that are equally preferred (i.e. same Local-Preference, same AS-Path length and same per-AS lower MED for the corresponding paths). In this case, the IGP distance to the exit point, or BGP nexthop, is used to select the best one. The closest exit point is chosen to minimize the cost of carrying packets through the AS. Outgoing traffic engineering is thus performed through careful design of IGP topology.

1.4.2 Incoming traffic engineering

Performing incoming traffic engineering is more difficult, because the goal is to influence the choice of the router that will receive incoming traffic for some desti-

nations. However, this ingress router is selected by routers of neighboring ASes. A first method consists in advertising selective prefixes or more specific prefixes on different links. For example, in figure 1.6, the stub AS owns prefix $10.1/16$, and can summarize its address space by advertising only this prefix in BGP. However, it wants to engineer its traffic such that all traffic towards the destinations contained in the subprefix $10.1.1/24$ enters the network via Provider 1. Thus, it advertises the more specific prefix $10.1.1/24$ to this provider. Thanks to longest-prefix match, all routers in the Internet will prefer the more specific path via Provider 1 over the less specific path via Provider 2.

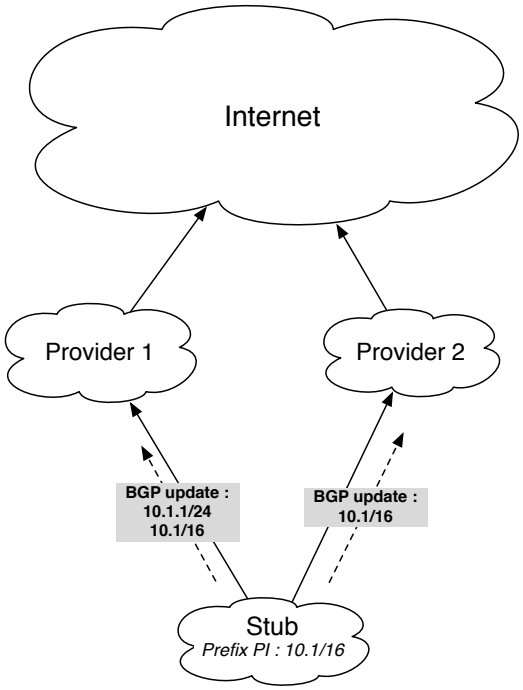


Figure 1.6: Traffic engineering with more specific prefixes

Operators can also change the BGP attributes of the paths that they advertise to their neighbors in order to control their incoming traffic. First, as the AS-Path length is used in the decision process, an AS can artificially increase this length by prepending its AS number several times instead of once. This will decrease the preference of that path compared to others in the decision process of the neighbor AS routers. In the example of figure 1.7, AS1 uses AS-Path Prepending with selective advertisement: It divides its prefix space in two halves, and advertises both more specific prefixes on its two links. On one link, it prepends its AS number twice for the first prefix, and on the other link, for the second prefix. As a result, AS2 will prefer to use the direct path to AS1 for traffic destined to prefix $1.0.0.0/24$ and the path via AS3 for prefix $1.0.1.0/24$. AS3 will go through AS2

for prefix $1.0.0.0/24$ and directly to AS1 for prefix $1.0.1.0/24$. The drawback of this method is that it is difficult to predict the traffic shift resulting from a certain amount of prepending, and operators often need to proceed by trial and error [QPBU05].

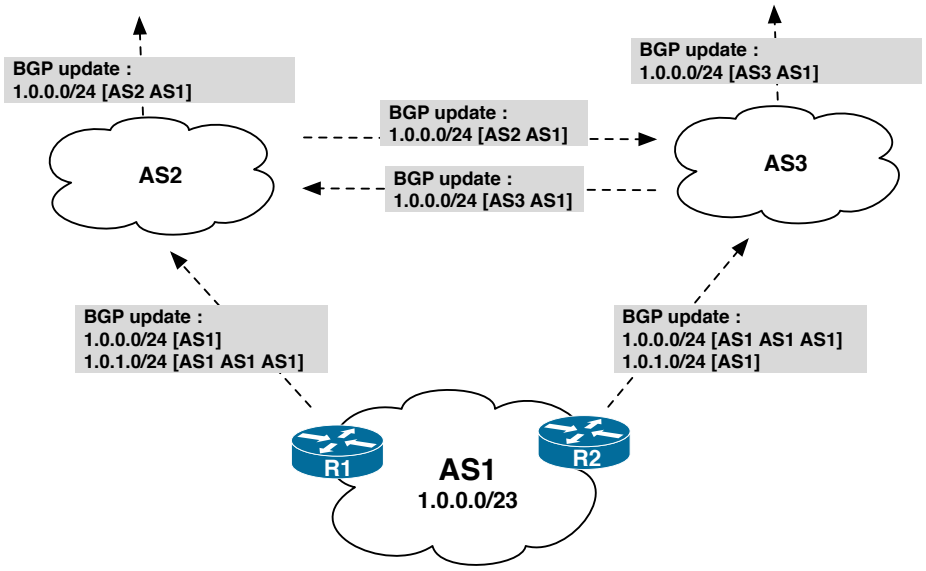


Figure 1.7: Traffic engineering with AS-Path prepending

The MED attribute can also be used to influence the best path selection of a neighbor [QUP+03]. This can be used for example when the Hot Potato policies of two neighbors are conflicting. In the example of figure 1.8, AS 1 is a customer of AS2, and advertises prefix $1.0.0.0/24$ on two eBGP sessions with AS2. Router $R6$ of AS2 will thus receive two paths to that prefix, and choose the one with the closest exit point, i.e. the path $R5 - R3 - R1$. However, AS1 would prefer to receive/send the traffic towards/from this prefix on its cheapest link, i.e. the $R1 - R2$ link. AS1 can thus take an agreement with AS2 such that AS2 enforce its own Hot Potato routing policy using the MED attribute: AS 1 might for example set the MED attribute of the paths it sends on its eBGP session to the IGP cost to the destination. The MED of the path received by $R4$ will be 1, while the MED of the path received by $R5$ will be 100. Thanks to the decision process that will select the path with the lowest MED value, the traffic will flow through the $R2 - R4$ link instead of the $R3 - R5$ link. This is called *Cold Potato Routing*, because the traffic stays longer in AS2 to enter the neighboring AS the closest possible to the destination.

Finally, neighboring ASes can also agree to use communities [CTL96] between each other to control their incoming traffic [BQ03]. A customer may for example

set a specific community on some of its paths to inform its provider that it should set a lower preference on those paths.

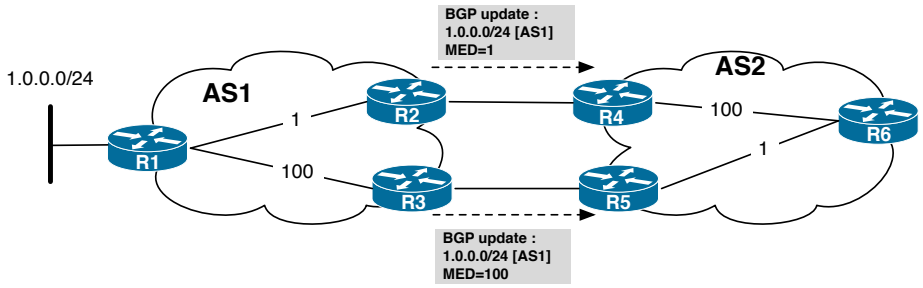


Figure 1.8: Cold Potato Routing

1.5 Conclusion

In this chapter, we have presented the context of the thesis. We have explained the notion of routing, and presented the BGP protocol that is used in the Internet to establish paths between destinations. This protocol provides a wide range of attributes to operators such that they can try to influence incoming or outgoing traffic flows, and implement their routing policies. This flexibility has some drawbacks, and the nature of the Internet structure itself creates concurrency between operators for the establishment of the paths. This results in a complex protocol, whose behavior is sometimes difficult to predict. In the next chapter, we will present a survey of the issues encountered in the Internet because of BGP, then, later in this thesis, we will tackle the challenge of improving interdomain routing stability.

Chapter 2

BGP Challenges

The Border Gateway Protocol is used as the Interdomain Routing Protocol in the Internet for more than twenty years. At the time of its original design, the Internet was totally different compared to what it is today [LCC+09]. The characteristics of the Internet have evolved, in terms of infrastructure, materials and connectivity. The most obvious change is of course its size, but the relationships between actors have also drastically changed. They are now driven by business, and trust cannot be assumed anymore. More and more services are offered, which have to be supported by the network infrastructure. This evolution pushes BGP into its limits, and the protocol as well as the routers must be regularly adapted to be able to route and forward packets to an increasing number of destinations. This chapter presents a survey of the main issues and research topics related to the situation of Interdomain Routing today, i.e. scalability, correctness, misconfiguration and security.

2.1 Scalability

The Internet has grown increasingly since the early days, both in terms of advertised prefixes as in terms of participating Autonomous Systems and routers.

Figure 2.1 [Hus] shows the evolution of the number of AS number advertised in the Internet. While only a few thousands ASes were participating to BGP before 2,000, this number is today higher than 30,000. Even though the number of ASN is not anymore increasing as sharply as during the Internet boom, it is still growing regularly. This increase is an issue, as the BGP AS Number are 16 bits long. With only 65,536 available values, the AS space is indeed close to be exhausted. This lead to a modification of a BGP protocol, such that 32 bits AS numbers are now allowed to support further expansion [VC07].

Figure 2.2 shows the number of BGP entries obtained from the BGP Oregon Route Views collector (AS6447)[Rou]. The evolution of the number of prefixes is similar to the evolution of the AS numbers, with a sharp increase during the Internet boom, then a slowdown around 2001. The figure reveals that today's routers must be able to forward packets to more than 300,000 Internet prefixes.

Traffic exchanged in the Internet is huge, and the performance of the routers must evolve accordingly in two aspects: From the dataplane viewpoint, they have to be able to forward packets at faster rates. They must be able to parse IP headers and determine the interface on which packets have to be forwarded in a few nanoseconds. Their Forwarding Information table must thus be optimized to perform quick best-match prefix searches, even in presence of a number of reachable prefixes exceeding 300,000. From the control-plane viewpoint, routers have to be able to support the flow of BGP messages related to this increasing number of reachable destinations. This flow and the size of the routing state to maintain are also proportional to the interconnection degree of the router, which may also evolve with the size of the network. BGP is under constant pressure due to the huge number of destination prefixes it has to manage.

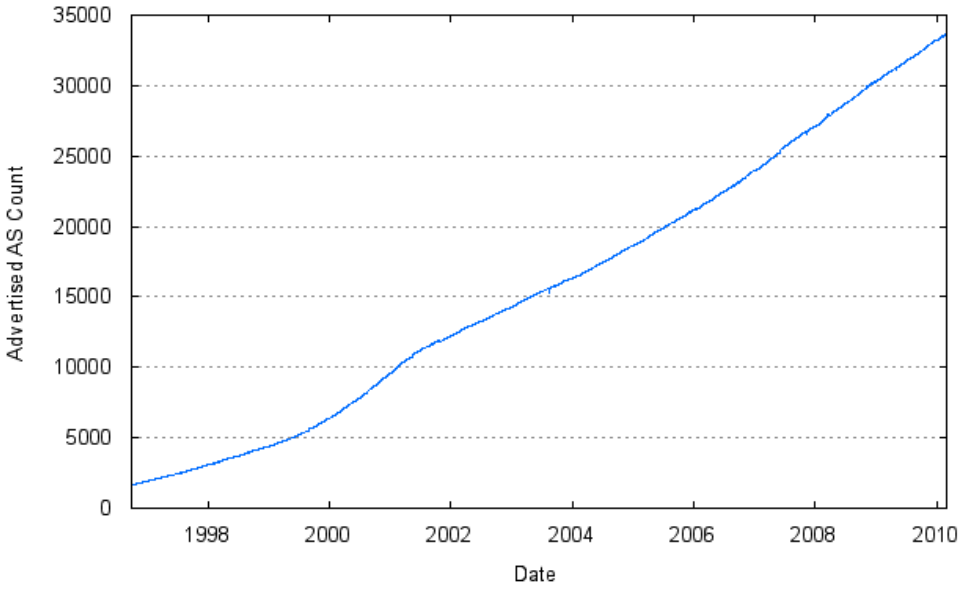


Figure 2.1: Evolution of the number of advertised AS numbers
(source: http://bgp.potaroo.net)

In this section, we review the two visible consequences of the increase of the Internet size: The number of prefixes that have to be maintained in the routing tables and the number of BGP messages that are exchanged to ensure the reachability of all those prefixes. We explore and discuss the related scalability issues and the solutions that have been proposed to solve them.

2.1.1 Routing table size

The first scalability challenge is the evolution of the number of prefixes advertised in BGP (figure 2.2). This number of prefixes is itself dependent upon several factors.

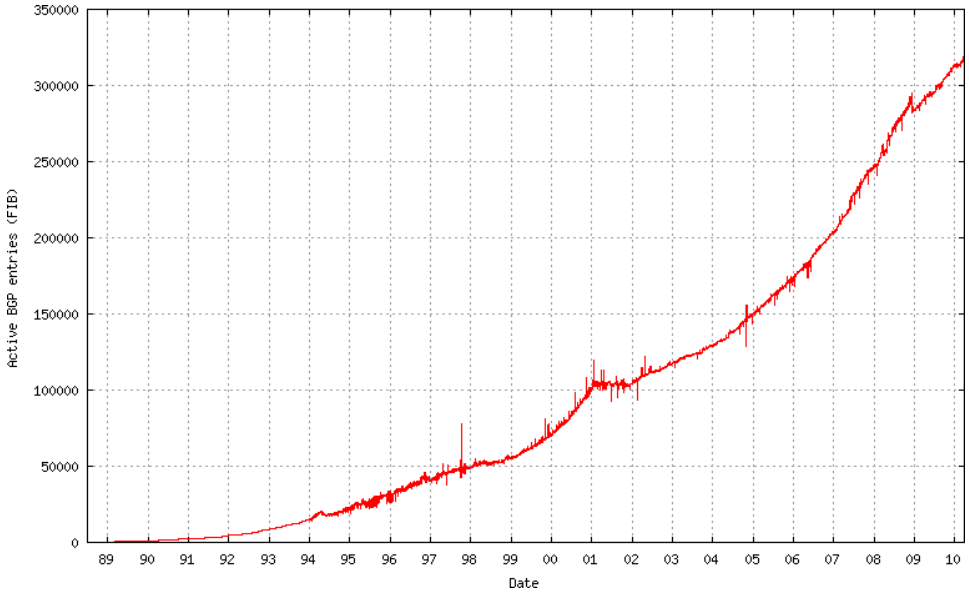


Figure 2.2: Evolution of the number of active BGP entries
(source: http://bgp.potaroo.net)

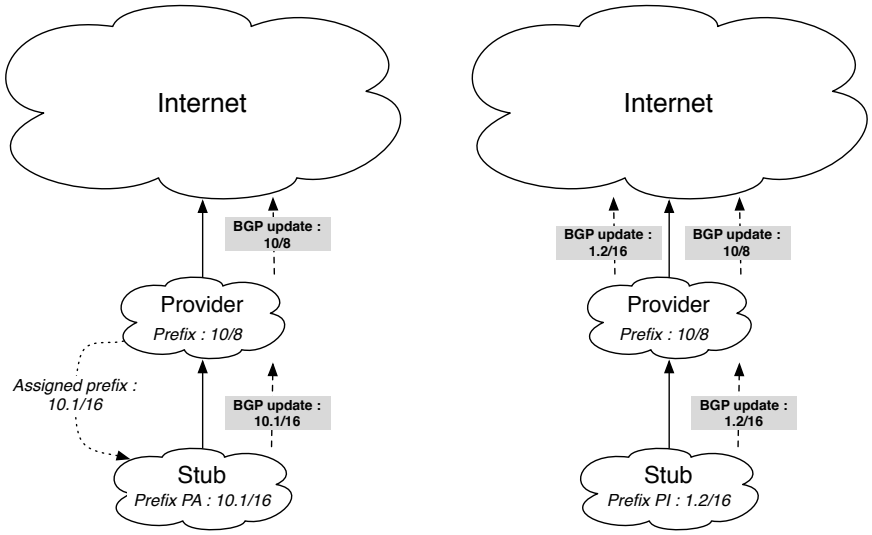
Number of reachable IP destinations

First, of course, the number of reachable IP destinations is increasing. However, the authors of [BGT04] showed that, from 1998 to 2002, the size of the routing table increased by more than 100% while the number of destinations reachable in that routing table only increased by 25%. This suggests that the increase in the number of reachable destinations is not the main factor to explain the routing table growth.

Prefix aggregation and multihoming

Thanks to CIDR aggregation [FL06], sets of destination can be summarized in a common IP prefix of any length, while only four prefix lengths were previously used [Pos81]. The introduction of CIDR in 1994 allowed to maintain a linear growth of the BGP routing table until 1998, while other metrics of Internet size grew exponentially [Hus01]. Unfortunately, after 1998, the growth rate was again increasing, as shown in figure 2.2.

The granularity of CIDR prefixes is thus a second factor for the scalability. Indeed, depending on the assignment of IP adresses, prefix aggregation can be performed to reduce the routing state. For example, if a stub AS wants to connect to the Internet, it will need a set of IP adresses for its machines to be reachable, and a peering contract with a provider that will provide him connectivity to the Internet. The stub may ask its provider to give him a subset of its IP adresses (Provider



(a) Provider-Assigned prefix (b) Provider-Independent prefix

Figure 2.3: Prefix aggregation

Assigned (PA) prefix) as in figure 2.3(a), or ask a Routing Internet Registry to give him a brand new prefix, independently of the provider (Provider Independent (PI) prefix) [KBC+06], as shown in figure 2.3(b). As PA adresses are a subset of the provider prefix, the provider advertises the adresses of the stub while advertising its whole prefix. A single BGP entry is thus sufficient for advertising the destinations of both the stub and the provider. With PI adresses, the prefix of the stub cannot be aggregated with the prefix of its provider, and two BGP entries are then necessary. CIDR has thus a positive impact on the scalability when PA adresses are used.

However, even with PA adresses, customer prefixes are not always aggregated with the provider prefixes. Multihoming is one reason for that [ALD+05]. When a stub is connected to two providers for robustness purposes, its prefix is only aggregatable by the provider that provides the prefix. The other provider will have to advertise the prefix from the stub along with its own prefix, and thus, the small prefix is also propagated in the Internet (figure 2.4). Furthermore, if the stub wants to ensure that the traffic arrives via both providers, the primary provider might also need to advertise the more specific prefix along with the aggregated prefix to compete with the more specific prefix advertised by the second provider.

Sometimes, Autonomous Systems fail to aggregate their prefixes because of pre-CIDR practices. Such ASes might for example advertise two /16 prefixes (class C) instead of one /15 prefix. It might also be impossible for them to perform aggregation, because the prefixes assigned to the AS might not be contiguous, and are thus not aggregatable. This is typically the case when a provider has exhausted its allocated address space, and the RIR was not able to assign him an additional

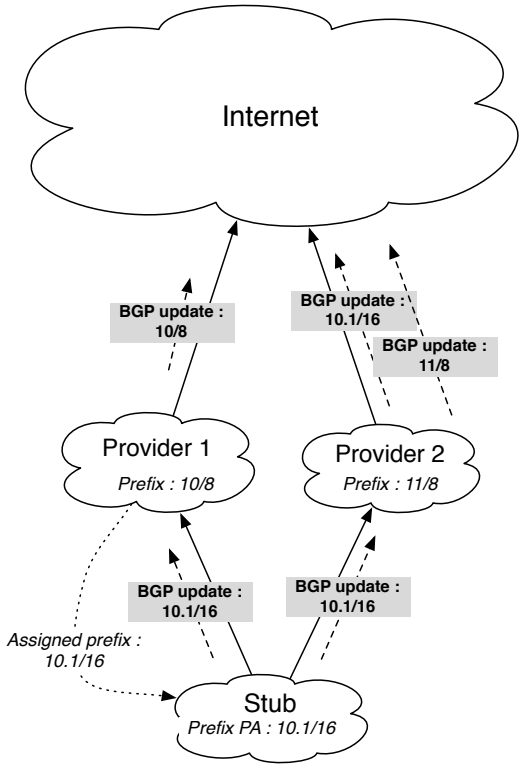


Figure 2.4: Multihoming

contiguous prefix. In their analysis of the routing table growth, the authors of [BGT04] show that address fragmentation is the factor that contributes most to routing table growth.

Traffic engineering

Traffic engineering is another reason for de-aggregating prefixes, as explained in chapter 1: Thanks to IP longest-prefix match forwarding, an AS can balance its traffic across its links by advertising different sub-prefixes on them, along with the fully aggregated prefix that serves as a backup route [QUP+03] [MZF07].

Security

Security might also be a cause of prefix de-aggregation: Advertising the longest possible prefixes limits the ability of attackers to hijack important destinations [RIS08] [MZF07]. Prefix hijacking occurs when some Autonomous Systems starts advertising a prefix that it does not own. Depending on the length of the hijacked prefix and thanks to the longest-prefix match, a variable part of the Internet will

start routing the packet destined to the hijacked prefix to the attacker's network. A network may try to defend himself by advertising the longest possible prefix to its most important destinations.

Dealing with routing table size

The growth of the routing table size has several implications for the routers. First, the memory size of the Routing Information Base of a router should be sufficient to hold all prefixes advertised by the peers of the router. However, the RIB is part of the control-plane, and as such, its memory can be extended by adding RAM to the router. The speed of memory access to the RIB is important to support processing BGP Updates arriving at increasing rates. However, scaling the RIB is probably not as critical as scaling the Forwarding Information Base, which stores the information needed by the dataplane to efficiently forward packets and needs access time small enough to support high speed packet forwarding. Thus, the scalability bottleneck is probably the number of prefixes in the FIB, instead of the number of entries in the RIB (Adj-RIB-Ins and Adj-RIB-Out included).

A first action that can be taken to limit the size of the routing table is to limit the length of the prefixes accepted by a BGP router to 24. This can be implemented by inbound filters, such that all prefixes longer than /24 received on eBGP sessions are discarded, saving memory for both the FIB and the RIB [BBGR01]. The limitation of this scheme is that it impairs the reachability of legitimate long prefixes when there isn't any less specific prefix available to forward the corresponding traffic [Hug04]. Operators may also filter invalid (bogon) prefixes to reduce the size of their routing tables and for security purpose. Those prefixes are not-yet-allocated prefixes, and IP addresses called "martian prefixes", i.e. private and reserved IP addresses [BBGR01].

Another way to reduce the FIB is Virtual Aggregation [BFCW09]. In this case, the global routing table is spread across several routers in the AS, and the packets are forwarded to the router knowing the route to the corresponding routing prefix.

Finally, starting from the observation that current Internet uses IP adresses both to identify and to locate any host, the Locator/ID Separation Protocol (LISP) has been proposed as a solution to Internet scalability [Mey08]. With LISP, only the locators (i.e. router's addresses) are advertised in the routing system, while end-host identifier are via a separated mapping mechanism. Routing scalability is ensured thanks to locator aggregation.

2.1.2 Churn

The term *Churn* in the context of interdomain routing refers to the high number of BGP messages that can be seen in the Internet. By essence, BGP is a very quiet protocol if the network is stable, as incremental BGP Updates are generated only upon routing change. However, studies of BGP activity in the global Internet have shown that the number of BGP messages exchanged is really important. Because

all those BGP messages have to be processed, BGP churn causes high-CPU load on smaller routers.

[HA06] reports a rate of up to 500 thousands BGP Updates per day by the end of 2005. Furthermore, the number of BGP messages exchanged is growing [HA06] [EKD10] [EKD08], even though the exact extent of this growth is still unclear. [HA06] mention a growth of 200% of the raw churn for the single 2005 year, while after deep investigation and filtering of churn components, the authors of [EKD10] show that the baseline churn rate grew by only 20-80% from 2003 to 2008, which is less than the corresponding growth of the routing tables.

Events resulting in BGP messages can be classified in two categories:

1. **Reachability events**: Those events relate to the reachability of destinations. They reflect legitimate dynamics of the network, i.e. destinations that become physically reachable or unreachable, or topology changes. Paths changes following a link failure also belong to this category.

2. **Pathological events**: The events that are consecutive to pathologies in the network, or to routing artefacts. Such an event is for example a link failure which does not disconnect destinations but results in transient reachability losses, or implementation issues in the routers leading to unnecessary BGP message propagation.

BGP activity of the first category is related to the instability of the Internet infrastructure, i.e. destination changes and link or node failures that disconnect a part of the network.

As BGP messages related to events of the first category reflect variations in reachability, it is thus the role of BGP to propagate these changes. The second category, however, concerns BGP messages that do not directly relate to the reachability of destinations. This pathological churn has an impact on the routing and forwarding plane, as it increases the load of the routers. More important, the convergence for those events can lead to packet losses as some routers can reach a state where they do not have a path to the destination even though it is still reachable.

In this section, we explore the causes of pathological BGP message exchanges, and for each, we review the solutions proposed to reduce the related churn.

Flapping links

Some interdomain links are unstable and fail frequently (flapping links) [BFF07] [WMRW05] [WGWQ05]. Depending on the location of the link, this can lead to reachability loss or not. In both cases, their frequency lead to pathological churn, as each of these failures causes the transmission of a number of BGP Withdraw messages, followed by the corresponding BGP Updates upon recovery of the failure. Prefixes impacted by those failures are thus responsible for a lot of BGP messages. It has been shown that the majority of BGP instability was caused by a small number of prefixes, and that there was not a lot of traffic to and from those destination.

Actually, the prefixes that carry most of the Internet traffic are remarkably stable [RWXZ02].

Mechanisms have been proposed to tamper the churn related to flapping links. The Route Flap Damping mechanism [VCG98] dampens paths that are too frequently advertised then withdrawn, then re-advertised, and so on. When the frequency of path change exceed the limit, the path is no longer accepted by the router. However, Route Flap Damping is difficult to configure with proper values. [MGVK02] shows that Route Flap Damping can delay the convergence of stable prefix in case of occasional failure, simply because transient routing states are explored and lead to several BGP messages for the same event. It is not recommended to use this mechanism anymore [SP06].

Implementation issues

In 1997, measurements of Internet churn [LMJ97] revealed that the number of BGP Updates exchanged per day was one or more orders of magnitude higher than expected, up to several millions of prefix updates per day. Most of those BGP messages are classified as pathological, and simple modifications of router implementations succeeded in reducing BGP churn by one order of magnitude by 1999 [LMJ99]. The problem was that most implementations of BGP were stateless (i.e. without Adj-RIB-Outs). Thus, whenever a router changed its best path selection, it propagated BGP Updates even in case of change in local attributes. Adding (at least partial) state to BGP routers had a dramatic impact on BGP churn. A more recent study about churn shows that today, the Internet is healthier and that duplicates BGP messages only account for 15% of BGP traffic [LGW+07]. However, another analysis reports higher duplicate rate, up to 40% [EKD10]. The origin of today's duplicates has been investigated by [PJL+10], and is attributed to interactions between iBGP and eBGP. [PJL+10] also found that duplicates were responsible for the majority of router processing loads during their busiest time, while higher processing loads can lead to more sessions reset, routing loops and packet losses. Furthermore, [WZP+02] shows that duplicates occurrence is exacerbated when the network is under stress.

Path exploration

Another cause of BGP churn is the path exploration problem, which is inherent to path vector protocols. When a path becomes unavailable, BGP routers will explore several invalid or non-optimal paths before reaching the final state. Each path change during the convergence leads to additional BGP messages, and thus contributes to the churn [OZP+06].

In figure 2.5, $AS1$ advertises prefix $10.0.0.0/8$ to the Internet. As a result, $AS5$ learns 3 paths to that prefix, via $AS2$, $AS3$ and $AS4$, with AS-Paths length of, respectively, 2, 3 and 4. Upon failure of the link between $AS1$ and $AS2$, there are several possible convergence scenario, depending on the processing times

of each BGP message. However, in the worst case, the following sequence may happen: $AS5$ receives a BGP Withdraw from $AS2$, and thus switches on the path via $AS3$ and send a BGP Update accordingly to $AS6$. Then, $AS3$ sends its BGP Withdraw, and $AS5$ switches on the path via $AS4$, with a second BGP Update to $AS6$. Finally, $AS4$ withdraws its path, and $AS5$ sends the BGP Withdraw to $AS6$. In this situation, instead of withdrawing the prefix upon notification of the failure by $AS2$, $AS5$ explores the two other paths even though they are also invalid. Two unnecessary Updates were sent to $AS6$, and the convergence was delayed by this path exploration.

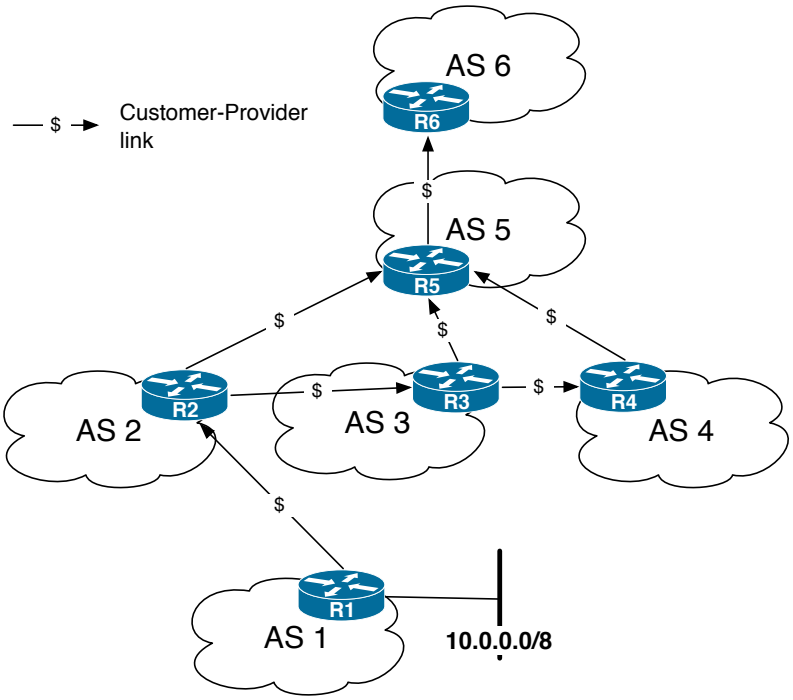


Figure 2.5: Path exploration

A first technique to reduce the churn of path exploration is to use the Minimum Rate Advertisement Interval (MRAI) timer [RLH06]. This timer is designed to insert a minimum delay between two advertisements to the same neighbor or for the same prefix. Thus, all transient advertisements that would have occurred before the timer elapses are replaced by the final path advertisement. Churn is reduced, but the counterpart of this mechanism is that the convergence is delayed [LABJ01]. Choosing a proper MRAI timer value is difficult, and is actually dependent on the part of the network where it is deployed [GP01].

Lambert et al propose another timer mechanism, called MRPC timer [LBU09], that takes the path preferences into account before computing the timer delay. This

way, path exploration can be prevented, and the BGP convergence is improved by a proper ordering of the messages.

Ghost Flushing [BBAS03] has also been proposed to reduce path exploration, by removing invalid paths that are transiently explored. By propagating bad news (i.e. path failure) quicker than good news (i.e. candidate paths), the convergence process avoid intermediate state. EPIC, RCN and other mechanisms propose to insert route cause event information inside BGP messages, such that routers can discard all paths impacted by the failure before receiving the corresponding Withdraw [CDZK05] [PAMZ05] [LCR+07].

Finally, as path exploration is inherent to BGP convergence, a simple way to prevent the related churn is to prevent the convergence itself. Upon link failure, convergence is needed for two reasons: First, to invalidate the failed path, and second, to find an alternate path to the destination. If the alternate path is readily available inside routers, the convergence is limited to invalidate the failed path in all routers that use it such that they switch on their alternate path. In this case, only BGP Updates messages with the new best path will be sent to the neighbors, except if policies do not allow this advertisement.

However, if the alternate path is not available upon reception of the withdrawal for the path, which is unfortunately often the case [WWGQ09] [UT06], the router looses its ability to forward packet to the destination until it receives the alternate path from another router. Thus, instead of advertising a new best path, routers will transiently send BGP Withdraw messages to their neighbors. If the neighbor is also lacking an alternate path, it will propagate the Withdraw as well, and so on until some router is able to advertise the alternate path. Withdraw-related churn is thus even worse, as it is more likely to disrupt data forwarding in routers lacking alternate paths.

2.2 Convergence issues

Along with churn and scalability concerns, BGP also suffers from convergence issues. We have already mentioned that, in case of link failure without disconnection of the network, some routers may transiently loose reachability to the destination even though alternate paths are available. But convergence in itself is not always guaranteed, and routing oscillations may occur under certain circumstances.

2.2.1 Transient interdomain failures

Interdomain links and eBGP sessions fail frequently [BFF05] [WSGP07]. However, disconnecting a destination from the Internet because of such a failure is a rare event. Except with single-connected stub, several links must fail in order to render a destination unreachable. Thus, most of the time, the convergence following a single-link failure consists in switching to an alternate path to the impacted destination such that traffic can reach it. However, as mentioned in the above discussion

about churn, the lack of alternate paths inside routers delays the convergence, as routers must exchange BGP messages in order to obtain the alternate path. Even when a link comes up again, some routers in the Internet can reach a transient state where they have no more path to the destination [WWGQ09]. This has a serious impact on the dataplane performance [KKK07] [WMW+06], as packets are dropped as long as there is no path to their destination in the router. It is thus crucial to limit the convergence time in case of failures. Zhang et al. insist on the fact that maximizing packet delivery requires to reduce convergence delay AND to mask transient failures to the rest of the network [ZMZ04]. This can be performed by providing sufficient path diversity to reroute the traffic until BGP has converged.

To this end, Wang et al. propose a new protocol in which routers are allowed to advertise non-best paths as alternate paths for their neighbors [WG09]. The protocol is incrementally deployable with BGP, but requires the cooperation between neighboring ASes. The R-BGP [KKKM07] proposal also relies on precomputed recovery paths for fast recovery, and is able to use information about the cause of the failure to prevent forwarding loops. However, it only uses path diversity at the AS-level, not at the router level. In [LGGZ08], the AS-level diversity is similarly used to compute complementary paths, i.e. paths that are not affected by the same events. Bonaventure et al. propose a way to reroute the traffic through a tunnel to another link in case of link failure [BFF05]. This allows to protect neighbors on-demand, as an additional service. A Routing Control Platform [CCF+05] can also be used by ISPs to compute backup paths on behalf on routers, but such centralized route servers must be carefully designed with robustness in mind, because in face of failure, routers won't be able to get their routing information anymore. Another approach presented in [SMS06] consists in differentiating the processing of BGP Updates by setting a higher priority on the BGP Updates that have an impact on the best path selection. This reduces convergence time, and thus reduces the duration of failures.

However, those solution are not currently largely deployed, and transient routing failures are still an important issue to consider.

2.2.2 Routing oscillations

Delayed convergence in case of failure is not a single issue with BGP convergence. Actually, there are some configuration where the protocol may diverge. The problem has been widely studied, and routing policies have been identified as the root causes of such anomalies. Indeed, policies express the preferences of operators for some paths over others. As those preferences are local to the ISP, the policies of different ISPs can conflict and result in routing oscillations [GW99] [VGE00]. The consequences of routing oscillations are that without stable routing state, routers are not able to correctly perform packet forwarding to the destination.

In the configuration of figure 2.6 presented in [GW99], three ASes are connected to $AS0$ advertising a destination. Due to local policies, each AS prefers the path via its right neighbor over the direct path, as shown in the figure by the list

of AS Paths ordered by preference. Clearly, those policies are conflicting. This leads to routing oscillations: Upon advertisement of the destination, all three ASes choose the direct path, as they do not know other way to join the destination. Then, they advertise this path to their neighbor, and each AS will switch on its right neighbor path. However, as soon as each AS chooses the longest path, it will also withdraw its direct path to its neighbor. Thus, the longest paths cannot be used anymore, and the three ASes have to switch back on their direct path. They start advertising the direct path to each other, and the cycle of advertisement/withdrawal restarts. The authors of [GW99] prove that, whatever the sequence of BGP messages, this system will never reach a stable state.

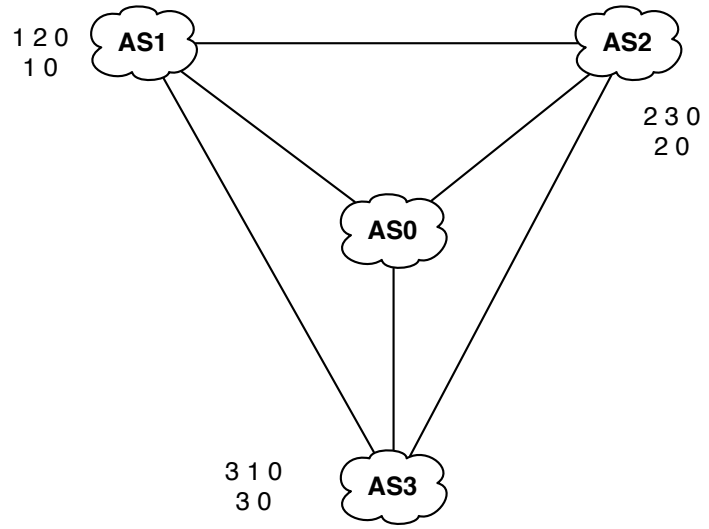


Figure 2.6: Oscillating topology

Analysing if the Internet contains configurations that would result in routing oscillations is difficult. First, a full view of the topology is needed, along with all ISPs policies. However, ISPs are not always keen to announce their routing policies, even if some are published in Internet Routing Registries in RPSL format. Second, detecting if a given configuration will result in routing oscillations is a NP-hard problem. Finally, even though no oscillation is present during normal operation, they can arise upon link failure. Indeed, in some stable configuration, removing one link may result in a topology similar to the one presented in figure 2.6, and routing oscillations could appear.

Fortunately, in practice, policies are often driven by business considerations, as explained in chapter 1: ISPs do not provide transit for the traffic they are paying for (i.e. from providers or peers), and usually prefer the paths from their customers, for which they are paid for. Gao and Rexford have shown that routing oscillations are prevented if all ISPs follows those design guidelines for their policies [GR01].

However, given the expressiveness of BGP, ISPs are not limited to those classical policies, and there is no guarantee that routing oscillation can not occur. Route Flap Dampening can be used to mitigate the effects of routing oscillations. However, it only slows down the convergence [GW02a], while it would be definitely more efficient to identify or remove the cause of the oscillation. Furthermore, Route Flap Damping has been shown to impact the propagation of legitimate routing events [MGVK02]. Another solution is to modify BGP with a new attribute registering the history of the AS-Path [GW00]. This would allow to dynamically detect and suppress routing oscillations.

In addition to those inter-ISPs routing oscillations, there can also exist routing oscillations inside ISPs, i.e. between routers belonging to the same AS [GW02b] [GW02a]. While the formers were related to conflicting routing policies between ASes, the latters are caused by conflicting path preferences between routers. In addition to oscillations, iBGP also suffers from other routing anomalies: Even if a stable routing state is reached, the best path choices of the routers can also conflict and lead to forwarding deflection, or worse, forwarding loops. The MED attribute and the use of Route Reflection, in particular, are responsible for those instabilities. We will detail those issues in the next chapter.

2.2.3 Non-deterministic convergence

Even when BGP convergence is stable, there exist some configurations where its outcome cannot be predicted, i.e. there are several stable routing states that can be reached depending on the timing of BGP messages [GW99].

On figure 2.7, $AS1$ and $AS2$ both prefer to reach the destination advertised by $AS0$ through each other instead of using the direct path. If $AS1$ advertises the path 10 to $AS2$ first, $AS2$ will choose that path as best and never advertise path 20 to $AS1$. Thus, $AS1$ is forced to use the direct path because it does not learn about the other one. If $AS2$ is the first AS to advertise its direct path, the result is symmetric: $AS2$ cannot use the path through $AS1$.

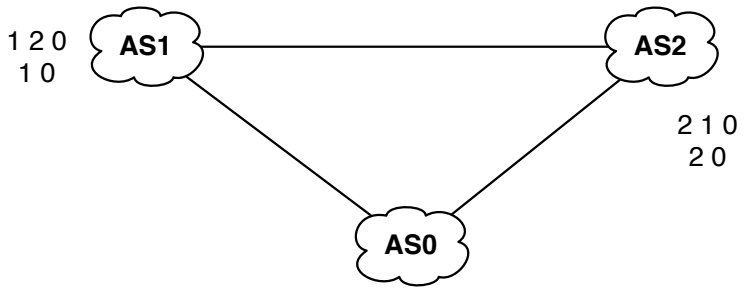


Figure 2.7: Non-deterministic topology

This situation is problematic because it can impair some routing policies. Such configurations are called *BGP Wedgies* [GH05]. In the example of figure 2.8, $AS0$ has two providers, and prefers to use $AS3$ as primary and $AS1$ as backup. Communities can be used to implement this policy: $AS0$ advertises its prefixes to $AS1$ with a backup community, such that $AS1$ will prefer other paths over this one. However, the intended outcome can only be obtained if $AS0$ advertises its prefixes to $AS3$ before advertising them to the backup provider. Indeed, if $AS2$ does not know the path 230 upon reception of the backup path from $AS0$, it will propagate the path 20 to $AS3$. $AS3$ will be able to choose between the two paths 10 and 30, and will prefer the path via $AS1$ because it prefers paths from customers over paths from peers. $AS1$ never learns about the primary path, and the traffic from both $AS1$ and $AS2$ flows through the backup link instead of the primary link.

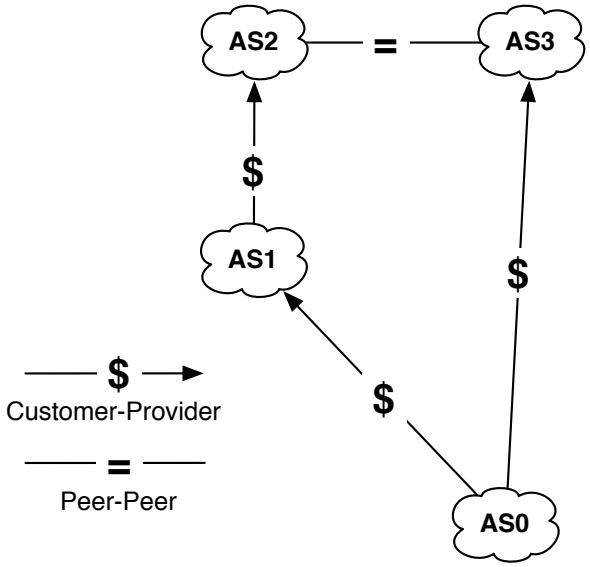


Figure 2.8: Non-deterministic backup policy implementation

Even if the initial convergence allowed to reach the desired outcome, the system can switch on another solution upon link failure. With a correct backup policy implementation in figure 2.7, $AS2$ only knows the path from $AS3$ while $AS1$ is aware of both path. All traffic flows along path 1230. If the primary link between $AS0$ and $AS3$ fails, $AS1$ advertises the backup path to $AS2$ to allow traffic to be rerouted through the backup link. However, as $AS2$ now knows about the 10 path, it will continue to forward its traffic through its customer even when the primary link comes back. The policies of $AS0$ cannot be enforced anymore, except by explicitly withdrawing the backup path then re-advertising it.

2.3 Misconfiguration

Configuring BGP routers is a crucial task as proper behavior of the network depends upon it. Routers must be configured regularly, when new devices enters the network, when new services are added, or when new customers subscribe to the services of the ISP [EMS+07][CGG+04]. Network configuration is a complex task, as lots of features must be set up, and configuration languages are often very low-level. Furthermore, those langages are vendor-dependent, sometimes even version-dependent. The configuration task scales with the number of routers, and configuration files of large networks may contain millions of lines [FB05].

This complex configuration task is usually performed by human operators, and is thus error-prone. Several studies have measured the impact of misconfigurations. [MWA02] analysed BGP feeds, and reports up to 1200 prefixes suffering misconfiguration each day. The authors of [FB05] performed static analysis of the configuration of 17 ASes, and were able to identify more than a thousand faults. In [ZPW+01], misconfiguration is identified as a cause for Multiple Origin AS conflicts. Some forwarding loops were also attributed to misconfiguration [XGF07].

There can be several errors when configuring BGP routers. First, BGP filters can be incorrectly set up, leading the AS to advertise or to accept BGP paths that do not conform to its policies. Second, prefix advertisement can be erroneous, with advertisement of non legitimate prefix (hijacking) or leaking of private network prefixes. Consequences are multiple [FB05] [MWA02]. Policies and peering contract are violated [FMR04], additional BGP messages are exchanged, increasing global Internet churn, and connectivity is impaired [MWA02]. Several examples of large Internet disruptions due to BGP misconfiguration exists. In 1997, AS7007 advertised most of Internet prefixes [nan97] and attracted all the related traffic. In 2008, a Pakistani ISP advertised of a subset of YouTube IP addresses including its DNS servers, resulting in YouTube becoming unreachable from part of the Internet [RIS08]. More recently, in April 2010, a small Chinese ISP also leaked a (probably) full Routing Table into BGP, and its providers propagated the about 10 percent of those hijacked BGP advertisements in the Internet, including the IP addresses of a set of highly popular websites.

Multiple tools have been proposed to help operators in the network configuration task. For the configuration of BGP, two approaches are possible. [FB05] proposes the RCC tool (Router Configuration Checker), which is able to detect configuration errors from existing configuration files. Another solution is to use high level description of the network and policies, and to automatically generate the configuration files. This approach is proposed in [GGRW03] to automate the provisioning of BGP customers. PRESTO [EMS+07] uses small pieces of configuration (configlets) to represents different services or service options, and the whole network configuration is build by composing those configlets. NCGuard [VPB08][VQB09] goes even further with a higher-level network description language allowing to describe BGP policies, combined with a tool that validates and generates the corresponding network configuration files.

2.4 Security

Security is of increasing importance, with malicious activities being more prevalent in today's Internet. Some actions can be taken to improve the security of BGP. Route filtering must be properly configured to deny all invalid prefixes, i.e. unallocated, private and reserved IP prefixes. This allows to limit *bogon* prefix advertisement, and it has been shown that some Denial of Service attacks or spam campaign were originated from such prefixes [cym] [FJB05]. However, such filters must be regularly updated to avoid filtering newly allocated prefixes [FJB05].

BGP is not a secure protocol by itself, in the sense that any router can advertise any IP prefix. BGP does not provide any mechanism allowing to check whether a BGP advertisement is legitimate. This opens the door to *prefix hijacking*, either intentionally or due to misconfiguration, as mentioned in the previous section. When advertising a prefix that it does not own, a router attracts all traffic destined to that prefix. The prefix will likely experience reachability problems, and it is difficult for operators to identify the cause of such disruptions. Hijacked prefixes can also be used to carry out malicious activities using someone else identity.

Several approaches are proposed to fight prefix hijacking. A first solution consists in preventing hijacking by securing BGP advertisement with Public Key Infrastructure (see SoBGP [Ng04] and S-BGP [KLS00]), but the resulting computational overhead is large and they require to modify the protocol. At the IETF, the SIDR working group A more basic protection against prefix hijacking is that providers should filter out all BGP advertisements from their customers that does not concern the customers prefixes.

Reactive solutions have also been proposed. First, prefix hijacking must be detected. Such a detection can be dataplane or control-plane based. Dataplane detection is performed by active probing [ZZH+08] [ZJP+07], while control-plane detection relies on passive monitoring of BGP feeds and use of Internet registries [SF07]. The accuracy of hijacking detection is however often limited by data sources and the number of vantage points. Second, when prefix hijacking is detected, measures must be taken to counter the attack. First, manual action can be taken, by filtering malicious paths or advertising more specific prefixes, but automated reactive mitigation responses are also possible. Pretty Good BGP [KFR06] mitigates attacks by delaying new, suspicious BGP advertisements based on BGP historical data. A suspicious advertisement is for example a prefix that is suddenly advertised by a new AS while having been advertised by another one for a long time. Another mitigation solution [ZZHM07] consists in purging bogus paths and promoting valid paths by shortening their AS-Path.

Thus, solutions exist against prefix hijacking. While some are difficult to deploy, such as So-BGP or S-BGP, others are designed to be incrementally deployed, and can even be effective with a small number of participating ASes [ZZHM07]. At the IETF, the Secure Inter-Domain Routing Working Group (SIDR) is working on defining an architecture for an interdomain security framework [sid].

2.5 Conclusion

In this chapter, we have explored the several aspects of Interdomain Routing which are of concern today. First, the number of entries in the routing tables increases, and raises scalability challenges for the infrastructure as routers must be able to maintain all those destinations and manage the resulting control-plane flows. Second, the protocol in itself suffers from convergence issues under some circumstances, leading to routing oscillations, routing inconsistencies and forwarding loops. And finally, human intervention can impair the behavior of BGP, either because of router misconfiguration, or via malicious activities on the protocol. There is thus still a large amount of work for researchers to improve interdomain routing such that it will be able to support the Internet evolution.

Chapter 3

Enlightening iBGP

In the previous chapter, we explored the challenges that the Internet is facing today. Scalability, correctness, misconfiguration and security are serious problems, but it is difficult to address them globally. Indeed, the Internet is a distributed system, and the participating entities (ASes) are often competitors and unlikely to collaborate without sufficient financial incentive. The interest of operators is focused on their network, not on the global Internet. However, an Autonomous System is in itself a subset of the Internet, with several BGP routers interacting with each other.

In this chapter, we explore the characteristics of BGP from the inside of an AS, and show that scalability is also a concern at this level. Route Reflection has been introduced to improve scalability and manageability. But scaling iBGP with Route Reflection has drawbacks. We show in this chapter that the limited path diversity propagation in iBGP contributes to the Internet instability. This analysis will rely on a survey of existing work on iBGP, and be supported by several measurement results.

3.1 In search for a scalable iBGP with Route Reflection

While Internet scalability mainly depends on the number of participating ASes and reachable prefixes, iBGP scalability is also driven by the number of BGP routers in the AS, and more precisely by the interconnection degree of iBGP routers.

Indeed, the cost of maintaining a BGP session depends on several factors. First, the router must maintain the TCP connection over which the session is established. Second, the router must process all BGP messages received and sent over the session and, even more important, keep track of those BGP messages.

A BGP router has to remember the paths received from all its BGP peers, because they serve as input for the decision process. It must also remember the paths that it sent to its BGP neighbors, in order to compute incremental BGP Updates and avoid duplicate messages. There is thus a memory cost associated with each BGP session, which will be proportional to the number of destinations learned and advertised by the router on that session.

	Number of sessions	Max. number of paths in Adj-RIB-Ins
Full-Mesh	$\#Routers - 1$	$Size_{RT} * (\#Routers - 1)$
Route Reflector	$(\#RRs - 1)$ $+(\#clients)$	$((\#RRs - 1) + (\#clients))$ $.Size_{RT}$
RR Client	2	$Size_{RT}.2$

Table 3.1: Number of sessions and number of iBGP paths in a router. $Size_{RT}$ is the number of prefixes in the routing table, $\#Routers$ the number of routers in the AS, $\#RRs$ the number of Route Reflectors in the iBGP organizations, and $\#clients$ the number of clients of the Route Reflector.

Finally, all iBGP sessions must be configured, and this task is often performed manually. Several researchers have studied the faults that affect large IP networks [MWA02]. These studies show that configuration errors are responsible for a large number of failures. Configuration errors on iBGP sessions are of course more likely to occur in a network having to maintain a large number of those sessions.

For all those reasons, the scalability of an iBGP organization thus depends on the number of iBGP sessions that are established and on the number of prefixes advertised on those sessions. The complexity of the iBGP organization also has an impact on configuration easiness.

In the case of a Full Mesh of iBGP sessions, the number of sessions to be maintained by each router is large: it is equal to $n - 1$, n being the number of BGP routers in the AS. Thus, to reduce the burden of maintaining all Full Mesh routing state, Route Reflectors [BCC00] have been introduced to limit the number of iBGP neighbors of ASBRs. As explained in chapter 1, an ASBR will typically maintain iBGP sessions with only two Route Reflectors. This is clearly more scalable than an iBGP Full Mesh, because they only have to store at most twice the number of available prefixes. Route Reflectors have to maintain one iBGP session with each client, and one iBGP session with each other Route Reflector. Table 3.1 summarizes the ressources needed by a Full Mesh and by an organization with Route Reflection. Even though smaller than with a Full Mesh, the cost for a Route Reflector is higher than for a plain ASBR, but Route Reflectors are typically larger routers with sufficient memory to support Route Reflection overhead.

3.1.1 Case study

Throughout this thesis, we will perform some analysis on the data of three different Internet Service Providers. One is GEANT, the pan-european research network, while the two other are large Tier-1 ISPs. We will name them ISP A and ISP B. The characteristics of those networks are summarized in table 3.2.

ISP	# of routers	iBGP org.	# of sessions	# of paths	# of prefixes	Method to obtain eBGP Paths
GEANT	23	Full-Mesh	253	655,021	154,668	iBGP collector
ISP A	108	2-level RR	363	987,012	162,666	Adj-RIB-Ins dump of 5 top-level RRs
ISP B	-	1-level RR	-	1,795,268	281,587	Adj-RIB-Ins dump of 14 RRs

Table 3.2: Characteristics of the three ISPs

Category	Number of routers
Level 0 (top-level RRs)	16
Level 1 RRs	57
Level 2 (clients only)	32

Table 3.3: iBGP organization of ISP A

GEANT

The data used for the GEANT network was collected on February 2005. At that time, there was 23 routers all connected to each other in a Full-Mesh of iBGP sessions, resulting in 253 iBGP sessions. Those border routers had 64 eBGP sessions in total, with 41 different neighboring ASes. Data about BGP paths were collected using a workstation running a software implementation of BGP (GNU Zebra), and maintaining iBGP sessions with 22 of the 23 border routers. It was thus able to collect the best paths of those 22 routers. The available snapshot of those paths contains 655,021 paths to 154,668 different prefixes.

ISP A

The first of the two Tier-1 ISPs is the one for which we have the more data available. We have information about the full BGP configuration of january 2006, with 108 border routers, spread on a hierarchy with two levels of Route Reflectors, as shown on table 3.3. This results in a total of 363 iBGP sessions. The ISP A peers with 217 neighboring ASes, exchanging BGP paths over 420 eBGP sessions.

The data about BGP paths available for this network were obtained by dumping the Adj-RIB-Ins of five top-level Route Reflectors. This results in 987,012 paths to 162,666 different destinations.

	GEANT	ISP A	ISP B
Number of routers	23	108	-
Number of sessions with RR	43	363	-
Number of sessions in a Full-mesh	253	5778	-
Number of paths per RR client with RR	309,336	325,332	563,174
Number of paths per router in a Full-Mesh	655,021	987,010	1,795,268

Table 3.4: Comparison of iBGP organizations in the three ISPs

ISP B

The second Tier-1 ISP is a large network of several hundreds of border routers organized in a Route Reflection hierarchy with a single level of Route Reflectors. We do not dispose of detailed information about the BGP topology nor the eBGP peerings, but we have a dump of the Adj-RIB-Ins of 14 Route Reflectors, performed on April 2008. This dump contains 1,795,268 different paths towards 281,587 prefixes.

Cost of iBGP organizations

Based on the data available for those three networks, we can compare the cost in terms of iBGP session and in Adj-RIB-Ins size for each of them. This comparison is summarized in table 3.4. In the case of GEANT, we approximate the cost for Route Reflection by designing an organization with two redundant Route Reflectors having all other routers as clients. We do not have any information to infer the number of sessions in ISP B for each organization.

Concerning Adj-RIB-Ins sizes, for the Full-Mesh, we approximate the number of paths to be maintained by each router as the number of unique paths collected in each network. For Reflectors clients, we approximate the number of paths as twice the number of destinations known by the AS, as each client receives one path for each destination from two Route Reflectors if redundant Route Reflection is used.

Clearly, in the two large ISPs, a Full-Mesh is not scalable. The number of sessions in the network increases by a factor of 20 in ISP A, leading to nearly six thousands of sessions to configure and maintain. Adj-RIB-Ins sizes increases by a factor of three for both Tier-1's RR clients when the iBGP organization is replaced by a Full-Mesh.

We validate our approximation on ISP A by building a C-BGP [QU05] model based on the BGP configuration, and by re-injecting the collected paths into the corresponding border routers. We then run the simulator to replay the BGP convergence, and measure the number of paths in the Adj-RIB-Ins of each router. We perform a similar analysis by replacing the original Route Reflection organization by a Full-Mesh of iBGP sessions. Results are shown in figure 3.1. This figure show the cumulated distribution of Adj-RIB-Ins sizes in terms of number of paths. With a Full-Mesh, all routers have about one million paths, and with Route Reflection,

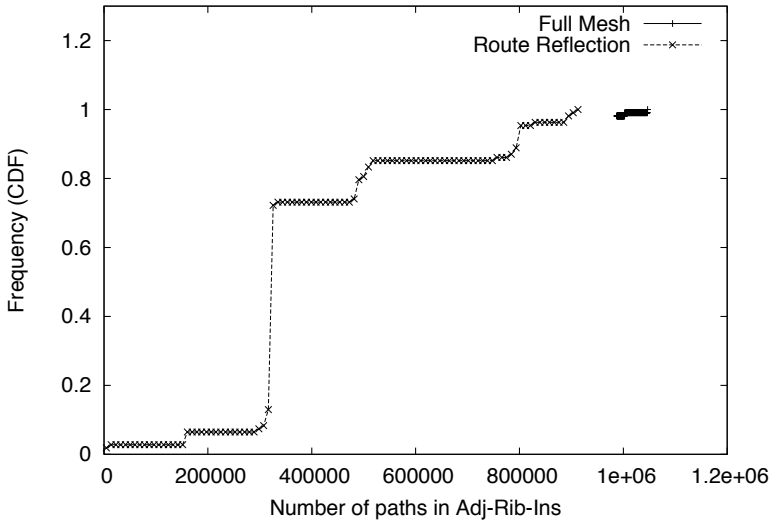


Figure 3.1: Distribution of the number of paths in the Adj-RIB-Ins of ISP A's routers

75% of the routers have 350,000 paths, which corresponds to RR clients receiving two paths each. Table 3.5 shows the total number of paths stored in the Adj-RIB-Ins for each organization. Even though RR clients receive two paths for each prefix from their Route Reflectors, the total number of paths is more than twice lower than with a Full-Mesh.

Route Reflection is thus a solution for a more scalable and less error-prone iBGP than a Full-Mesh of sessions, but of course, it comes with a cost, as we will see later in this chapter.

iBGP organization	Total of Adj-RIB-Ins paths
Full-Mesh	105.72×10^6
Route Reflection	43.7×10^6

Table 3.5: Sum of the numbers of paths stored in ISP A's routers

3.2 Path propagation in iBGP

The main goal of iBGP is to propagate BGP information to all BGP routers of an AS, such that they can route packets from and to the Internet. To ensure this, each iBGP router needs at least one path for each destination. But if reachability is the primary goal, additional properties can be obtained by having more than one path for each prefix. As mentioned in the previous chapter, path diversity

is important for fast failure recovery. Wang et al insist on the fact that *"the key to avoid transient failures is to improve the visibility of alternative routes in the BGP system"* [WWGQ09]. Furthermore, when several paths are available, load balancing can be performed among the different BGP nexthops [mul].

With a Full-Mesh, in the extreme case, BGP routers can receive up to n paths to a prefix, n being the number of BGP routers in the AS. With Route Reflection, an ASBR connected to two RRs will receive at most two iBGP paths for each prefix. In this section, we first analyse the amount of diversity available at the borders of ISPs, then we quantify how diversity is propagated inside ISPs.

3.2.1 Path diversity at the borders of an ISP

Considered as a black box, an AS usually learns several paths for each prefix. First, a destination can be reachable via several neighboring ASes. Non Tier-1 ISPs are often connected to several providers to obtain path diversity [ACK03][BGT04]. Those paths will differ by the first AS number in the AS-Path, and we call this type of diversity **Nexthop-AS diversity**.

Second, there are usually several eBGP sessions between two ISPs, for robustness and load balancing purpose. In [MVdSD+09], we evaluate the multi-connectivity of ASes in the Internet via multicast information: `mrinfo` [Jac95] probes were sent to Internet routers, which replied with the list of their physical interfaces. We obtained information from 10,000 routers, about more than 100,000 interfaces. Based on RouteViews routing tables, we mapped the IP addresses of the routers to their AS, and collected information on interdomain links to perform our analysis. We identified 1,000 border routers belonging to 200 different ASes. A more detailed description of the methodology is available in [MVdSD+09].

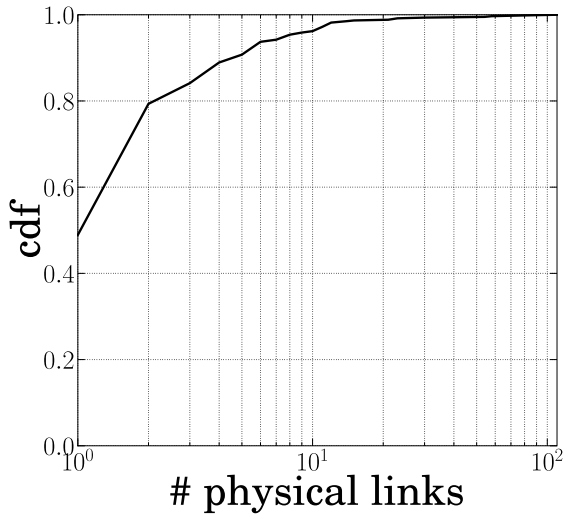


Figure 3.2: Number of physical links between ASes

Results are summarized in figure 3.2. As not all routers reply to `mrinfo`, these results are a lower bound on the real multi-connectivity in the Internet. The X-axis represents the number of physical links (log scale), while the Y-axis represents the cumulated distribution of pairs of ASes. Half of the AS pairs are connected through only one physical link, which means that the other half is connected through multiple physical links.

As peering agreements often require prefix advertisements to be consistent across the multiple links between two ASes [FMR04], there are usually as many paths available per prefix through a given neighbor as the number of links with this neighbor. We call this type of diversity **Nexthop-Router diversity**, because all those BGP paths have a different Nexthop attribute.

The nexthop router can be the local eBGP router or eBGP router of the neighboring AS, depending on wether the latter are reachable in the IGP of the network. If the external eBGP routers are not reachable in the IGP, the local eBGP routers will set the nexthop of the paths received on their eBGP sessions to their own IP address. This mechanism is called **Next-Hop-Self** [WMS04].

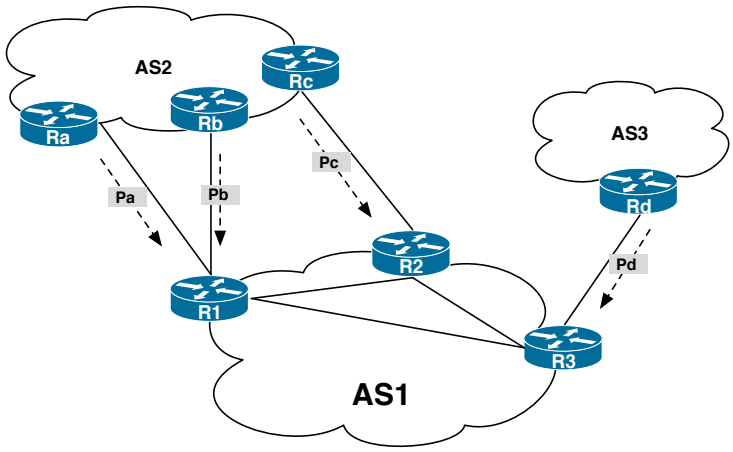


Figure 3.3: Nexthop-router diversity and Nexthop-AS diversity

In figure 3.3, $AS1$ is connected to two neighbors: $AS2$ with three physical links, and $AS3$ with only one physical links. Both neighbors advertise the same destinations. Thus, $AS1$ receives two nexthop-AS diverse paths, one via $AS2$ and the other via $AS3$. If Nexthop-Self is used, there are three nexthop-router diverse paths: via $R1$, $R2$ and $R3$. If Nexthop-Self is not used, the number of available nexthops is four: Ra, Rb, Rc and Rd.

Thanks to the richness of the Internet graph which provides Nexthop-AS diversity and to the multi-connectivity of ASes which provides even larger Nexthop-router diversity, there are usually several paths to each BGP destination available inside an AS.

3.2.2 Path diversity inside ISPs' routers

A router advertises its best path to each destination over both iBGP and eBGP sessions. As only one path is chosen among several ones for propagation, this implies that the remaining paths will stay hidden inside the Adj-RIB-Ins of the routers while they could be useful to other routers for several reasons. Pei and Van der Merwe have highlighted the problems linked to the invisibility of the paths in the case of BGP/MPLS VPNs in [PVdM06].

A first case of restriction of path propagation in iBGP is when a router receives several paths on its eBGP sessions. On figure 3.4, a prefix is advertised on four eBGP sessions to an ISP, from two neighbors. Router $R3$ receives the prefix over two eBGP sessions, and thus knows two external paths: Pb and Pc. However, it can only advertise one of them to its iBGP neighbors, and no other router will learn about the other. We call this loss of diversity **Multi-session Path Loss**. Notice that if the router uses Nexthop-self, both paths are similar from the viewpoint of other routers, as the nexthop address is the same. In case of failure of the session on which Pc is learnt, $R3$ immediately switches on Pb and other routers never learn about the failure. However, if Nexthop-self is not used and some other routers in the AS are using Pc, they will learn about the failure via the IGP, but do not know the existence of the alternate path Pb until BGP re-converges.

Second, even if only one eBGP path is received by a border router, this path can be less preferred than an iBGP-received path with higher local-preference, shortest AS-Path or lower MED. In figure 3.4, router $R4$ receives path Pd from the provider of the ISP, and path Pa from its iBGP sessions with the Route Reflectors. As path Pa was advertised by a customer, it has a higher local preference than path Pd. Thus, $R4$ prefers Pa and does not propagate Pd over iBGP sessions. We call this diversity loss **Non-preferred Path Loss**.

Those two scenarios occur whatever the iBGP organization, because eBGP paths are hidden at the ASBR and are not propagated in iBGP. With a Full-Mesh of iBGP sessions, there is no other restriction, but with Route Reflection, iBGP paths are propagated across several hops in the AS. They can thus be hidden at each of those hops, i.e. by any Route Reflector on the iBGP propagation path. Indeed, as always with BGP, Route Reflectors can only advertise one path per prefix. In the example of figure 3.4, both Route Reflectors $RR1$ and $RR2$ know the two paths that were advertised on iBGP by ASBRs, i.e. Pa and Pc. However, they can only advertise one of them. They will thus advertise Pa, because the nexthop of Pa is closer to both of them than the nexthop of Pc in terms of IGP distance. We call this sort of diversity loss **Route Reflection path loss**.

As a result of all those diversity losses, even though the ISP knows four paths to the prefix, routers $R1$ and $R2$ only learn about one of those paths, i.e. Pa. Notice that if the IGP distances were different, $RR1$ and $RR2$ could have chosen different best paths (ex: Pa for $RR1$ and Pc for $RR2$). Both $R1$ and $R2$ would have received a second path. In practice, as redundant Route Reflectors are often located in the same PoP, they will usually semect the same best paths and send

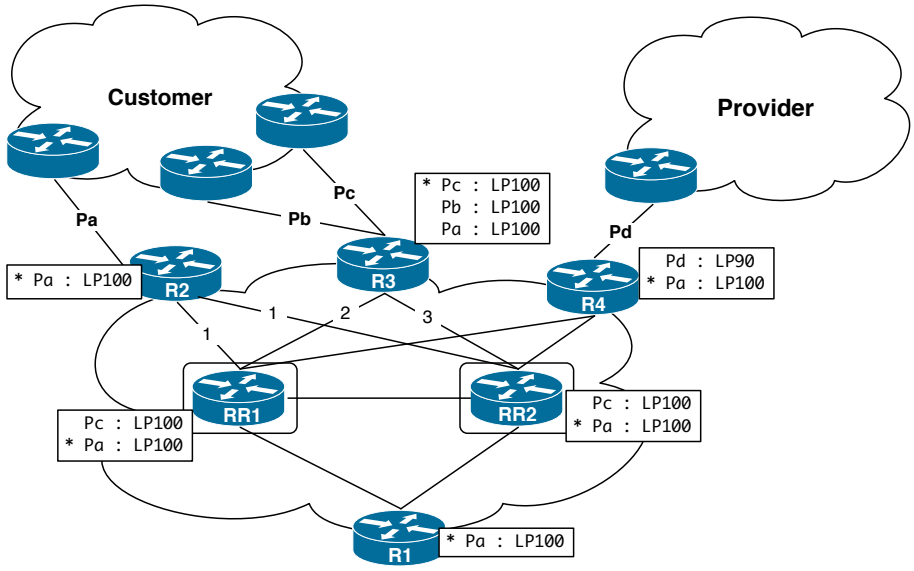


Figure 3.4: Configuration leading to diversity loss

the same paths to their clients. Thus, the clients receive two BGP Updates for that prefix, but only one path because the nexthop is the same.

Best External advertisement [MFCM10] can be used to mitigate the **Non-preferred Path Loss** and the **Route Reflection Path Loss**. This feature allows a router to advertise its best path among its external paths to its iBGP neighbors when its best path is an internal path. In figure 3.4, this would allow $R4$ to advertise path Pd to the Route Reflectors, and $R1$ to learn Pc from the Route Reflectors. However, Best External cannot help $R1$ to obtain an alternate path nor $R3$ to advertise Pb.

Thus, in an ISP, even if path diversity is available at the borders, iBGP path propagation often prevents some routers to learn alternate paths.

3.2.3 Case study

In order to evaluate the extent of the lack of iBGP diversity in real ISPs, we performed an analysis on the three networks presented earlier in this chapter. For each of those ISPs, we obtained data about the paths exchanged in each of them, as follows :

- For GEANT, we received the Adj-RIB-Ins of one of the routers. As GEANT uses a Full-Mesh of iBGP sessions, this routing table contains all the paths except those hidden because of **multi-session path loss** and **Non-preferred path loss**.

- For the Tier-1 ISPs, we received a dump of some Adj-RIB-Ins from top Route Reflectors (2 levels of Route Reflection for ISP A and one for ISP B). Those routing tables are a good sample of the global routing state, but of course, several paths are lacking. Similarly to GEANT, paths hidden through **multi-session path loss** and **non-preferred path loss** at non-dumped routers are lacking. Furthermore, other paths were also missed, through **Route Reflection path loss** at non-dumped Route Reflectors.

In addition to those routing tables, we also obtained sufficient information to build a model of the BGP topology of GEANT and ISP A.

The study we performed in this section is similar to and complements the analysis of Uhlig and Tandel [UT06].

Global path diversity

A first analysis we performed consists in concatenating all paths of each ISP, to evaluate the diversity globally available to the ISP. Due to the diversity loss by dumping only a subset of the routers, the results are a lower bound on the real global diversity of each ISP.

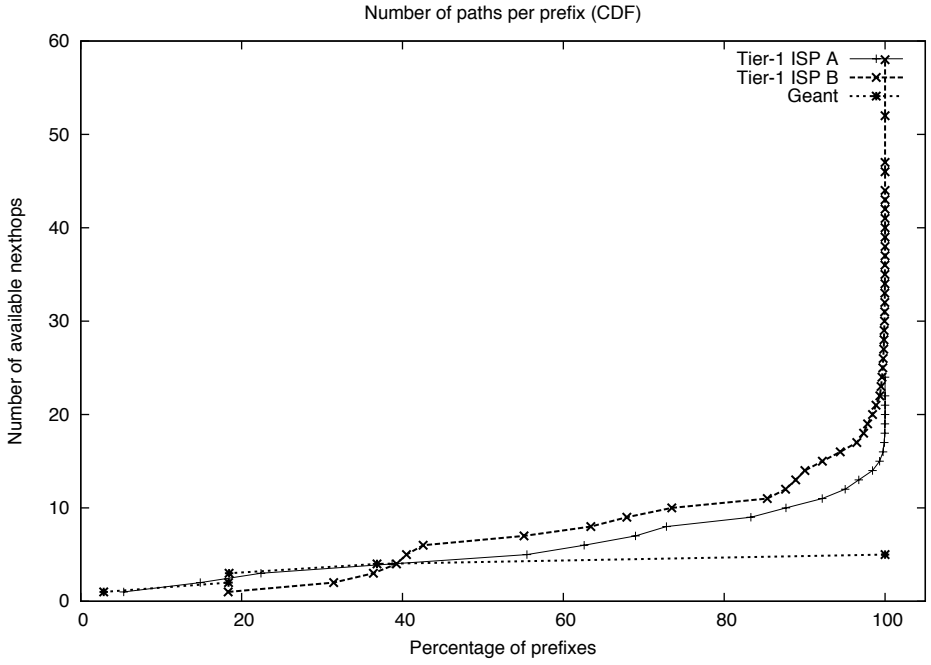


Figure 3.5: Diversity at the borders of three ISPs

Figure 3.5 shows the number of nexthops available for each prefix in each AS. This figure shows that Nexthop-Router diversity is available for most of the prefixes: For GEANT, 97 percent of the prefixes have at least two nexthops, while

for ISP A, this value is 94 percent. ISP B shows the lowest diversity with only 82 percent of the paths having alternate nexthops. This analysis shows that there are only few prefixes for which no diversity is available at the borders of the ISPs.

We performed more accurate analysis on the prefixes lacking diversity in ISP B and discovered that 80 percent of them were advertised by customers. This means that most routers won't benefit from alternate paths for those destinations. In case of failure of the session with a customer, the service cannot be provided to the customer anymore, except if a less specific prefix covers those destinations.

In 60 percent of those prefixes, we were able to discover another eBGP session with the customer owning the prefix, on which it would have been possible to receive an alternate path. A first explanation can be that the customer did not advertise the prefix on all its eBGP sessions, even though this is often required in service level agreements [FMR04]. Or the path was indeed advertised on all sessions, but with different BGP attributes. For example, if the client asks via a BGP community to the provider to set a lower local-preference on paths received over the backup link, this alternate path is hidden because of **non-preferred path loss**. In those two cases, the customer is somewhat responsible for this insufficient path propagation. A third possibility is that the customer advertises equally preferred paths on all its eBGP sessions, but the Route Reflectors all select the same best path. Thus, alternate paths are hidden through **Route Reflection Path loss**

For the remaining 40 percent of customer prefixes lacking diversity in the Route Reflectors, we could only discover one eBGP session with the customer. Diversity was possibly available inside the ISP through another neighbor, either a customer or a peer, but here again, less preferred BGP attributes have hidden potential alternate paths at the border routers.

From this analysis, we conclude that given the current state of the art of iBGP, in order to maximize the path diversity available to the providers, customers should:

1. Be multi-connected to their providers
2. Consistently advertise all their paths on all their links to their providers
3. Be careful while performing backup routing policies with communities and lower local-preference, or when using AS-Path prepending.

Of course, once customers provide alternate paths, providers should take advantage of them for fast recovery services.

Local path diversity

In our three networks under test, sufficient path diversity is available. In order to evaluate the propagation of the alternate paths, we use the C-BGP tool[QU05] to advertise the paths available in our dataset inside a model of GEANT and ISP A (we didn't have enough information to model ISP B). Once the simulation has converged, we count the number of routers that have at least one alternate path for each prefix.

This simulation gives us an approximation of the real availability of alternate paths in the studied ISPs. Indeed, the paths that were not propagated to the dumped Route Reflectors would have slightly improved the diversity of a few routers compared to what we simulate. However, this bias is reasonable, as in the extreme case, only the routers upstream of the Route Reflectors on the propagation paths would benefit from this hidden diversity.

The results of figure 3.6 show that diversity propagation is bad in the Tier-1 ISP, while it is much better in the case of GEANT. The X-axis gives the percentage of prefixes with diversity, while the Y-axis shows the cumulated percentage of routers. For the Tier-1 ISP, 80% of the routers have at least two paths for less than 50 percent of the destinations, while for GEANT, all routers have diversity for more than 95% destinations, thanks to the Full-Mesh. Given the way we obtained the routing tables, we can attribute most of this diversity loss in ISP A to **Route Reflection Path Loss**. We logically observe that the the routers having more diversity in ISP A are the top-level Route Reflectors.

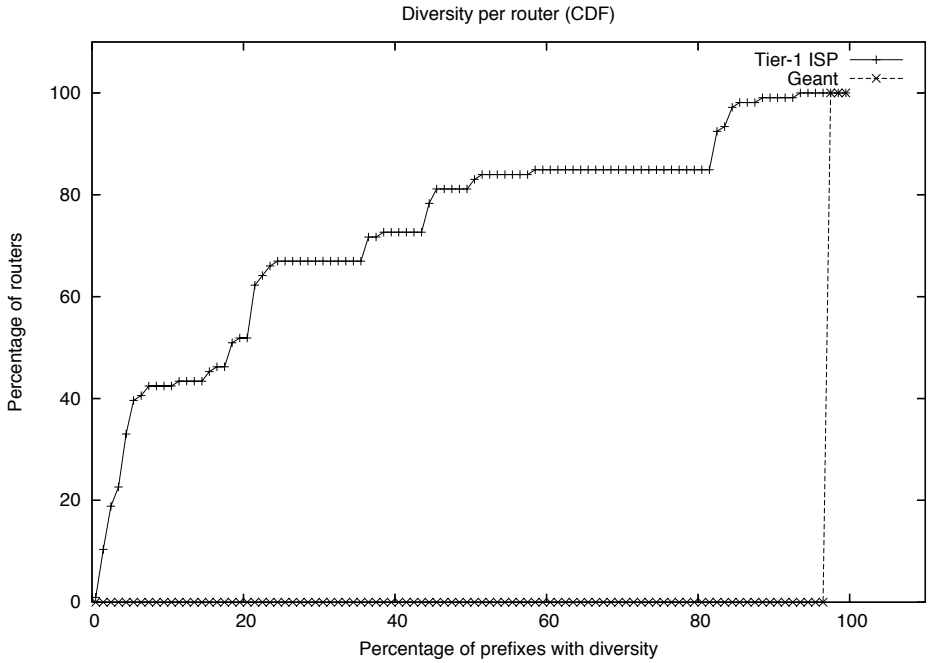


Figure 3.6: Cumulated distribution of routers having Nexthop-Router diversity for each prefix

We used a second methodology to evaluate the propagation of alternate paths in iBGP, using the network of ISP A. This time, we generated synthetic prefixes, one per neighboring AS having at least two eBGP sessions with ISP A. Thus, for each multi-connected neighbor, we advertised one prefix over all eBGP sessions with ISP A, and measured the resulting diversity inside routers in two cases: First,

using a Full-Mesh of iBGP sessions between all routers, and second, using the existing two-levels Route Reflection topology. The results are shown in figure 3.7, with the neighboring ASes being ordered per increasing diversity. With a Full-Mesh, all routers have alternate paths for all synthetic prefixes, except with 17 ASes that have all their eBGP sessions on the same border router (**Multi-Session Path Loss**). With Route Reflection, nearly all neighbors will see alternate paths for their prefixes in less than 50 percent of the routers of ISP A. The "luckiest" neighbor will see alternate paths for the destinations it advertises in 80 percent of the routers of ISP A.

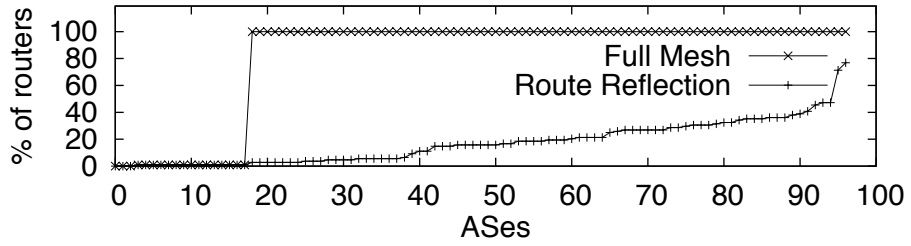


Figure 3.7: Percentage of routers with alternate paths for the prefix advertised by each neighboring AS

This simulation confirms that, even if the ideal case where several paths are advertised with similar BGP attributes on all eBGP sessions with a given neighbor, **Route Reflection Path Loss** severely prevents alternate paths to be propagated in ISP A. We can generalize, and state that it is thus difficult for ISPs using Route Reflection to guarantee fast recovery in case of failure to their customers.

3.3 Consequences of a lack of iBGP diversity on failure recovery

As explained in the previous chapter, alternate paths are important for failure recovery [WWGQ09]. We showed in the previous section that, most of the time, alternate paths are available at the border of an ISP to allow eBGP link failure recovery. Thus, ISPs have all they need to ensure a minimal impact in case of failure of their eBGP links. However, due to the bad propagation of path diversity in iBGP, eBGP link failures often have a large impact, even outside the responsible ISP.

3.3.1 Duration of reachability losses inside the ISP

When a path becomes unavailable, a router must switch as quickly as possible to an alternate path to minimize packet losses. The duration of this path transition has consequences on the dataplane and the control-plane : On the control-plane, when there is no alternate path, the router must send BGP Withdraws to its neighbors,

starting a new path exploration and possibly leading other routers to also loose their reachability [PVdM06].

On the dataplane, as long as the alternate path is not yet installed in the Forwarding Information Base (FIB) and no less-specific prefix is available, the router is forced to drop traffic to the destinations impacted by the failure.

In the best case, the alternate path is already known by the router, and traffic loss is limited to the duration of the FIB update, which is proportional to the number of prefixes impacted by the failure. In the worst case, the dataplane reachability is impaired for the duration of the control-plane re-convergence in addition to the time to update the FIB.

The duration of the control-plane convergence in case of failure depends on three factors. The first is the propagation time of the notification of the failure. If Nexthop-self is used, routers are notified of the failure via BGP, while without Nexthop-Self, they learn the information as soon as the IGP converges, which is much quicker.

Second, the convergence duration depends on the MRAI configuration of the routers. If the MRAI timer is used with the default settings [RLH06], a router may need to wait for up to five seconds before advertising an iBGP Update to another router.

Third, the location of the alternate path also influences the duration of the failure. When the path is hidden by another border router because of **Less-preferred Path Loss**, more iBGP hops are needed to obtain the path than when the alternate path is hidden by the Route Reflector of the ASBRs adjacent to the failure. With a simple one-level Route Reflection topology, in the worst case, when Nexthop-Self is used and the alternate path is hidden by an ASBR of another PoP, three iBGP hops are needed to propagate the BGP Withdraw to the ASBR with the alternate path, and three others to propagate the alternate path back to the first router. The duration of a hop depends on the propagation time on the link, plus the processing time inside the router [FKMT04].

Thus, if the MRAI is applied on both BGP Updates and BGP Withdraws [LABJ01], each hop is likely to delay the BGP messages by the MRAI value, thus, at worst, the duration of six MRAI values is needed in addition to the propagation time accros six hops for control-plane convergence. If it is not applied on BGP Withdraws, the convergence duration is the propagation time of the BGP Withdraw, plus three times the MRAI value, plus the propagation time of the BGP Update. When two levels of Route Reflection are used, the distance between a primary router and the alternate nexthop can be up to five iBGP hops.

In the network of figure 3.8, an ISP receives two paths from neighbor ASY. Due to backup policies, path Pb has a lower local-preference, and is thus hidden by router $R2$. If Nexthop-Self is not used, $R2$ learns the failure of Pa thanks to the IGP, and as it knows the alternate path Pb, it can immediately select it as new best path. The duration of the failure for $R2$ is thus the IGP convergence time, plus the time to change the nexthop of the destinations advertised by ASY in its FIB. If Nexthop-self is used, the duration of the failure for $R2$ is the propagation

time of the BGP Withdraw sent by $R1$, i.e. thee iBGP hops, in addition to the FIB update. For $R1$, the duration of the failure is longer, as it does not know an alternate path. Thus, it needs to wait until $R2$ advertises the alternate path Pb. Thus, the convergence time of $R1$ is the convergence time of $R2$ plus the propagation time of the alternate path, i.e. three iBGP hops. In the worst case (i.e. Nexthop-Self), this represents in total six iBGP hops plus the update time of the FIB of $R1$. During all this time, $R1$ is unable to forward packets from ASX to ASY, and drops the traffic.

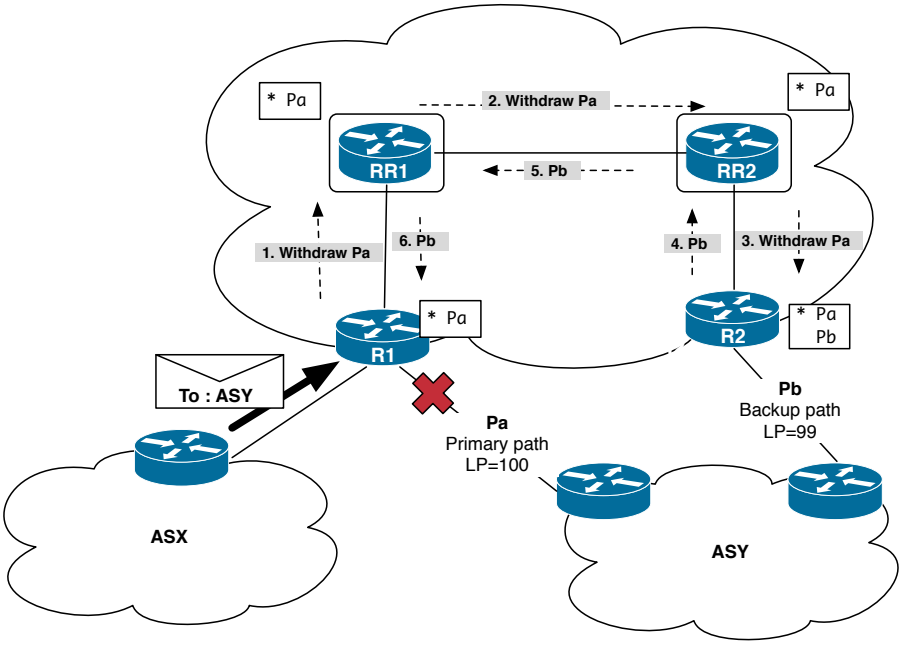


Figure 3.8: iBGP convergence upon link failure

Wang et al. [WGWQ05] measured the duration of such transient failures inside a Tier-1 ISP, and report that even though such failures are usually short-lived, some last several seconds. IBGP convergence has thus an impact on the performance of end-to-end paths.

3.3.2 Increased BGP churn

IBGP convergence in case of failures not only has an impact on packet losses, but also on BGP churn in the global internet. As soon as a router changes or removes its best path, it has to propagate that new best path to all its neighbors, i.e. on iBGP AND eBGP sessions. In the best case, all routers know an alternate path via the same Nexthop-AS as the failed path. Thus, only the nexthop attribute of the path needs to be changed. As the nexthop attribute is set to the IP address of the border

router before advertisement over eBGP session, the path seen by eBGP neighbors does not change and no eBGP message is needed to reflect the change in path. The failure is resolved locally.

A second scenario is when the alternate path is available to all routers, but its nexthop-AS is not the same as the one of the failed path. This time, the AS-Path is modified, and the change must be propagated over eBGP sessions. Depending on the AS-Path length of the new path and on the policies to which it is subject, this path change may have an impact on global routing. Indeed, if the alternate path is via a peer while the failed path was via a customer, the AS must withdraw the advertisement of this prefix over its eBGP session with providers and peers. If those neighboring routers do not have alternate paths, reachability losses and churn propagation may occur in other ASes as well.

Finally, when alternate paths are not available to all routers, iBGP convergence has an even more important impact on BGP churn. Indeed, path exploration can occur inside the AS while alternate paths are exchanged between the routers. Routers may explore different alternate paths before finally selecting their post-convergence path. During this convergence, they may transiently be in a state where they have no path to the destination [LABJ01]. In this case, they will send BGP Withdraw messages over all their eBGP sessions, thus exporting the failure outside the ISP. They may also explore alternate paths with less favorable export policies than the post-convergence paths, also leading to transient BGP Withdraw messages being leaked outside the AS.

Withdraw-Blocking property

To prevent the unnecessary announcement of a local failure to the entire Internet, BGP Withdraws must be stopped as close as possible to the location of the failure. We call **Withdraw-Blocking** a router or an AS that is able to stop the propagation of a BGP Withdraw.

Definition 3.3.1. An AS is said to be **Withdraw-Blocking** for a destination D if that AS advertises D on at least one eBGP session and does not propagate a BGP Withdraw to a neighbor not advertising D itself, upon failure of its primary path towards that destination.

On the topology of Fig. 3.9, $AS3$ is Withdraw-Blocking for destination D. For example, if the link between $AS1$ and $AS2$ fails, the BGP Withdraw is propagated by $AS2$ to $AS3$. $AS3$ knows the alternate path via $AS5$, such that it can advertise this alternate path to $AS4$ instead of a propagating the BGP Withdraw. A BGP Withdraw is still sent to $AS5$, but as this neighbor uses and advertises the alternate path, this Withdraw will not result in a connectivity loss.

Obviously, the alternate path must itself remain stable during the failure, which means that it must not be itself impacted by this failure. We will call such stable path a *valid* path. Thus, an AS must know a valid alternate path to reach the destination in order to be Withdraw-Blocking. However, this is not sufficient, as the

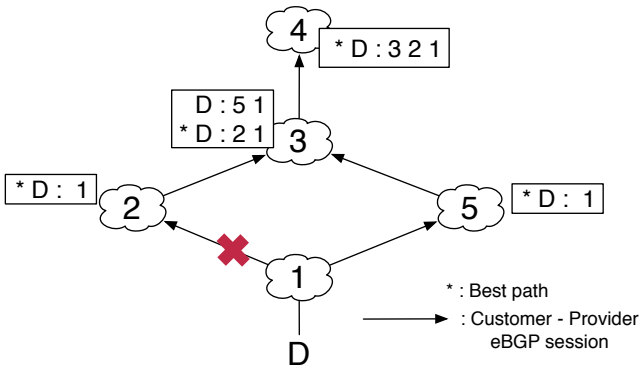


Figure 3.9: Withdraw-Blocking AS

AS must forward this alternate path to its neighbor to replace the BGP Withdraw message. Policies can prohibit the announcement of the alternate path on some eBGP sessions [GR01]. Therefore, we define a new property for an alternate path:

Definition 3.3.2. Let S_{P_X} be the set of eBGP sessions on which a path P_X would be advertised if it were the only path available inside the AS. A path P_A to destination D is **export-policy compliant** (EPC) with another path P_B to the same destination if S_{P_B} is included in S_{P_A}.

Theorem 3.3.1. *An AS is Withdraw-Blocking for a destination if and only if it knows a valid export policy compliant alternate path for its primary path to that destination.*

When the policies used in the AS are the classical routing policies [GR01], this theorem can easily be proven:

Proof. Two cases must be considered. First, if a BGP Withdraw is received from a provider or a shared-cost peer, the only sessions on which the AS was advertising the destination D are customer sessions. In this case, any alternate path is export-policy compliant, and can be advertised to customers in replacement of the failed path. The second case is when the BGP Withdraw is received from a customer. If there exists an alternate path via a peer or a provider, this alternate path is not export-policy compliant. It can be advertised to customers, but not over session with peers or providers. The destination is thus withdrawn on those sessions. If the alternate path is learned from the same or another customer, this path is export-policy compliant. As a customer path is preferred over peers and providers paths [GR01], it will be selected as best once the primary path is withdrawn. The propagation of this customer alternate path is not constrained by policies and a BGP Update message is sent instead of a BGP Withdraw. Thus, when the AS knows a valid export-policy compliant alternate path, the AS is Withdraw-Blocking.

□

The validity constraint of the alternate path is difficult to ensure for distant failures, because all alternate paths known by the AS can be impacted by a given failure. However, when the failure is local, i.e. concerns an interdomain link directly connected to the AS, all other available nexthops are valid and can be used as backup.

In practice, most ISPs are able to block transient BGP Withdraws, because they often are multi-connected or have policy-compliant alternate paths. We measured the policy-compliance of alternate paths in GEANT and in Tier-1 ISP A. GEANT has export-policy compliant alternate paths for 97 percent of its prefixes, and ISP A for 91,8 percent of its prefixes. Again, those results are a lower bound on the real policy-compliance of alternate paths, because we do not have all alternate paths in our dataset.

As we already showed that path propagation was bad in iBGP, we need to refine our theorem to the router-level.

Theorem 3.3.2. *An AS is Withdraw-Blocking for destination D if all routers of the AS know at least one valid alternate path to D that is export-policy compliant with their primary path.*

Proof. If all routers of the AS have at least one valid export-policy compliant alternate path, any router that receives a BGP Withdraw is able to send a BGP Update with the alternate path instead of propagating the BGP Withdraw for the primary path on its eBGP sessions. Withdraw propagation is then blocked directly at the border of the AS. Also, when an external nexthop fails, all routers that learn the failure via the IGP have an alternate path, and none will send a Withdraw. □

Propagation of unnecessary BGP Withdraws

The propagation of unnecessary BGP Withdraws by an AS is responsible for transient losses of connectivity [WMW+06, KKKM07]. We have analysed RouteViews BGP feeds [Rou] to evaluate the occurrence of such BGP Withdraws in the Internet. We define a Withdraw as iBGP-caused if it was sent by a router of some AS but at least one other router of the same AS did not send a BGP Withdraw for the same destination during the same period of time. Indeed, if the second router does not send a BGP Withdraw, this means that either it uses another path, or that it knew an alternate path to replace the withdrawn one. In such a situation, at least two paths are available inside the AS, but the valid alternate path is not known by all routers.

For our evaluation, we took the BGP data from the first two weeks of October 2008 on the RouteViews Oregon collector, and considered the BGP messages received from pairs of routers belonging to the same AS.

First, we filter all BGP messages received during reboot periods using the BGPMCT algorithm [ZKL+05]. Second, we classify a BGP Withdraw for a destination as iBGP-caused if it is seen on one session with an AS while the other router of this AS has a stable route, i.e. no Withdraw for that destination is seen

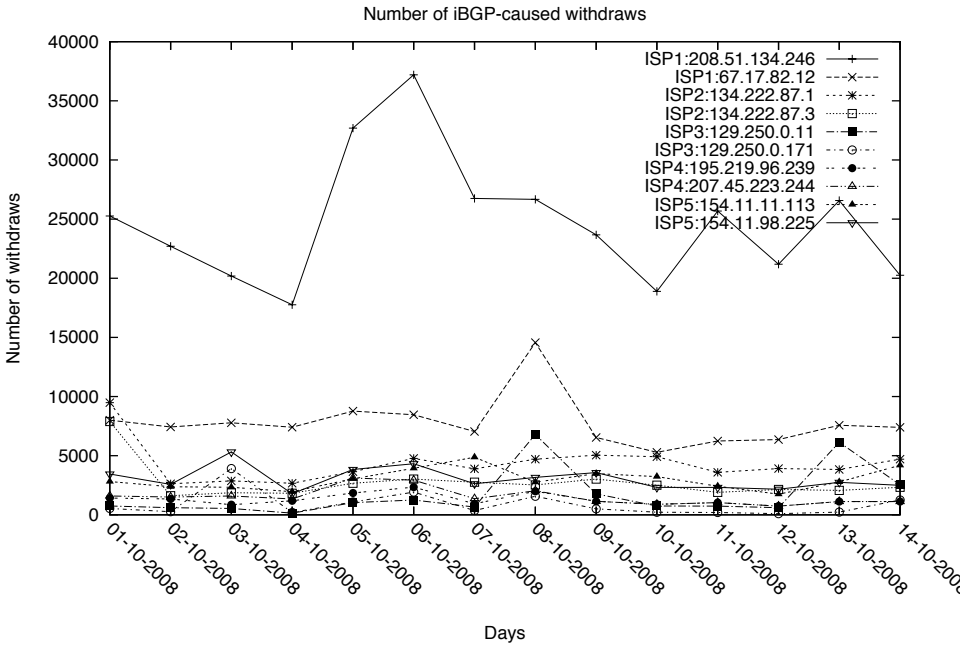


Figure 3.10: Number of iBGP-caused BGP Withdraws

on the session with the other router during 30 seconds before and after the BGP Withdraw. This is an upper bound to the propagation time of the BGP Withdraw for the path inside an AS, if we assume that, at worst, the BGP Withdraw has to cross a whole two-levels iBGP hierarchy between two edge routers, which gives 5 BGP hops.

Results are displayed on figure 3.10. The x-axis shows each day of measurements, while the y-axis shows the number of iBGP-caused Withdraws. Each curve represents one monitored eBGP session, identified by the IP address of the router peering with the collector.

The figure shows that most routers send several thousands of iBGP-caused Withdraws per days. One router in particular even sent on average more than twenty thousands of BGP Withdraws per day. On a per-hour basis, results show peaks of more than 2,000 iBGP-caused Withdraws per hour. Variations between the results for different routers are probably due to different iBGP configurations, but we don't have information about the organizations of the observed ASes.

Still, this analysis shows that for all the routers that we analysed, the number of iBGP-caused Withdraws is large, and reducing this particular churn would help reducing transient losses of connectivity in the Internet. Thus, operators should ensure that their routers are Withdraw-Blocking by providing them sufficient diversity.

3.4 Consequences of a lack of iBGP diversity on routing correctness

In addition to bad failure recovery, iBGP path losses also impact the routing correctness. Given an ISP and a set of paths advertised on eBGP to this ISP, all routers should be able to forward traffic on their preferred path. However, some paths are hidden by routers even though they are the preferred paths of some other routers, or would have been useful to other routers for a complete path selection. Furthermore, iBGP sessions are not necessarily established congruently to the IGP graph, and the physical path between two iBGP neighbors can contains intermediate BGP routers between the two iBGP neighbors that may have conflicting path selections, leading to forwarding anomalies.

In this section, we first explain the impact of a bad iBGP path propagation on routing optimality, then we show that these propagation issues can even lead to routing anomalies inside an ISP with non congruent IGP and iBGP topologies.

3.4.1 Path sub-obtimality

An issue with iBGP is that Route Reflectors may hide some paths based on local criteria. This occurs when several paths of equal global preference (local-pref, AS-Path length, MED value) are available, and the decision process chooses its best path based on the IGP distance to the nexthop.

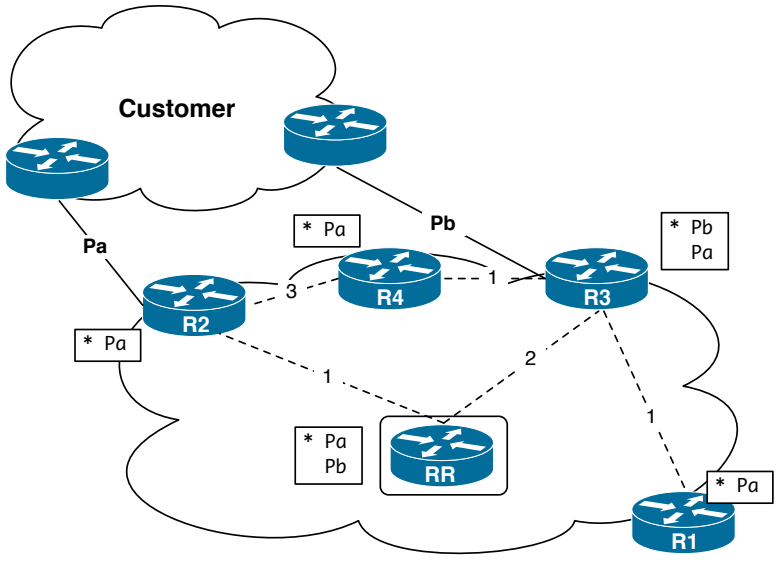


Figure 3.11: Topology with sub-optimal routing. The Route Reflector has iBGP sessions with $R1$, $R2$, $R3$ and $R4$. Dashed lines represents IGP links with their IGP costs

On the topology represented in figure 3.11, the Route Reflector receives two paths, and selects its best based on the IGP cost. Pa is its preferred path because the distance to the nexthop $R2$ is one, compared to two for the alternate path Pb. Then, it advertises Pa to $R4$, which, as it only knows that path, will select it as best. However, the best path for $R4$ is actually Pb, because its nexthop is closer to $R1$ than the nexthop of Pa. Thus, by selecting the path it advertises to $R4$ based on its own local criteria, the Route Reflector induces sub-optimal routing in the ISP. Hot Potato routing is not guaranteed anymore.

Sub-optimal routing can be prevented by using Intelligent Route Reflectors [BUQ04], where the Route Reflectors compute the paths they advertise to their client based on the preferences of those clients. Routing Control Platforms [FBR+04] [CCF+05] offer similar services.

3.4.2 Forwarding loops and deflections

In addition to sub-optimality, Route Reflection can also result in forwarding anomalies. First, **forwarding deflections** can occur when a router along the forwarding path prefers another nexthop than one selected by the ingress router. This typically happens with multi-hops iBGP sessions. The traffic is then deviated to another exit point. In the best case, the deflecting router immediately sends the traffic outside the AS, but traffic can also be sent back inside the AS toward another exit point.

Traffic deflection occurs in the example of figure 3.11. Router $R1$ receives path Pa from the Route Reflector, but when $R1$ forwards traffic to the nexthop $R2$ of this path, it uses $R3$ as an intermediary node. However, $R3$ uses path Pb for that destination, and will thus directly forward the traffic on its eBGP link.

Even though deflection might look quite innocent, in certain situations, they can be problematic. Indeed, if a router deflects traffic back to an upstream router, a **forwarding loop** will occur. It is even possible to find deflection situations that result in forwarding loops between Autonomous Systems [GW02b]. The example of figure 3.12 shows a topology were deflections result in an intradomain forwarding loop. Two paths to a destination are known by the ISP, $P1$ and $P2$. The two Route Reflectors each prefer their eBGP-received path, and advertise it to their respective clients. Thus, $R1$ uses $P1$, and $R2$ uses $P2$. Due to the IGP graph, the path from $R1$ to $RR1$ is via $R2$, and the path from $R2$ to $RR2$ is via $R1$. When $R1$ wants to sent traffic to its nexthop $RR1$, it forwards the packets towards $R2$. However, for the same destination, $R2$ uses nexthop $RR2$, and it will thus deflect the traffic to its preferred exit point. But as $R1$ is on its forwarding path to $RR2$, traffic gets deviated again towards $R2$, and a forwarding loop is created.

Detecting whether a given iBGP configuration will result in forwarding deflection is hard. It is actually NP-Hard [GW02b], but there exists sufficient conditions on the topology to determine forwarding correctness of a topology [GW02b]:

1. A Route Reflector must prefer paths received from clients over paths received from non-clients

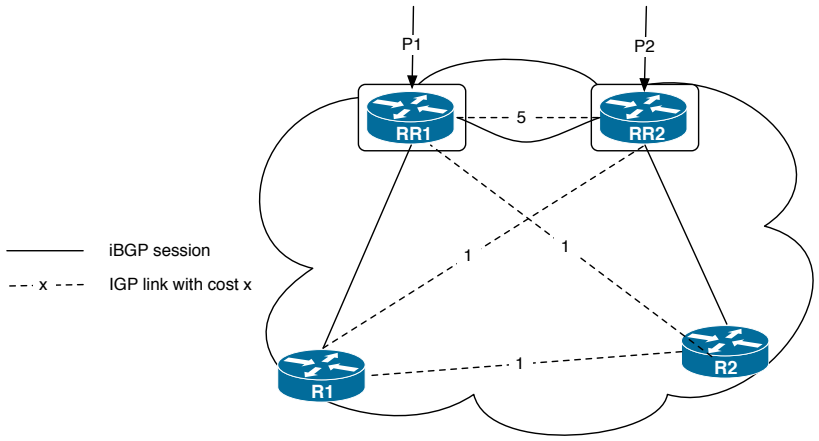


Figure 3.12: Topology with forwarding loop

2. The shortest path between two routers must be a valid propagation path, i.e. the reverse of the path followed by packets is a path that iBGP messages are allowed to follow.

Another solution that does not impose constraints on the topology of the network is traffic encapsulation: If a tunnel is used to transport packets from the ingress to the egress, only the ingress node needs to look inside its own routing tables, and deflections and forwarding loops are prevented. IP or MPLS encapsulation can be used to this end. Note that two levels of encapsulation must be used, one to the egress router, and one to the egress interface. A single level of encapsulation would prevent intermediate node to deflect the traffic, but the egress router would still be able to deviate packets to another egress link. The second level of encapsulation prevents the egress router to perform a lookup in its routing table.

3.4.3 Convergence issues

In eBGP, conflicting policies of ASes can result in routing oscillations. Similarly, in iBGP, there exists configurations where the convergence is not guaranteed. In this case, the role played by conflicting policies in eBGP is held by conflicting path selection based on local criteria.

Actually, there are two different sorts of iBGP routing oscillations: IGP/BGP induced oscillations [BOR+02][GW02b], and MED-induced oscillations [GW02a]. IGP/BGP induced oscillations are due to interactions between the IGP graph and the Route Reflection topology, while MED oscillations are caused by interactions between IGP costs and the MED attribute.

IGP/BGP oscillations

With a Full-Mesh of iBGP sessions, a router advertises a path only when it received it from an eBGP neighbor. Thus, the IGP tie-break only applies on iBGP-received paths, and the advertisement of paths over iBGP sessions is not influenced by the IGP tie-break rule of the decision process. With Route Reflectors, however, the advertisement of a path may depend on the IGP preferences of a router. We already explained that this could lead to sub-optimal routing and to forwarding deflections or loops. In addition, this can result in routing oscillations, because the best path selection of a router may depend on the choice of another router. In the topology

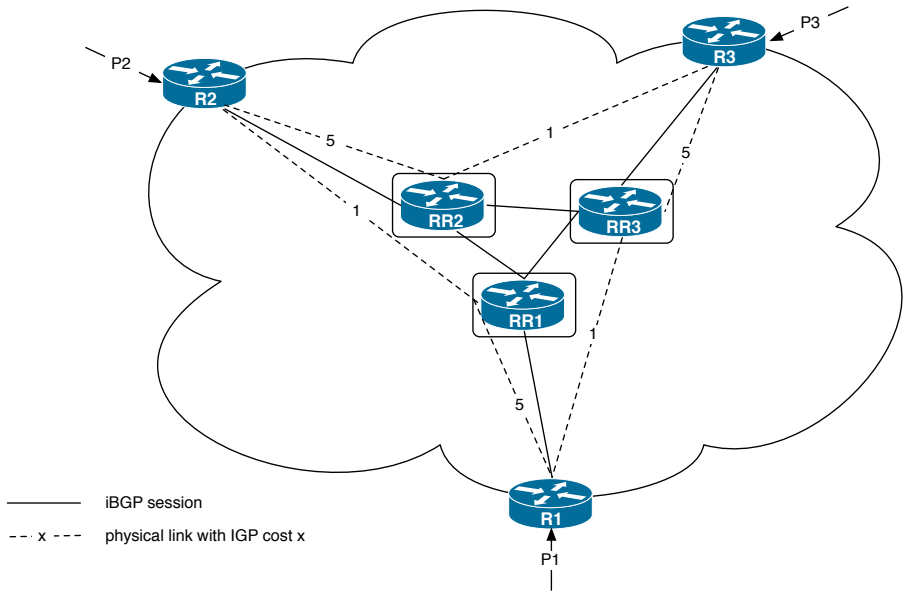


Figure 3.13: Topology with IGP/BGP induced routing loops

of figure 3.13 [GW02b], three Route Reflectors have one client each, but the IGP configuration is such that the client of their right-handed neighbor is closer to them than their own client. Initially, all RRs only know about their client paths, and advertise it to their neighbors. Once they learn the path from their right neighbor, they all select it as best, and withdraw their own path. But their current best path then becomes unavailable, and they have to switch back to their client path. The system is identical to the eBGP-oscillating topology of section 2.2.2.

Other routing anomalies exist as well, where the final routing state is non-deterministic, i.e. there are several possible outcomes depending on the timing of iBGP advertisements [BOR+02][GW02b].

Griffin and Wilfong [GW02b] have proven that showing that a given iBGP configuration results in routing loops is a NP-hard problem. Fortunately, they also provide two sufficient conditions for a topology to be routing loop-free. Here again,

these conditions are similar to the conditions for eBGP loop-free topologies.

1. The directed graph composed of the client/Route Reflector sessions is a Directed Acyclic Graph

2. All routers prefer the paths from their clients over the paths from their non-clients.

MED-induced routing loops

The second case of iBGP routing oscillations is due to the MED attribute. As explained in chapter 1, the MED attribute is only comparable between paths received from the same neighboring AS. A path that was initially preferred due to the IGP tie-break can be discarded when another path with a lower MED is received, even if that new path is not preferred over the others from the IGP distance viewpoint. Thus, the outcome of the decision process depends on whether other routers have advertised other, non necessarily better paths from the same neighbor or not.

In figure 3.14[GW02a], router $R1$ receives two paths Pa and Pb. Pb from ASY has a MED value of 1, and is preferred by $R1$ over Pa at the tie-breaking rule. Router $R2$ receives another path Pc from ASY. Initially, $R1$ advertises Pb to the Route Reflector, while $R2$ advertises Pc. The Route Reflector compares the paths, and selects Pc because it has a lower MED, then advertises that path to $R1$. When it knows about Pc, $R1$ cannot advertise Pb anymore because of the MED. However, it does not select Pc as best, but Pa instead. This time, with Pa and Pc in its Adj-RIB-Ins, the Route Reflector changes its mind and prefers Pa, because it is closest from the IGP viewpoint than Pc. It withdraws the advertisement of Pc to $R1$, and $R1$ thus switches back to Pb, as it is not invalidated anymore by a path with lower MED. We are back in the initial situation, and the system oscillates.

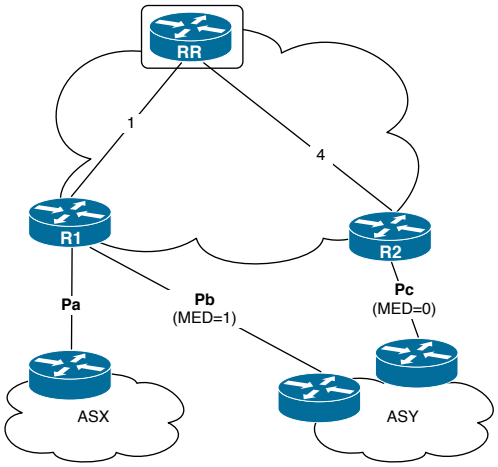


Figure 3.14: Topology with MED-induced routing loops [GW02a]

MED oscillations are a known issue among operators, and is one of the cause of flapping prefixes. Wu et al. identified the MED attribute as responsible for 18,3 percent of the oscillating BGP Updates they observed in their study [WMRW05].

Solutions to solve routing loops

Clearly, along with non-congruent IGP and iBGP topologies, one of the origins of routing loops is the limitation of the best path-only advertisement in iBGP. Thus, several authors have proposed to advertise additional paths in iBGP, and the encoding for BGP multiple paths advertisement (BGP Add-Paths) is undergoing standardization [WRC09a].

Different sets of paths have been identified as sufficient for each type of routing loop: For example, Walton et al. [WRC09b] propose to advertise the best path of each group of paths received from the same AS. This allows paths subject to the MED attribute not to be in competition for propagation with paths that do not contain the MED attribute, and this set is thus sufficient to prevent MED oscillations. [FR09] even refines this set of paths by limiting it to the best paths among paths non subject to MED, and the best path of each group of paths advertised by an AS using MED.

However, IGP/BGP routing loops are not solved by this subset of paths. Basu et al [BOR+02] propose to advertise all AS-wide preferred paths, i.e. all paths having the best local-preference, shortest AS-Path and lowest MED among comparable paths. They prove that this set of paths is sufficient to prevent all routing and forwarding anomalies. By advertising multiple paths between Route Reflectors and selective paths between each Route Reflector and its clients (i.e. paths selected based on the client preferences), the authors of [MC04] are also able to prevent both sorts of routing loops to happen.

Flavel et al. [FR09] also propose a different method for solving iBGP oscillations. This method does not involve multiple paths advertisement, but adds an additional step in the iBGP decision process. This new decision step consists in preferring paths with a minimum iBGP hop count, prior to using the IGP tie-break. This solution does not involve any protocol change, but hot-potato routing is not ensured anymore, and the semantic of the MED attribute can be violated.

3.5 Conclusion

In this chapter, we relied both on the state of the art and on measurements on routing data of three ISPs to analyse iBGP routing. We have shown that even though ISPs have in theory a total control over their iBGP routing, in practice, current organizations lead to either scalability issues or diversity losses. The consequences of the diversity losses are multiple, ranging from routing sub-optimality or even inconsistencies to recovery issues.

Part II

Improving interdomain routing through iBGP

Chapter 4

Taxonomy of iBGP solutions

In the first part of this thesis, we identified several weaknesses of the current iBGP. On several aspects, iBGP does not meet the needs of ISPs, and finding a new way to propagate eBGP paths inside an AS is an important research field. In this chapter, we identify various requirements for iBGP, then we present different proposals for iBGP replacement or improvement. We finally propose a classification of these solutions based on the list of requirements.

4.1 iBGP requirements

4.1.1 Scalability

The first requirement that we mentioned in chapter 3 is that the iBGP system should be scalable, i.e it should be able to support a large number of speakers, without causing a memory overload. Furthermore, it should not lead to long periods of high CPU usage during convergence due to the processing of iBGP messages. The classical Full-Mesh is known to suffer from scalability problems, and Route Reflection aims at solving this issue.

4.1.2 Forwarding correctness

The iBGP system should result in correct forwarding, i.e. without deflection and forwarding loops. This property is difficult to ensure when legacy BGP with hop-by-hop forwarding is used along with Route Reflection, but an encapsulation technique such as MPLS from the ingress node to the egress link can guarantee correct forwarding.

4.1.3 Routing correctness

The iBGP routing must be stable, so that the system does not leave room for MED or IGP/BGP path oscillation. While a Full-Mesh organization is stable, Route Reflection is known to suffer from such issues.

4.1.4 Routing optimality

The iBGP system should result in optimal routing with respect to BGP policies and Hot Potato routing. A Full Mesh of iBGP sessions allows for optimal routing, but, as shown in chapter 3, Route Reflection does not.

4.1.5 Path diversity

Path diversity is the availability of multiple paths to one given prefix in the Adj-RIB-Ins of the routers when the network is stable. This diversity is desirable for fast recovery in case of path withdrawals, as it provides alternative paths that can be used as replacement of the withdrawn path. Another application of path diversity is load balancing: load balancing traffic among redundant peering links has been proposed within the inter-domain routing working group of the IETF, to satisfy requirements of operators to enable finer traffic engineering over their peering links, as requested in [Gil06]. A show-stopper for the deployment of such solutions is the low path diversity that has been found in ISPs networks. Obviously, path diversity is desirable. Any iBGP system should then provide as much path diversity as possible, while still staying scalable.

4.1.6 Failure isolation

The particularity of the iBGP routing is that all routers participating are under the same administrative control. Thus, nothing should prevent the propagation of the information. In particular, when an alternate path is available for some destination, the routing system should be able to recover from the failure of the primary without leaking failure notification to the rest of the Internet.

4.1.7 Automatic configuration

This requirement is motivated by cost reduction plans of ISPs [Gil06] as well as by the fact that iBGP topologies are known to lead to configuration errors due to human operations [MWA02]. It follows from those discussions that the iBGP system should configure itself on its own, with as few human effort as possible. It should adapt to internal topology changes and automatically reconfigure itself, without human intervention. Such internal topology changes are typically UP and DOWN events in the case of BGP speakers. Ideally, the only things that an operator should configure are its inter-domain policies [VQB09]. Note however that automating the configuration of the iBGP organization should not lead to instabilities, i.e. the set of iBGP sessions established between BGP speakers must not be continuously changed by the BGP system.

4.1.8 Robustness

The robustness requirement is the capacity of any iBGP system to support the removal or failure of iBGP nodes. More specifically, the reachability of external destinations must not be compromised when k iBGP sessions or BGP speakers are removed from the iBGP topology. k should be configurable by the operator. Topologies with redundant Route Reflection provide robustness in the case of failure of Route Reflectors, with k being equal to the number of Route Reflectors per client minus one.

4.1.9 Simplicity

It is important for operators to be able to understand the behavior of their network. Thus, the complexity of the iBGP organization must be limited in order to ease troubleshooting in case of problems.

4.1.10 Incremental deployment

Legacy BGP speakers should be integrable within the system without harming too much the fulfilment of the other requirements. As few changes as possible should have to be made to the BGP protocol itself.

4.2 Survey of iBGP solutions

Lots of proposals have been published to improve current iBGP with Route Reflection. In this section, we present each of them based on the requirement they aim to fulfill.

4.2.1 Optimization algorithm to build iBGP topologies

Several authors rely on algorithms to statically compute Route Reflectors topologies that are forwarding correct and give optimal routing [VVKB06][BUM08]. However, these topologies do not guarantee path diversity nor routing stability. On the opposite, Pelsser et al. build iBGP topologies that provide path diversity [PQU+10]. Unfortunately, they do not guarantee routing stability, but propose to rely on the solution proposed by Flavel et al [FR09]. Both Xiao et al. [XWN03] and Buob et al. [BUM08] guarantee robustness in their iBGP topology. A common issue with these algorithmic approaches is that they usually do not leave room for flexibility : The addition or withdrawal of a session can break the desired property and force the operators to recompute a correct iBGP topology and re-establish iBGP sessions.

4.2.2 Oscillation prevention

We already mentioned several works dedicated to oscillation prevention. Either they rely on a modification to the decision process [FR09], or they propose to advertise several paths instead of one on iBGP sessions [MC04][BOR+02][FR09] [WRC09b]. However, even though the advertisement of multiple paths in iBGP can be used to support fast failure recovery thanks to alternate paths [MFFR08], none of the proposals above considers the issue of path diversity. Another approach is to give more information to Route Reflectors in order to compute better paths for their clients [BUQ04][MC04]. This increases routing optimality, but once again, path diversity remains poor. Routing Control Platform [FBR+04] [CCF+05] can be used to separate routing from forwarding, but the network becomes vulnerable to failures of the devices dedicated to path computation and distribution.

4.2.3 Increased path diversity

Path diversity can be improved by advertising the Best-External path in iBGP [MFCM10], but as explained in chapter 3, this does not prevent Route Reflectors from hiding alternate paths to their clients. Operators should either build their iBGP topology with this objective in mind [PQU+10], or deploy multiple path advertisement in their network [MFFR08]. Bonaventure et al. [BFF07] propose a solution for fast recovery that does not strictly increase path diversity, but achieves the same objective by providing a protection tunnel to an alternate link in case of failure.

4.2.4 Auto-configuration

Raszuk et al. have proposed a solution to allow routers to automatically detect BGP speakers within an AS and establish iBGP sessions among them [RAM03]. However, the current solution only allows to establish iBGP Full Meshes. It should be possible to automatically establish more scalable iBGP topologies. The solution of fast recovery of Bonaventure et al. [BFF07] also allows for an automatic discovery of backup links to build protection tunnel.

4.2.5 Incremental deployment

Incrementally deployable solutions are those that allow a subset of the network to use the new features, while letting other routers operate normally. Solutions relying on a specific structure of iBGP sessions are incrementally deployable as long as additional iBGP sessions can be established step by step on top of the existing iBGP organization, until the desired property is fulfilled. This is the case of the nexthop-diverse topology of Pelsser et al. [PQU+10]. Best External Advertisement can be activated on selected border routers, without impacting the behavior of their iBGP neighbors. The fast recovery solution of Bonaventure et al. is also incrementally deployable [BFF07].

Requirements	iBGP solution
Scalability	Route Reflectors
Routing optimality and Forwarding Correctness	Full-Mesh, Double Encapsulation, Vutukuru et al. [VVKB06], Musuruni et al. [MC04], Buob et al. [BUM08], Intelligent Route Reflectors [BUQ04], Routing Control Platform [FBR+04][CCF+05], Basu et al. [BOR+02]
Routing Correctness	Full-Mesh, Flavel et al. [FR09], Walton et al.[WRC09b], Basu et al. [BOR+02], Add-Paths [MFFR08]
Path Diversity	Best External [MFCM10], Protection tunnel [BFF07], Nexthop-diverse iBGP topology [PQU+10], Add-Paths [MFFR08],
Failure Isolation and reduction of Internet churn	Nexthop-diverse iBGP topology [PQU+10], Add-Paths [MFFR08], Protection tunnel [BFF07]
Automatic Configuration	iBGP auto-mesh [RAM03], Auto-discovery of protection tunnel[BFF07]
Robustness	Redundant Route Reflectors, Reliable RR topology [XWN03] [BUM08],
Simplicity	Full-Mesh
Incremental Deployment	Best External, Nexthop-diverse iBGP topology [PQU+10], Protection tunnel [BFF07]

Table 4.1: Taxonomy of iBGP solutions

4.3 Conclusion

In this chapter, we have classified the existing proposals of iBGP organizations and improvements based on a list of requirements. Table 4.1 summarizes the properties of each iBGP solution described in this chapter. This table shows that the solutions proposed in the literature usually aim at a single requirement.

Each operator will set its own priorities on the requirements that we identified, and will have different scalability and deployment constraints. Thus, each operator might be interested in different iBGP mechanisms, or even different combinations of mechanisms. An example of such combination is given by Pelsser et al. [PQU+10], when they propose to use their iBGP organization improving the path diversity of the routers with the routing oscillation prevention solution of Flavel et al. [FR09].

In this thesis, we aim at improving the set of available mechanisms, with a particular focus on solving the iBGP path diversity issue. First, we will propose three solutions to solve the issues related to path diversity. The first one aims at limiting the churn propagation to improve failure isolation, by propagating diversity availability via a community. The second proposes to exploit the path diversity received from multi-connected neighbor to provide fast recovery for their destinations. The third is a complete organization built on the principle of the second solution, providing a full automation of the configuration.

Finally, we provide an in-depth analysis of another iBGP mechanism called Add-Paths that consists in advertising multiple paths for a prefix over iBGP sessions [WRC09a]. We identify and analyse different proposals of sets of paths that can be advertised, some of them having already be mentioned in this chapter [WRC09b][BOR+02][MFFR08]. Even though the scalability of this solution must be carefully evaluated before deployment, we show that it provides several options for iBGP improvement, and can fulfill several requirements at once. The flexibility and simplicity of the Add-Paths solution [WRC09a] thus make it a good candidate for future deployment.

Chapter 5

Preventing the propagation of unnecessary BGP Withdraws

We showed in chapter 3 that the lack of path diversity in iBGP was responsible for several issues in today's routing. When a router looses its path to a destination, it must send BGP Withdraws to all BGP peers, and in particular to its eBGP peers. If those eBGP neighbors do not have alternate paths, they will also encounter transient losses of connectivity, and BGP Withdraws can be propagated even further. If all alternate paths were correctly disseminated to all routers, this problem would not occur as the failure would be recovered immediately and locally.

We analysed the problem of unnecessary BGP Withdraw propagation in chapter 3, and showed that part of the BGP Withdraws in the global Internet are caused by a bad propagation of alternate paths in iBGP. Preventing the dissemination of such BGP Withdraws would help reducing the global Internet churn. Also, avoiding unnecessary BGP Withdraw propagation would allow ISPs to improve the stability of the prefixes advertised by their customers in case of link failure. Furthermore, providing a solution to this problem is affordable, as it can be tackled within the iBGP organization.

When a full-mesh of iBGP sessions is used, it is easy to provide diversity to all routers. Diversity is blocked in a router when there is a better iBGP path (i.e. higher local preference, lower MED or shorter AS path) in the AS. If the advertisement rule is modified such that a router announces its best eBGP-learned path for each destination to its iBGP peers, up to one path per router is propagated in the AS. This is performed using the mechanism called Best-External Advertisement [MFCM10].

If an AS uses Route Reflection, it is also possible to prevent BGP Withdraw propagation. Using Best-External with Route Reflection is not sufficient, although the best external paths are advertised to Route Reflectors. They do not propagate this diversity further in the network because they only advertise one path per prefix. The Add-Paths solution proposed at the IETF modifies the BGP protocol for advertising several paths for each prefix [WRC09a]. This allows for a perfect diver-

sity propagation thus achieving the Withdraw-Blocking property, but this increases the memory usage and the number of BGP messages needed for exchanging those paths. For these reasons, it is probably not suitable for ASBR with limited resources.

In this chapter, we propose a solution that allows routers to propagate the information about the existence of alternate paths without major modification to BGP. The principle is that upon reception of a BGP Withdraw, a router that knows that an alternate path exists in the AS can wait until iBGP has converged before sending a BGP Withdraw over its eBGP sessions. The AS becomes thus Withdraw-Blocking without requiring its routers to store all BGP paths learned by the AS. This solution works purely in the control-plane, and can be deployed without modifying the FIB of the routers.

5.1 Tagging paths with diversity

The principle of our solution is that, whenever a router knows an alternate path, it attaches a `PATH_DIVERSITY` BGP community to the primary path when it advertises it to its iBGP peers, including the one from which the path has been learned if that path was learned via iBGP. This is needed because the router that sent the primary path also needs to learn the existence of the backup path.

The primary path is then distributed in the AS with the `PATH_DIVERSITY` community. Legacy BGP routers that do not support the `PATH_DIVERSITY` community simply propagate the path with the community following classical iBGP rules, without taking its signification into account. The `PATH_DIVERSITY` community is removed when the path is advertised over eBGP sessions.

When the primary path is withdrawn, all routers that support the community and do not have an alternate path themselves will not propagate the BGP Withdraw on their eBGP session. Instead, they start a timer and re-advertise the BGP Update message for the withdrawn path with a local-preference of 0 to their iBGP neighbors. We thus allow the path to remain temporarily in the routing table of the router. This does not prevent traffic losses, as explained later, but blocks the transmission of unnecessary BGP Withdraws. Also, the routers that do not support the `PATH_DIVERSITY` community will not remove the primary path when receiving the advertisement with the low local-preference and will not send BGP Withdraws before receiving the alternate path. With this local-preference value, alternate paths will always be preferred over the primary by routers that know them and they will be propagated in the AS [FDPF10].

If the timer expires and no alternate path has been received, the router sends the BGP Withdraw on its eBGP sessions. The timer is needed if the alternate path is withdrawn shortly after the primary, which can happen when both paths are impacted by the same failure. In that case, the alternate path cannot be propagated in the AS even if the `PATH_DIVERSITY` community has been attached to the primary path. The timer prevents the BGP convergence to be blocked, waiting for

an alternate path that does not exist anymore. A suitable value for the timer should be established by evaluating the iBGP convergence time. This value will typically depend on the type of iBGP organization of the AS, and the number of primary paths that can be impacted by a given eBGP link failure.

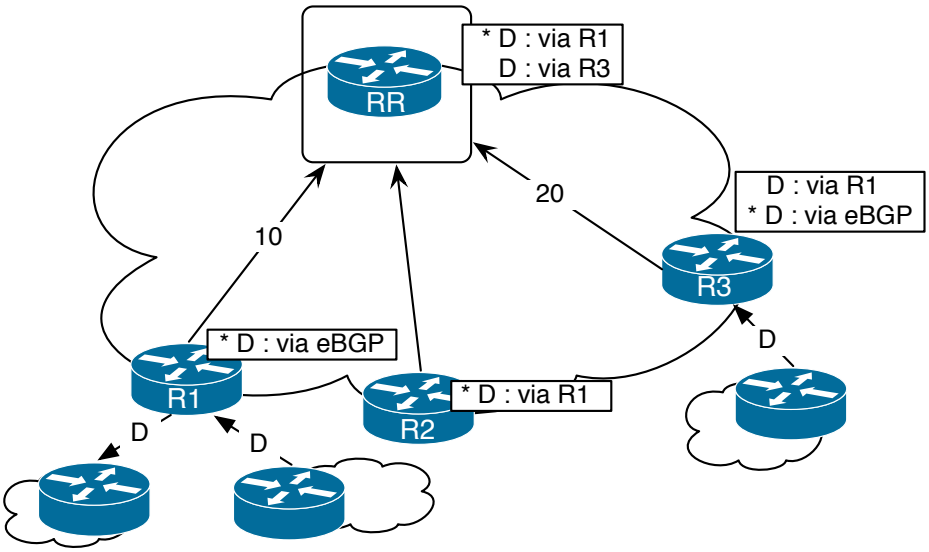


Figure 5.1: Route Reflection

In the example of Fig. 5.1, R3 tags the path learned via $R1$ with the community, and sends it back to the Route Reflector, which in turn advertises it to all its clients including $R1$. Thanks to this community, $R1$ knows that there exists an alternate path, and if the primary path is withdrawn, it will not send a BGP Withdraw on its eBGP sessions. Instead, it will wait until the Route Reflector advertises the alternate path via $R3$.

Alg. 1 BGP Update reception for a prefix P

```
1:  if BGP path received for prefix P then
2:      Run decision process for prefix P
3:      /* Check diversity */
4:      if alternate path exists in Adjribins then
5:          Tag diversity community to the best path for P, depending on the policies applied to the alternate
            path.
6:      end if
7:      if Best path changed then
8:          Propagate new best path on eBGP sessions and iBGP sessions (including the originator of the path)
9:      end if
10:     if Best path unchanged, but a community has been tagged then
11:         Advertise on iBGP sessions, including originator of the best path.
12:     end if
13: end if
```

5.2 Dealing with export policies

In chapter 3, we gave a set of conditions to ensure that an AS would not propagate unnecessary BGP Withdraw on its eBGP sessions. We called such AS Withdraw-Blocking. The conditions states that a valid export-policy compliant path must be available to all routers. In this chapter, we relax this condition by requiring the routers to know about the existence of such path.

However, the mechanism presented above is not sufficient to ensure that an AS is Withdraw-Blocking. Indeed, the first alternate path received during the convergence is not necessarily export-policy compliant with the primary one. In this case, the router will have to send a BGP Withdraw on the eBGP sessions over which that alternate path cannot be advertised. We refine our solution to face this issue by relying on two community values, `EPC_DIVERSITY` and `NON_EPC_DIVERSITY`. The procedure for tagging those diversity communities is explained in Algorithm 1: A router tags a path with either community depending on whether its alternate path is export-policy compliant with the primary or not. The export-policy compliance can be easily computed by a router if the paths are tagged with a community that identifies their origin [Mey06], i.e. if they come from a customer or from a peer or provider. This is a good practice rule that is often used. The router then re-advertises the path to its iBGP neighbors, including to the one from which it was learned.

Algorithm 2 is applied when a router receives a BGP Withdraw. The principle is that when a router receives a BGP Withdraw for a path tagged with one of the communities, and for which it does not have an alternate path, it does not send any BGP Withdraw over eBGP sessions. Instead, it waits until it receives that alternate path. If, during the convergence, a first alternate path that is not export-policy compliant is learned while the path is tagged with `EPC_DIVERSITY`, the router still waits for the export-policy compliant path instead of sending BGP Withdraws. When common policies are used [GR01], the export-policy compliant path will finally be selected as best, and no BGP Withdraw is sent over eBGP sessions.

On Fig. 5.2, RR1 knows an export-policy compliant path via RB, so it adds the community `EPC_DIVERSITY` to the BGP message. RR2 also has diversity for that path. As its alternate path is not export-policy compliant (this is a path received from a provider while the primary comes from a customer), $RR2$ also tags the community `NON_EPC_DIVERSITY`. All routers know that diversity is available, and the AS is Withdraw-Blocking. For example, if the link between RA and $R1$ fails, $R3$ has no diversity but knows that an export-policy compliant path is available, so it does not advertise a BGP Withdraw to its eBGP peer. When $RR2$ learns the failure via the IGP, it will not yet send a BGP Update for D with the path via RC, because it is not export-policy compliant. Instead, it waits until it receives the export-policy compliant path. Eventually, $RR2$ then $R3$ will learn the alternate path via RB, and $R3$ can send a BGP Update with the export-policy compliant path on its eBGP session.

Alg. 2 Withdraw reception for a prefix P or failure of a nexthop

```
1:  if BGP Withdraw received from eBGP session or BGP nexthop becomes unreachable then
2:      Run decision process for each impacted prefix
3:      if best path unchanged then
4:          check diversity for this prefix, update communities and re-advertise over iBGP if needed
5:      else
6:          if best path is tagged with EPC_DIVERSITY then
7:              if Export-policy compliant path available in Adj-RIB-Ins then
8:                  Propagate alternate path as new best path over iBGP sessions and eBGP sessions
9:              else
10:                 Set local-preference of the path to 0
11:                 Wait until export-policy compliant path is received, or timer expires
12:                 if Timer expires then
13:                     Propagate BGP Withdraw
14:                 else
15:                     Propagate alternate path as new best path over iBGP sessions and eBGP sessions
16:                 end if
17:             end if
18:         else if best path is tagged with NON_EPC_DIVERSITY then
19:             if alternate path is available in Adjribins then
20:                 Propagate alternate path over iBGP sessions and over policy-compliant eBGP sessions
21:                 Propagage BGP Withdraw over non policy-compliant eBGP sessions
22:             else
23:                 Set local-preference of the path to 0
24:                 Wait until any alternate path is received, or timer expires
25:                 if Timer expires then
26:                     Propagate BGP Withdraw
27:                 else
28:                     Propagate alternate path as new best path over iBGP sessions and over policy-compliant
                        eBGP sessions
29:                     Propagage BGP Withdraw over non policy-compliant eBGP sessions
30:                 end if
31:             end if
32:         else
33:             Act as usual
34:         end if
35:     end if
36: end if
```

5.3 BGP convergence

Using the proposed communities slightly increases the number of BGP messages exchanged during the initial convergence, as an additional BGP Update is emitted when the path is tagged with a diversity community. In the worst case, two additional BGP Updates will be emitted by a router, one when a non export-policy compliant alternate path is known to exist, and a second when the existence of an export-policy compliant alternate path is learned. Also, upon failure of an alternate path, a few BGP messages are also exchanged to update the communities of the primary path. However, those BGP messages will not be announced outside the AS, hence the message overhead is limited to the AS.

The diversity communities also do not impact routing stability: Tagging the diversity communities is a deterministic process that does not lead to routing loops. Indeed, when the paths to a destination are stable, once a diversity community has been tagged to a route, it is not removed as long as there is a corresponding alternate

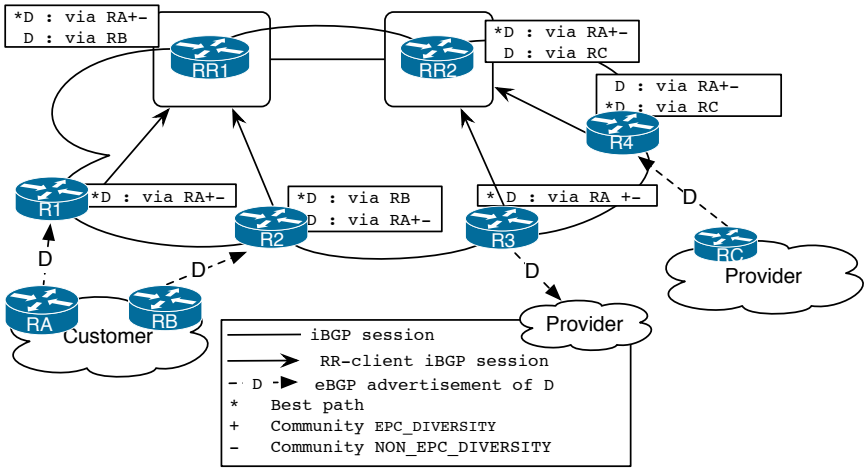


Figure 5.2: Announcing diversity in a community

path in the AS. Sending the tagged path back to the sender or the original path also does not result in routing loops. As a path is at most tagged twice, it is sent back to the sender at most twice and then the routing state becomes stable. Also, if the alternate path fails, the router that tagged the primary path stops re-advertising it with the backup community, and the tagged path is replaced by the original path in all routers after iBGP convergence.

5.4 Impact on the dataplane

When a router receives a BGP Withdraw for a destination and lacks an alternate paths, it doesn't have any nexthop in its FIB to which send the traffic, until it receives a new path from other routers. The traffic might then be dropped during the iBGP convergence. However, even if it cannot prevent the local loss of connectivity, waiting for the new path before sending the BGP Withdraw on eBGP reduces the losses of connectivity that would occur further in the Internet due to the unnecessary BGP Withdraw propagation.

5.5 Timer configuration

One question that remains open is the value of the timer that delays the propagation of eBGP Withdraws. This delay must be large enough to allow alternate paths to be propagated across the AS. For a single destination, the propagation delay has been studied by Wang et al. in their study of transient reachability losses [WGWQ05]. This delay is function of the MRAI settings in the AS, and of the number of hops between the alternate path and the router that needs it.

However, if the interdomain link that fails is a link with an important ISP (typically, a provider), lots of destinations can be impacted and be subject to the iBGP convergence. As all those BGP messages must be processed one after the other, there can be an long delay between the propagation of the alternate path for the first destination and the propagation of the alternate path for the last destination.

It is thus difficult to find a suitable value of the timer for all scenarios, as it is dependent on the iBGP topology, the MRAI values configured on the routers and the number of impacted destinations. Prospective measurements should thus be performed by ISPs wishing to deploy this solution. Indeed, timers too short lead to leaking eBGP Withdraws before receiving the recovery paths, and timers too long lead to unnecessarily delay the eBGP convergence that would provide alternate paths not yet available inside the AS.

5.6 Conclusion

In this chapter, we have presented a solution to prevent the propagation of unnecessary BGP Withdraw outside the AS when an alternate path is known. This solution ensures local recovery without the need to propagate recovery paths a-priori. Furthermore, it is purely control-plane, and could be deployed with a small update of routers software.

Chapter 6

On-demand provisioning of recovery paths

In chapter 3, we insist on the importance of providing alternate paths to routers for several reasons. One of them is fast recovery upon failure. When an interdomain link fails, three steps are necessary for the router to recover and be able to continue to forward packets.

First, the failure must be detected. For the router directly connected to the failed link, interface failure detection allows to detect the problem in about 10 milliseconds [FFEB05]. Other routers will be notified about the failure via the IGP if Nexthop-Self is not used. This typically takes about 200 milliseconds in today's routers [FFEB05]. If Nexthop-Self is used, the unreachable nexthop is only known by the adjacent border router, and the failure must be advertised in iBGP by explicitely withdrawing the paths learned over that link.

Second, an alternate path must be obtained from the control-plane. However, due to the way iBGP paths are propagated, alternate paths are not necessarily available to all routers, and an iBGP convergence may be needed at the time of the failure in order to propagate them. BGP convergence is slow, and packet are lost as long as the alternate path is not yet available. Two studies of packet flows upon BGP convergence have enlightened the impact of BGP on end-to-end performance for customers [KKK07][WGWQ05]. Providing alternate paths to BGP is thus a key point to reduce recovery time for the routers.

Finally, when the alternate path is available in the BGP RIB, the Forwarding Information Base must be updated accordingly. Historically, FIB databases were 'flattened': When translating the content of the RIB into the FIB, the BGP nexthop of each prefix is immediately matched with the corresponding interface in the corresponding FIB entry [Zin02]. This is summarized in figure 6.1. The BGP FIB is built by resolving the BGP nexthop of each prefix in the IGP FIB to obtain the corresponding interface. This interface is then inserted in the BGP FIB, as schematized. Thus, a single lookup is sufficient to obtain the outgoing interface for a given prefix. However, in case of link failure impacting a lot of destinations, the FIB en-

try of each of those destinations must be updated. This implies that for each prefix, the BGP decision process must be run to select a new best path, and that new best path must be installed in the FIB. In the example, if $X1$ fails and must be replaced by an alternate nexthop $X2$, all three entries represented must be modified accordingly. Thus, the FIB recovery time scales with the number of prefixes affected by the failure, and can last up to ten seconds for 100,000 prefixes [BFF07].

IGP Table

IP	Interface
X1	West via X3
X2	West via X3 East via X5
X3	West
X5	East
RR	East via X5

BGP Table

Prefix	Nexthop	Interface
10.0.0.0/8	X1	West
11.0.0.0/8	X1	West
12.0.0.0/8	X1	West
....		

Figure 6.1: Flattened FIB architecture

In this chapter, we explore how it is possible to tackle failure recovery with fast rerouting. Traffic rerouting can be performed directly by the border router adjacent to the failure, or globally, i.e. by all BGP routers using the failed link.

As explained earlier, the duration of the recovery depends on two factors: The availability of an alternate nexthop or exit point, and a way to quickly update the FIB entries. First, we present the differences between the two possible ways of performing fast rerouting. Then we review the solution of Bonaventure et al. [BFF05] which uses a modified FIB architecture and precomputed per-link protection tunnels to redirect the traffic.

We then explain in the third section that it is now possible to perform a scalable per-prefix modification of the FIB upon failure, which means that the solution of Bonaventure et al. can be improved to offer a per-prefix protection instead of per-link protection. We propose such a solution in the fourth section, by using additional iBGP sessions. Finally, in the fifth section, we show that the establishment of those additional sessions can be automated, which greatly reduces the configuration burden for the operators.

6.1 Fast rerouting upon failure

6.1.1 Rerouting performed by the egress router

The most predictable source of alternate paths for failure recovery is neighbor multi-connectivity. Multiple links between two neighboring ASes provide path redundancy when the same destinations are consistently advertised over the corresponding eBGP sessions. Consequently, this path diversity can be used to reroute traffic in case of failure. Local routers having eBGP sessions with a neighbor are

egress routers, or exit points, for the traffic destined to the destination advertised by the neighbor. Upon failure of its link with the neighbor, an egress router is the first to realize that the traffic needs to be rerouted to an alternate exit point. Other routers using this exit point have to wait for the IGP convergence in the best case (Nexthop-Self not used), or the BGP convergence in the worst case (Nexthop-Self) before being notified about the failure and finding an alternate path. Thus, if the egress adjacent to the failure is able to deviate the traffic to the alternate exit point during the re-convergence, traffic loss is minimized.

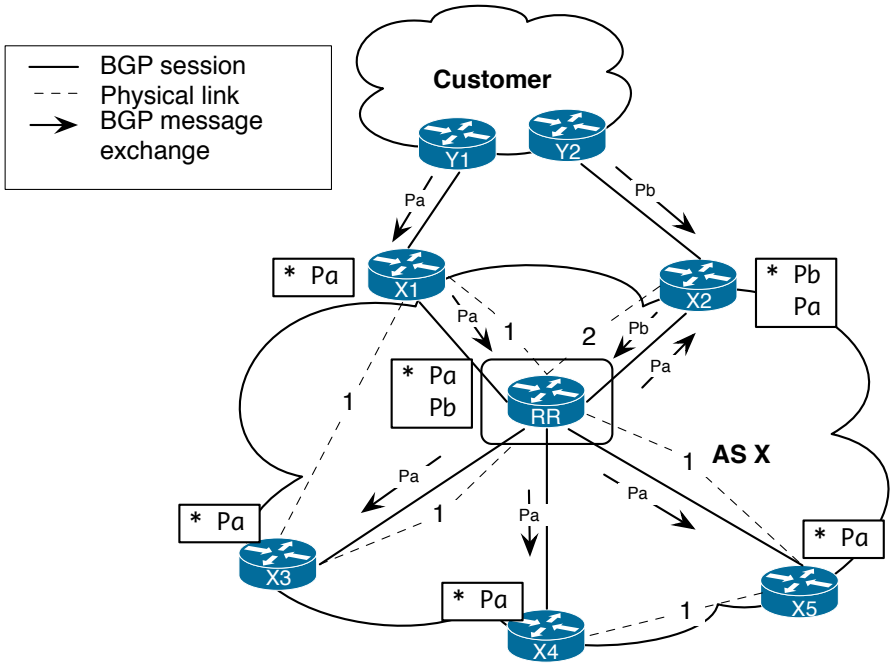


Figure 6.2: Example : BGP convergence

In the example of figure 6.2, two paths Pa and Pb are advertised to ASX by its customer ASY. Due to **Route Reflection Path Loss**, most routers only know path Pa. The resulting traffic flows are shown in figure 6.3: nearly all border routers use $X1$ as exit point to forward the traffic destined to the customer. Only router $X2$ uses its best path Pb to reach the destinations of the customer.

Upon failure of the link $X1$-$Y1$, $X1$ is the first router to detect the problem. Depending on whether Nexthop-self is used, other routers must wait for the IGP convergence or the BGP convergence to be notified about the loss of the path. In the meantime, they still forward packets to $X1$. As Pa is no longer available, router $X1$ cannot forward these packets to the destination. Instead of dropping those packets, $X1$ should reroute them to the alternate nexthop $X2$, such that they can still be forwarded to the destination, as shown in figure 6.4.

It is however still possible that the rerouted traffic be deflected by intermediate

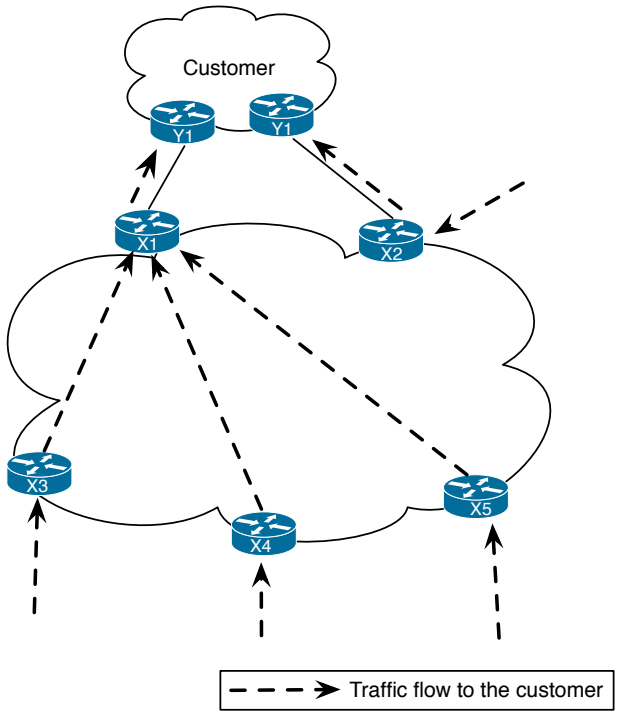


Figure 6.3: Example : Traffic flows

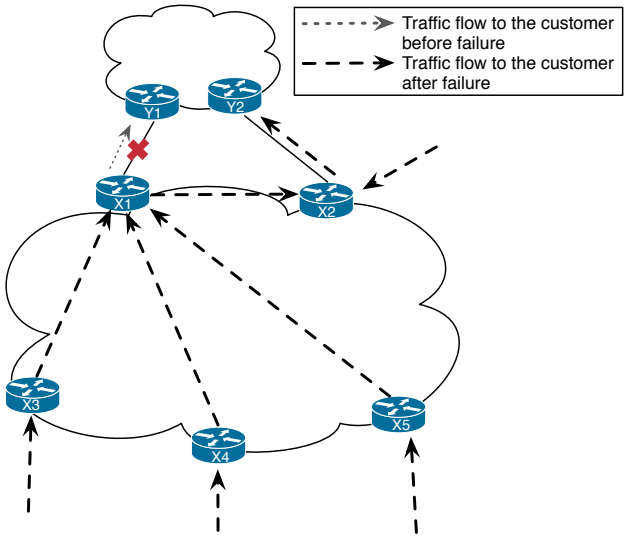


Figure 6.4: Traffic rerouting by the egress router upon failure

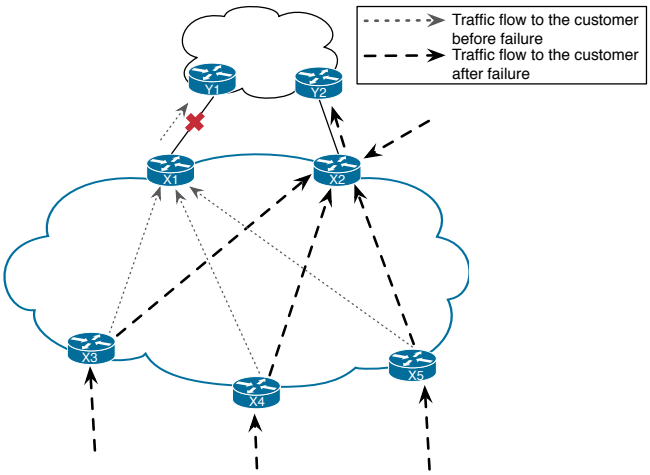


Figure 6.5: Traffic rerouting by the ingress router upon failure

routers between the two egress points. In the example, the Route Reflector is on the forwarding path from $X1$ to $X2$, but still uses the failed path Pa. The traffic is deflected back to $X1$, and there is a transient forwarding loop. The alternate egress router itself could send the traffic back to the primary one, if the alternate path is not its best one. This could occur for example if path Pb has an lower local-preference than path Pa, in which case Pa would be selected as best by $X2$.

In order to prevent such deflections or loops, the primary egress router should encapsulate rerouted packets toward the outgoing interface of the alternate egress router. This way, no other router performs a lookup in the routing table based on the destination prefix, and traffic correctly exits the AS.

6.1.2 Rerouting performed by the ingress router

Performing rerouting at the egress router is efficient in case of link failure. However, in case of node failure, the egress router is down and traffic is lost. In order to minimize traffic disruption, rerouting must be performed sooner, i.e. when the traffic enters the AS. This is illustrated in our example on figure 6.5: Upon failure of $X1$, the *ingress routers* receiving traffic for the customer should stop forwarding it to the egress router $X1$, because it cannot perform neither routing to $Y1$ nor rerouting to $X2$, and traffic will be lost. Thus, to minimize traffic loss, the ingress routers must know the alternate nexthop $X2$ in advance. Performing rerouting at the ingress router also has the advantage that the traffic is directly sent to the alternate nexthop, while with egress rerouting, the traffic is first sent to the primary nexthop, then deflected to the alternate nexthop. Ingress rerouting minimizes the cost of transiting traffic through the network compared to egress rerouting.

6.2 Per-link protection against link failure with protection tunnels

Bonaventure et al. have proposed in [BFF05] a mechanism to protect a neighbor against interdomain link failure, by rerouting all the traffic to the failed nexthop via a tunnel to another link with the same neighbor. This is thus a solution using the principle of egress router rerouting presented in the previous section. Their primary motivation is that there are lots of interdomain link failures, and that they are usually short-lived [BFF05]. However, the resulting BGP convergence can be long, as it sometimes impact lots of destinations. Furthermore, even when a backup path is available, replacing the primary nexthop by the backup nexthop in the FIB takes time. The authors used blackbox measurement to evaluate the FIB update time on several router models from different vendors. They found that it requires between one hundred and a few hundred of microseconds on average to update one entry in their FIB. When several hundreds of thousands of prefixes must be updated, the total duration of the FIB update can thus last for several tens of seconds. [BFF07].

The authors thus propose to pre-establish a protection tunnel to a backup link. This tunnel can be used to temporarily redirect all the traffic in case of link failure without having to wait for the update of all the impacted destinations in the FIB.

6.2.1 Description

The solution of Bonaventure et al. consists in four different steps. First, the primary egress router must locate an alternate egress router. Second, it must modify its FIB to pre-install the tunnel to the alternate exit point as backup outgoing interface for the protected nexthop. Third, upon failure, a single pointer of the FIB is modified to redirect all traffic to the protection tunnel. And finally, a graceful shutdown solution is applied to handle the BGP convergence of the other routers without traffic disruption.

Locating alternate egress points

The first step in the establishment of the protection tunnel is to locate the backup link. The backup egress router must be connected to the same eBGP neighbor, thus, the primary egress router needs to obtain the list of routers connected to the same neighboring AS. For this, a new BGP Update message is defined to allow the advertisement of the characteristics of currently active eBGP sessions. These messages are only advertised inside the local AS, and contain the following information :

- The Network Layer Reachability Information (NLRI) is the IP address of a loopback interface on the router that originates the path. This ensure the uniqueness of the NLRI, such that the path cannot be hidden by Route Reflectors.

- the AS Path only contains the AS number of the eBGP neighbor
- a tunnel attribute containing the parameters of the protection tunnel to be established
- an extended community value used to encode the Shared Risk Link Group of the eBGP session. This prevents using as backup link a link that risk to suffer from the same failure as the primary.
- an extended community value used to encode the BGP policy applied on the eBGP peering session. This is needed to prevent using, as backup link, a link on which a different set of destinations is advertised.

In the example of figure 6.2, $X1$ and $X2$ will advertise such BGP messages containing information about their eBGP session with ASY. $X1$ will thus select the link $X2$-$Y2$ as backup link, and use the information contained in the BGP message of $X2$ to configure the protection tunnel.

Installing a protection tunnel in the FIB

To prevent the update of the FIB entries of all prefixes advertised on the protected link upon failure, the FIB must be organized such that a single operation is sufficient to redirect all the traffic in the protection tunnel. This is not possible with the flattened FIB organization presented in the beginning of the chapter. A better FIB organization exists, and consists in using pointers between the BGP table and a nexthops table mapping the IP nexthops with the corresponding interface.

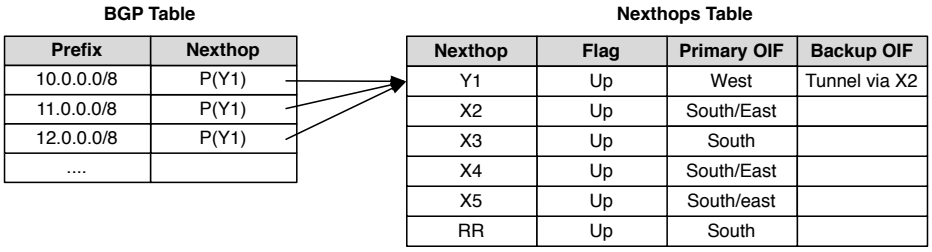


Figure 6.6: Improved FIB organization

Bonaventure et al [BFF05] show that with this organization, it is possible to perform fast rerouting, as only one entry in the routing table must be updated instead of modifying all prefix entries. In the Nexthops table, each entry contains the outgoing interface corresponding to the nexthop, as well as a backup interface to be used when the primary is down. This interface can contain information about the protection tunnel. Figure 6.6 represents such a FIB organization for router $X1$ of the network of figure 6.2. On the left, the BGP table only contains the mapping between each prefix and its BGP nexthop. The BGP nexthop is itself a pointer to

the Nexthops table, which contains information about the reachability of each BGP nexthop. In the case of the destinations advertised by ASY, the primary interface to $Y1$ is up, and a backup interface is available. This backup interface is the tunnel toward the alternate egress router $X2$, which also has a link with ASY. Upon failure of the link $X1-Y1$, the router simply sets the flag of the nexthop $Y1$ to Down, which means that packets won't be forwarded on the North interface anymore, but will be encapsulated and sent through the tunnel to $X2$

Traffic rerouting upon link failure

Upon failure of a nexthop, the router simply sets the corresponding entry in the Nexthop table to "Down", and the traffic is forwarded to the backup outgoing interface, i.e. the protection tunnel in our context. As a lot of interdomain link failures are short-lived, the router must be careful not to start the BGP and IGP convergence immediately, as some routers might be left without a valid path to the protected destination. This is the case in figure 6.2, where three routers only know the path Pa via $X1$. When the link $X1$-$Y1$ fails, if they learn the failure via the IGP, they will drop their traffic to ASY even though local rerouting at $X1$ is available.

If the failure is short enough, the peering link comes back when the protection tunnel is still effective, and the primary router simply updates its FIB entry to set the primary outgoing interface to "up" to stop using the tunnel. However, with the protection tunnel, packet flow is not optimal anymore in the network, as the traffic is deflected in the tunnel. This solution cannot be used in the long-term, because it could create congestion. Thus, after a certain time, if the link is still down, the router will trigger the advertisement of the failure in the IGP and in iBGP to allow ingress routers to switch to the backup nexthop instead of using the protection tunnel.

Graceful shutdown of the interdomain link

As mentioned earlier, the advertisement of the failure in the network can disrupt packet forwarding when some routers lack path diversity. When the failure is advertised in iBGP (Nexthop-Self is used), it is possible to prevent dataplane disruption while enabling the propagation of the alternate nexthop to all routers.

For this, Bonaventure et al. propose to allow the primary egress router to readvertise the failed destinations with the lowest local-preference (i.e. 0), to indicate that the path will be removed later. Routers with diversity for those destinations will thus prefer their backup paths, and advertise them to all other routers. Routers without diversity will continue to use the old path leading to the protection tunnel until they receive an alternate nexthop.

6.2.2 Analysis of the solution

This solution exploits the egress-based rerouting principle presented above to build a protection tunnel between a primary egress router and a backup session with the same neighboring AS. This solution is also exploitable in the case of dual-homed stubs having only one eBGP session with each provider. However, the solution assumes that all prefixes are similarly advertised on the primary and backup links. If this is not the case, some packets can be send on the backup link even though their destinations were not advertised on the corresponding eBGP session. Also, this solution is only applicable for protection against link failure. Protected destinations are still vulnerable to node failures, i.e. the failure of the egress node itself.

In the rest of this chapter, we propose to re-use the principle of this solution, but using a per-prefix protection instead of a per-link protection. In addition, we also propose a way to provide the alternate nexthop to the ingress router a-priori, such that normal convergence must not be delayed.

6.3 A hierarchical FIB to allow per-prefix fast rerouting

The FIB modification of Bonaventure et al. allows to quickly change the nexthop interface. However, it assumes that all rerouted destinations were advertised on the backup link. This is not necessarily the case. In this section, we present the Prefix Independent Convergence (BGP PIC) mechanism developed by Cisco [Fil07]. It consists in a hierarchical FIB architecture that allows to change the nexthop of each prefix to the corresponding backup nexthop in a scalable way. We also evaluate the scalability of this solution in the ISP A network.

6.3.1 Hierarchical FIB

Similarly to the FIB organization shown in figure 6.6, hierarchical FIB uses pointers to translate the dependencies between FIB elements, instead of resolving them to build a flat FIB. The hierarchical FIB is shown in figure 6.7. It generalizes this principle by using several levels of indirection. First, each prefix has a pointer to a data structure called *BGP Path List*. This BGP Path list contains the list of available nexthops. If Multipath BGP is used [mul], this means that the traffic can be load-balanced across all nexthop available in the BGP Path List. BGP Path List are shared across prefixes: Two destinations with the same nexthop(s) will point to the same Path List. Then, each Nexthop in the BGP Path List points to an IGP Path List, which represents the IGP Nexthops used to join this nexthop. If Equal Cost Multi-Path (ECMP) is enabled, an IGP path list can contain several IGP Nexthops. Then, each IGP nexthop points itself to the corresponding outgoing interface. A Linked List of dependent BGP Path List must be maintained in each IGP Path List, to be able to update the BGP Path List upon failure of the IGP Nexthop.

As such, hierarchical FIB allows to immediately recover from an intradomain failure after the IGP convergence by simply updating the IGP Path Lists entries, in-

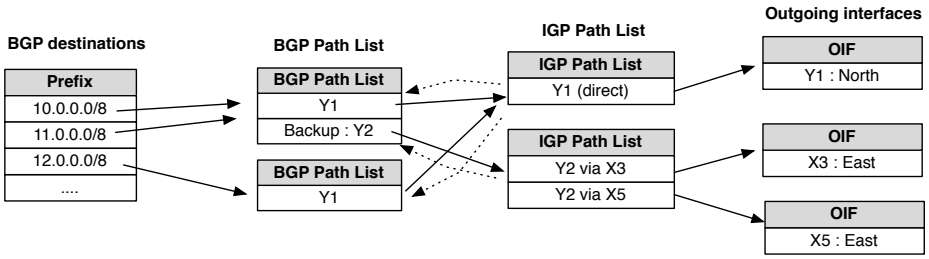


Figure 6.7: Hierarchical FIB organization

stead of having to update all prefix entries in a flattened FIB. But this organization can also support fast recovery upon interdomain link failure, by allowing a BGP Path List to contain a list of alternate nexthops. Thus, when BGP runs its decision process, in addition to select the best path(s) for each prefix, it must also select one or several backup paths, and install them into the corresponding BGP Path List in the FIB. Figure 6.7 show the hierarchical FIB of router $X1$ for the destinations advertised by ASY. Let's assume that $X1$ uses a mechanism to obtain the alternate path via $Y2$ for two of the three prefixes represented in the schema. It will thus build two BGP Path Lists in the FIB: One containing only the primary Nexthop $Y1$ and a second containing the primary Nexthop $Y1$ and the backup Nexthop $Y2$. Then, it installs the corresponding pointers between each prefix and its corresponding BGP path list. Upon failure of the $X1-Y1$ link, the North interface is set down, and the IGP Path List for $Y1$ is removed. Before removal, the router scans the Linked List of BGP Path Lists of the IGP Path List to update the BGP nexthop accordingly. This means that, in this case, the BGP Path List with $Y1$ alone is removed and destination $12.0.0.0/8$ becomes unreachable, and that the other BGP Path List looses its primary Nexthop $Y1$ and switches on the backup Nexthop $Y2$. Upon failure of the interdomain link, only two BGP Path Lists were updated while three BGP prefixes were impacted.

6.3.2 Scalability of BGP Path List sharing

One of the scaling factor of the BGP PIC hierarchical FIB is the number of BGP Paths List that have to be maintained in the FIB, and updated upon failure of a link. The number of BGP Paths List depends on the number of BGP nexthops used but also on the number of paths from the RIB that are placed in the hierarchical FIB. We take the most common scenario where Multipath BGP is not used, and there is thus only one primary Nexthop and possibly a backup Nexthop. In theory, in a network containing n BGP nexthops, there could be up to $n \times (n-1)$ BGP PL in such a scenario.

In the GEANT network, this translates into a maximum of 506 BGP Paths List, and in the ISP A, into a maximum of 11556 BGP Path List. However, in practice only a small fraction of all pairs of BGP nexthop will appear as BGP Path Lists.

Based on the paths collected in the iBGP Full-Mesh of GEANT, we obtained a total number of BGP Paths Lists of 54 per router. In the case of the ISP A, we analysed the Adj-RIB-Ins of each Route Reflector and obtained between 423 and 645 Paths Lists.

Upon the loss of a specific IGP Paths List, BGP PIC walks the linked list of the associated BGP Paths Lists. The scaling factor in case of link failure is the number of these associated BGP Paths Lists. For the ISP A, we computed for each Route Reflector the number of BGP Paths Lists impacted by the failure of each BGP nexthop. In the most impacted Route Reflector, the number of BGP Paths Lists modified by each link failure is usually much smaller than 10, with the percentiles 10, 50, 90 and 100 being respectively 0, 2, 5 and 85 impacted BGP Path Lists. In the case of GEANT, most of the failures impact 1 or 2 BGP Path Lists, with a maximum of 5.

Thus, in the ISP A, there is at most one hundred Paths Lists that must be updated upon a link failure. The implementors of BGP PIC performed testbed measurements, and report that only 7 microseconds are necessary to modify a single BGP Paths List upon failure of a Nexthop [Fil07]. Thus, in the case of ISP A, in the worst case, it would take 0,7 milliseconds to update the FIB after the detection of a link failure and either redirect the traffic to the alternate nexthop of each prefix or drop it if no alternate path is available.

6.3.3 Control-plane convergence after recovery

Even though the traffic is quickly rerouted to the alternate nexthop upon failure, thus enabling fast dataplane recovery, the control-plane must still be notified about the failure of the link to invalidate all the corresponding BGP advertisements. Such BGP convergence potentially impacts a large number of destinations, and is thus a slow process. However, this is hitless and does not lead to any performance degradation as the traffic is already rerouted to the pre-installed backup nexthop.

6.3.4 Perspectives with BGP PIC

Thanks to BGP PIC hierarchical FIB, a router can quickly recover from intradomain and interdomain link failures, provided that sufficient path diversity is available. For intradomain failures, path diversity is provided by careful design of the network. But for interdomain link failure, the availability of alternate BGP nexthops depends on the BGP path propagation, which is known to be poor when Route Reflection is used. Thus, to be able to fully exploit the features of BGP PIC, BGP path diversity must be increased. A solution such as Add-Paths can be used to this end, but requires to propagate a lot of additional paths across the network. In the next section, we propose a lighter alternative that can be deployed locally or globally.

6.4 Per-destination protection against failure

6.4.1 Fast rerouting upon link failure

The solution of Bonaventure et al [BFF07] uses per-link protection against failure, because updating all FIB entries is too long with a flat FIB organization. They thus propose a FIB organization that provides a tunnel to an alternate link as backup interface for the protected link. However, with BGP PIC, it is now possible to update all prefix entries in the FIB much more quickly. Instead of providing a backup interface (the protection tunnel) along with the primary outgoing interface to the nexthop in the nexthops table, this new FIB organization allows to provide a backup nexthop to each prefix. Upon failure, it is possible to quickly change the pointer from the primary nexthop to the backup nexthop in a scalable way. Thus, provided a backup nexthop is available in the Adj-RIB-Ins, per-prefix protection is available. The router can quickly react to the failure by forwarding packets to the legitimate BGP alternate nexthop of each prefix instead of using a tunnel to blindly redirect traffic to a backup link.

However, we have shown in chapter 3 that iBGP path propagation cannot guarantee that backup nexthops are available for all prefixes, even when such nexthops exist in the AS. But with egress rerouting, only one router needs an alternate path to minimize losses in case of link failure. It should be relatively easy to provide the required diversity to that single router without changing the iBGP organization or the BGP protocol. This can be performed by establishing an additional iBGP session between the primary egress router and the alternate egress router to exchange the paths of the protected neighbor.

A backup iBGP session differs from a normal iBGP session by the following:

- It is unidirectional: the router initiating the backup iBGP session does not advertise any path over this session. In the example of figure 6.2, router $X1$ would establish a backup iBGP session with router $X2$ to learn the paths that it received from the customer. $X2$ will probably also establish a backup iBGP session with $X1$. We keep these two sessions separate and allow the utilization of multiple BGP sessions between two routers as proposed in [AS05]. Furthermore, if $X1$ also needs a backup iBGP session with $X2$ to receive the paths learned from another AS, then a second backup iBGP session is established by $X1$ with $X2$. This allows router $X2$ to use a common export filter on all the backup iBGP sessions over which it advertises the paths from each AS. This significantly decreases the amount of processing required by the egress router and improves the scalability of the solution.

- The router that initiates the backup iBGP session uses the BGP Outbound Route Filters defined in [CR08] to indicate that it only wants to learn the paths received from a given AS. Those paths can be identified by an extended community indicating the eBGP session on which they were learned.

- The paths learned over a backup iBGP session are not used for the best path selection and are never advertised to iBGP or eBGP peers

As such backup sessions only carry the paths received from a given eBGP neighbor, i.e. a subset of the routing table, we call them *Lightweigh iBGP sessions*, or *libgp sessions* in short.

In our example, when $X1$ establishes a liBGP session with $X2$, it indicates that it only wants to receive the paths from the customer AS on this session. Then, $X2$ advertises all the prefixes learned from $Y2$ over the backup session with $X1$.

If the alternate egress router does not use its eBGP path as best path, typically because it has a lower local-preference, Best-External advertisement [MFCM10] must be configured on that router to ensure that the backup path is advertised on the backup iBGP session. This also requires traffic encapsulation upon failure of the primary link to guarantee that the traffic is rerouted to the backup link even when the alternate egress router isn't using this link itself.

6.4.2 Fast rerouting upon node failure

BGP PIC and the liBGP session allow the egress router to immediately redirect the traffic in case of failure of the interdomain link. However, it is not useful upon failure of the egress router itself. Node failure can be handled by using ingress router rerouting. A naive solution would be to allow each ingress router that uses a path learned from an egress router to create an automatic backup iBGP session with another router attached to the same AS. However, such a solution is not scalable as an ingress router would need an automatic backup iBGP session with one router attached to each peer AS in addition to its normal iBGP sessions. This would largely increase the number of iBGP sessions that need to be maintained and the memory consumption on each ingress router.

Instead of distributing alternate paths, a better solution is to distribute to ingress routers alternate nexthops for each BGP nexthop that needs to be protected from a node failure. For this, we rely on the fact that the egress router has learned an alternate path over its automatic backup iBGP session. We propose a new non-transitive extended community called $BACKUP\_NEXTHOP$ that egress routers use to encode the IP address of the alternate nexthop. When a BGP router that needs to be protected against node failure advertises a BGP message over a normal iBGP session, it encodes the $BACKUP\_NEXTHOP$ extended community along with the primary nexthop advertisement. This community contains the IP address and the encapsulation label of the path learned over the automatic backup iBGP session. The advantage of using extended community values is that they are already supported by existing BGP routers and thus can be transported by Route Reflectors transparently. This is important in comparison with BGP extensions such as Add-Paths [WRC09a] or the solution of Bathia et al. [Bha03] that require upgrades of RRs. Furthermore, the extended community values only require 8 bytes of memory per path and do not require additional iBGP sessions.

In our example, $X1$ receives alternate paths via $X2$, along with a label used to encapsulate traffic directly to the egress link. It will thus add a special community $BACKUP\_NEXTHOP$ containing that label and the alternate nexthop $X2$ to its BGP Updates when advertising the destination of the customer on its regular iBGP sessions. Upon failure of $X1$, the ingress routers detect the failure via the IGP, and check whether a $BACKUP\_NEXTHOP$ community was learned along with the failed nexthop for the impacted destinations. They will thus reroute their traffic towards $X2$, using the proper encapsulation label.

This advertisement of alternate paths along with a primary nexthop also suppress the need for using the graceful shutdown of the failed path presented in the solution of Bonaventure et al. [BFF05], as the ingress router can immediately switch on the backup nexthop upon notification of the failure. This also solves the problem of the dataplane disruption when Nexthop-Self is not used in the network.

6.5 Automating liBGP sessions establishment

Adding iBGP sessions manually is a known source of misconfiguration, especially when the iBGP organization is complex. This is typically the case with a Full-Mesh, where all routers must be reconfigured when a new iBGP peer is added to the network. Raszuk et al. proposed a method to automatically establish iBGP sessions by flooding BGP peering information with the IGP protocol [RAM03].

Based on this idea, we can also automate the establishment of the liBGP sessions. In this case, the information needed by a router for protecting an interdomain link with our method is the list of routers having an eBGP session with the same neighbor. However, opposite to the iBGP auto-mesh solution [RAM03], we do not rely on the IGP protocol. Once again, we build on the idea of Bonaventure et al.[BFF05] summarized in section 6.2 and use the new type of BGP information called **eBGP peering**. It is used by routers to advertise their currently established eBGP peering sessions.

Each router advertises one eBGP peering message for each established eBGP session. An eBGP peering message for liBGP contains the following information and is only advertised inside the local AS:

- the Network Layer Reachability Information (NLRI) is the IP address of a loopback interface on the router that originates the route
- the AS path only contains the AS number of remote AS on the current BGP session
- the BGP nexthop is set to the IP address of a loopback interface on the router that originates the route
- an extended community value that is used to mark all the paths that have been learned over this eBGP session, such that they can be selectively advertised on the corresponding liBGP sessions.

- an extended community value used to encode the Shared Risk Link Group of the eBGP session as in [BFF05]

- an extended community value used to encode the BGP policy applied on the eBGP peering session as in [BFF05]

This BGP extension is thus used to auto-discover the active eBGP peering sessions. The eBGP peering paths are never installed inside the FIB of routers.

To facilitate the propagation of the eBGP peering information, network operators can use a dedicated router that we will call **Contact Information Server**, or **CIS**. This CIS will gathers the information about which eBGP sessions are attached to which router. In practice, a simple Route Reflector can act as a CIS, as it only needs to reflect a number of paths equal to the number of eBGP peerings. There can be more than one CIS in the AS to ensure robustness.

In order to select a provider of path diversity, a router must consider the SRLG and policy attributes of the eBGP session of the provider, to prevent common failures and optimize the similarity of the backup paths with the primary ones, then select the closest alternate nexthop among the egress point corresponding to those criteria. This allows to minimize the impact of the routing deflection occurring upon rerouting on hot potato routing. Once a router has selected is best diversity provider among the routers peering with the neighboring AS to protect, it can establish its liBGP session with it and obtain the alternate paths needed for its protection objective.

6.6 Evaluation

6.6.1 Requirements for the protected neighbor

First, this solution is applicable only for neighbors that are multi-connected to the AS. Second, the efficiency of using automatic backup iBGP sessions is dependent on the symmetry of prefix announcements of each neighbor on its eBGP sessions with the local AS. If the neighbor announces all its prefixes on all sessions, the solution allows to protect all those prefixes.

However, if the announcements are asymmetric, i.e. some prefixes are advertised on one session and not on the others, all destinations cannot be protected in case of failure of the interdomain link or the corresponding egress router. Thus, if a customer requires a fast recovery service, it should ensure that the same prefixes are advertised over all its links. Traffic engineering can still be performed by using AS-Path prepending, MED or BGP community values to ensure that some links are preferred over others for some prefixes. Also, this solution requires to have parallel links to the protected neighbor.

6.6.2 Overhead of the solution

Using the liBGP sessions to provide diversity to the egress routers for their eBGP prefixes increases the size of the Adj-RIB-Ins, as each egress router received one backup path for each prefix learned over its eBGP sessions. Based on the data available for the ISP A presented in chapter 3, we computed the overhead for each router if additional backup sessions are established to protect all multi-connected eBGP neighbors. At the time of data collection, this Tier-1 AS peered with slightly more than 200 neighbor ASes that advertised about 180,000 different prefixes on about 500 BGP sessions.

Number of backup iBGP sessions

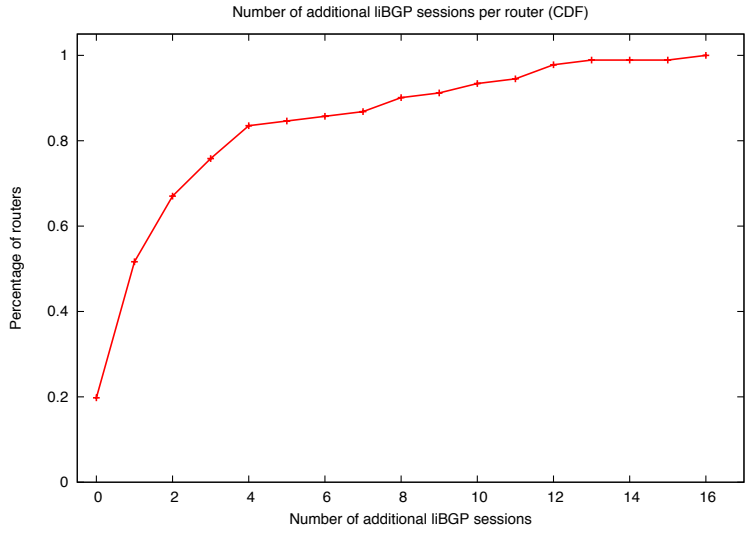


Figure 6.8: Distribution of the number of additional liBGP sessions per router in ISP A

For each router of the Tier-1 AS, we computed the number of its eBGP neighbors for which there is another egress point in the network. This number of eBGP neighbors is equal to the number of additional backup iBGP sessions required for that router.

The total number of backup liBGP sessions that have to be established in the Tier-1 is 256. The distribution of the number of additional backup sessions by router is shown on figure 6.8. The maximum number of liBGP sessions that an egress router needs is 17, but for the majority (80%) of the routers, at most four additional sessions are needed. The cost in terms of additional iBGP session is thus reasonable.

Size of the Adj-RIB-Ins

The number of additional paths in the Adj-RIB-Ins of a border router depends on the number of prefixes advertised by its eBGP neighbors for which it needs to establish an additional liBGP session. Based on the assumption that an eBGP peer advertises the same prefixes over all eBGP sessions with an AS [FMR04], we compute the additional number of paths in the Adj-RIB-Ins as the number of paths learned over eBGP sessions for which an additional liBGP session is established. Then, re-using the results of the simulations performed in chapter 3, we compute the resulting Adj-RIB-Ins sizes for each router.

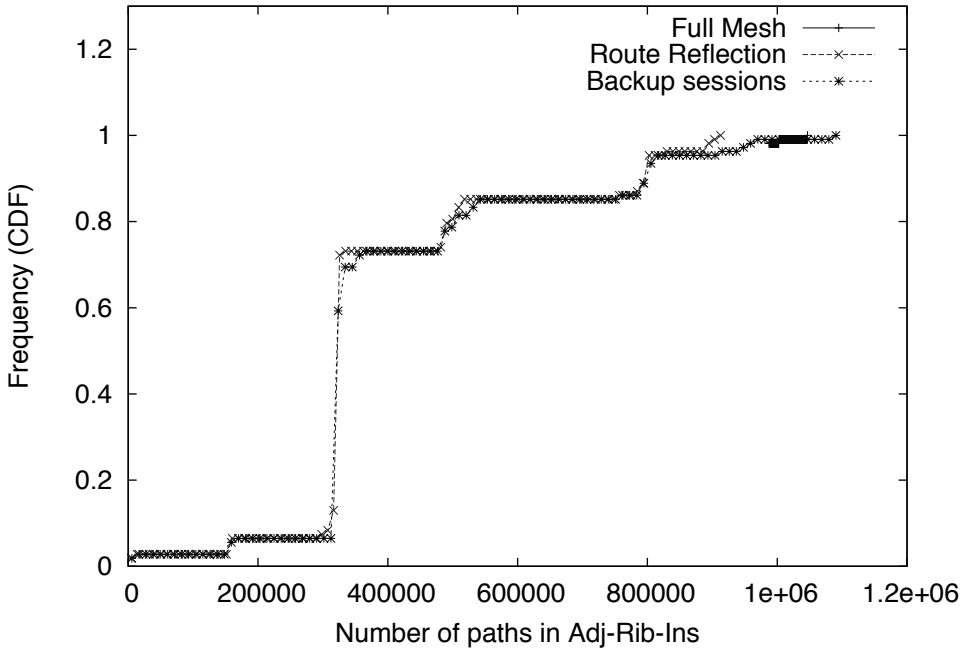


Figure 6.9: Cumulative distribution of the number of paths in the Adj-RIB-Ins

The results are shown on figure 6.9, along with the Adj-RIB-Ins sizes in the original organization and in a Full-Mesh. The curves represent the cumulated distribution of the number of paths per router. Clearly, the curves with and without backup sessions are very close to each other: the additional number of paths brought by the backup sessions is negligible for most routers. Only for a few routers, the total number of paths goes beyond one million paths, with a 200,000 paths increase compared with the maximum number of paths with Route Reflection. Those routers that probably have eBGP sessions with BGP peers advertising full routing tables. The total increase of paths in the whole ISP is shown in table 6.1, and is less than 2% compared to Route Reflection.

iBGP organization	Total of Adj-RIB-Ins paths
Full-Mesh	105.72×10^6
Route Reflection	43.7×10^6
Backup sessions	44.5×10^6

Table 6.1: Sum of the numbers of paths stored in ISP A's routers

6.7 Conclusion

In this chapter, we have explored several aspects of Fast Recovery mechanisms upon interdomain link failure. We build on the protection tunnels of Bonaventure et al. [BFF07] and on the PIC hierarchical FIB [Fil07] to propose a Fast Reroute mechanism that can be applied by the egress router as soon as the failure has been detected. This solution also allows to propagate backup paths along with the primary via a dedicated community, such that the ingress routers can also redirect traffic as soon as they learn about the failure. The subsequent iBGP convergence is thus hitless on dataplane performance, because the traffic has already been re-established. In addition to the recovery mechanism, we also introduce a mechanism allowing to automatically establish the liBGP sessions of our solution.

Chapter 7

Automated iBGP organization

In the previous chapter, we proposed a method to provide fast failure recovery when there are parallel links to the neighbor AS. Our solution relies on additional liBGP sessions on top of the existing iBGP organization to improve the number of recovery paths. However, when it is possible to design or re-design a network, the same principles can be re-used to build a brand-new iBGP organization. In this chapter, we present such a new iBGP organization, based on lightweight iBGP sessions on which only the eBGP paths received by a given eBGP neighbor are exchanged. The most interesting property of this solution is that, similarly to the liBGP sessions proposed in the previous chapter, the configuration of the entire iBGP organization can be fully automated. We will call this solution Automatic iBGP, or AiBGP. Another feature of our AiBGP organization is the on-demand provisioning of backup paths to ensure rapid convergence with BGP PIC and facilitate traffic engineering and load balancing.

7.1 Description of the AiBGP organization

We saw in the previous chapters that one source of nexthop diversity in a network is when prefixes are announced on several eBGP sessions with the same neighbor AS. This is a rather predictable source of diversity because an AS usually advertises the same prefixes on all its peering links [FMR04]. In this case, the routers connected to a given eBGP neighbor form together a set of egress points for the destinations advertised by the neighbor. We can use this logical grouping as a starting point for our new organization: A router that needs one or several paths to the destinations advertised by a neighboring AS can choose its egress points in the set of border routers connected to that neighbor. It will then establish iBGP sessions with those egress points.

7.1.1 Terminology and organization

In an AiBGP organization, we call the set of egress routers connected to a common eBGP neighbor the *Contact Group* of this AS. Routers belonging to a Contact Group for a neighbor AS are called the *Contact Nodes* for this neighbor. All routers that are not connected to a given neighbor AS are called **Clients** for that neighbor.

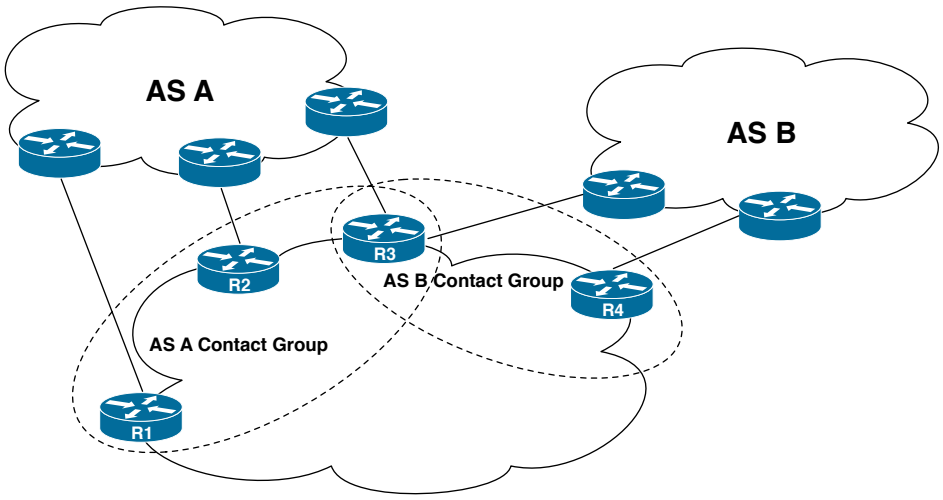


Figure 7.1: AiBGP Contact Group

In the example of figure 7.1, the network has two eBGP neighbors, ASA and ASB. Thus, there will be two Contact Groups, one for each AS. The Contact Group of ASA contains all routers with an eBGP session with that AS, i.e. routers $R1$, $R2$ and $R3$. These routers will be the Contact Nodes for ASA. Similarly, $R3$ and $R4$ are the Contact Nodes for ASB. Contact Groups can intersect: Router $R3$ is a Contact Node for both ASA and ASB.

Routers that are Clients for a given neighbor AS will establish iBGP sessions with one or several Contact Nodes of this AS Contact Group in order to receive the best paths of this/those Contact Node(s). Each client selects at least the router of the Contact Group that is closest in terms of IGP distances as Contact Node. The sessions are only partial iBGP sessions: instead of sending all its paths to a given Client, a Contact Node only sends the paths advertised by the related eBGP neighbor. This corresponds to the liBGP sessions presented in chapter 6. Those liBGP sessions are unidirectional, as paths are sent from the Contact Node to the Client. As in chapter 6, the paths advertised over each eBGP sessions are identified by a BGP community. Upon establishment of the liBGP session, the client uses BGP Outbound Route Filters [CR08] to specify the BGP community corresponding to the neighbor for which it needs the paths.

All routers belonging to a Contact Group must agree on the paths that have to

be propagated inside the AS: if, for example, there exists a path to a prefix with a lower MED than the other, all Clients of the Contact Group must learn this path so that they can select it as best. Therefore, there must exist a Full Mesh of liBGP sessions between the members of a Contact Group to allow them to exchange their best paths. In this particular case, the liBGP sessions are bi-directional, as Contact Nodes exchange their Contact Group paths with each other. Outbound Route Filters must then be used for both Contact Nodes. Best-External advertisement must be activated on those liBGP session, to prevent **Less-Preferred Path Loss** and increase the available diversity.

Using the same example as above, figure 7.2 shows the liBGP sessions established inside the ISP. All routers of ASA Contact Group and of ASB Contact group are fully meshed in their respective group with bidirectional liBGP sessions in order to agree on the set of paths to propagate. Dashed arrows show the Client-Contact Nodes liBGP sessions. First, all routers that do not belong to ASA Contact Group establish such liBGP sessions. In this case, $R4$ and $R5$ need paths to ASA destinations. They will each select the closest contact node for ASA, i.e. respectively $R3$ and $R1$. Similarly, routers $R1$, $R2$ and $R5$ need paths to destinations of ASB. They will thus establish unidirectional liBGP sessions with respectively $R3$ and $R4$.

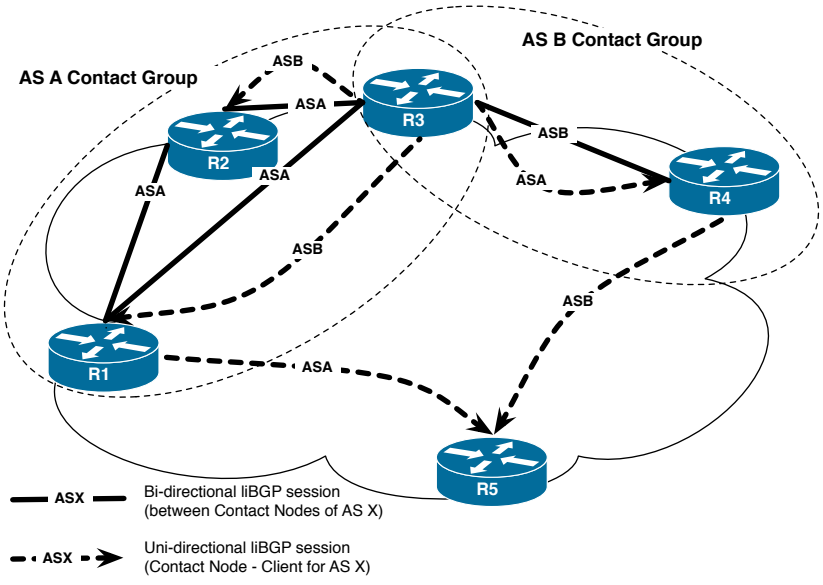


Figure 7.2: AiBGP sessions. Physical links are not shown to keep the figure simple

7.1.2 Automating the AiBGP organization

The specificity of the AiBGP organization is that the sessions are logically built based on the eBGP peerings of the AS. Configuring the sessions does not require any other input from the operators, which means that the configuration task does not need human intervention in theory. This opens the door for the full automation of our AiBGP organization, which will reduce configuration burden for the operators and prevent any iBGP misconfiguration. The automation mechanism presented in chapter 6 is a natural solution for this particular iBGP organization.

Start-up procedure of an iBGP router

All a router needs to establish its liBGP sessions is the list of the eBGP sessions of the AS. This information will be used by a router first to build the liBGP sessions with the other routers connected to the same neighboring AS (Contact Group establishment), then to build the liBGP sessions with Contact Nodes of ASes for which the router does not have direct peering (Client sessions establishment).

For this, we propose to re-use the Contact Information Server of chapter 6. When a router starts up, it needs to configure all the liBGP sessions needed to receive the available paths, and to advertise the paths it learns via its own eBGP sessions. This is done in four steps:

1. Localize the Contact Information Server (CIS): this can be easily done by using IS-IS or OSPF capabilities, as it has already been proposed by Raszuk et al. for iBGP auto-mesh [RAM03].

2. Establish an iBGP session with the CIS: this allows the router to receive all the information about the active eBGP peerings with neighboring ASes.

3. Join its Contact Groups: the first liBGP sessions that the router needs to establish are its Contact Group sessions. It will select, among the eBGP peerings learned during the previous step, all routers that peer with the same ASes as itself, and initiate a liBGP session with each of them in order to participate to the Full Mesh of the Contact Groups. After this step, the router knows all the best paths advertised by the eBGP neighbors that have been selected by its Contact Groups Peers, and can run its decision process to select the ones it prefers.

4. Obtain all other paths: for each AS to which it is not connected, the router chooses as Contact Node the closest router of the Contact Group of the AS in terms of IGP distance. It will then establish a unidirectional liBGP session with each of the selected Contact Nodes in order to receive the best paths learned from the concerned AS. If diversity or robustness is required, the router can choose more than one Contact Node for some or for all Contact Groups. For example, an operator can choose to have diversity for its clients neighbors but not for its shared-cost peers. A solution to implement this is

to attach an additional community to the eBGP peering information related to each eBGP neighbor that is gathered by the CIS. This community will specify the number of sessions required for that neighbor. Customer ASes will have this community value set to 2, while shared-cost peer would have no community, i.e. default value of one session.

Addition of a new eBGP peer

When a new eBGP session is configured on a router, this router checks in the peering information received from the CIS if there is already a Contact Group for the corresponding neighboring AS. In this case, it will establish bi-directional liBGP sessions with all other Contact Nodes of this Contact Group to obtain all the paths for the destinations of the neighboring AS. Then, it sends the information about the new peering to the CIS, which in turns propagates this information to all routers of the AS. Clients of the Contact Group can now decide to establish unidirectional liBGP sessions with the new Contact Node.

If the new eBGP session is the first session with the new neighboring AS, then the router directly sends the peering information to the CIS. All other routers will establish liBGP sessions with him to obtain the destinations of the new neighbor.

Example

In the example of figure 7.2, we detail the establishment of liBGP sessions of router $R2$ upon startup. First, $R2$ retrieves the IP address of the CIS (not shown on the figure), and establishes an iBGP session with it. Once it receives the list of eBGP peerings from the CIS, it looks for routers connected to the same eBGP neighbor as him, i.e. ASA, then establishes bidirectional liBGP sessions with all of them. $R2$ is thus fully meshed with $R1$ and $R3$. Once $R2$ has received all ASA paths from $R1$ and $R3$, it notifies the CIS that it can act as a Contact Node for ASS by sending him a special BGP Update message with its own eBGP peering.

Then, $R2$ looks for the other eBGP peers of the AS, in this case ASB. It selects the closest one $R3$ and establishes an unidirectional liBGP session with it. Upon session establishment, it specifies that it is interested in the paths from ASB.

7.1.3 Providing recovery paths

In the current state of the AiBGP organization, path diversity is still not correctly propagated. Alternate paths are available for the destinations of a given neighbor inside the Adj-RIB-Ins of the Contact Nodes for this AS, but the clients of this AS only receive one of those paths. Thus, except if the destinations are advertised by another neighbor, clients will not be able to perform fast recovery in case of failure of their nexthop. Several options are available to circumvent this weakness. First, a client can simply choose to establish an additional liBGP session with a second Contact Node of the same neighbor to obtain a second path to the destinations

of that neighbor. Second, the solution proposed in chapter 6 can be used, either by locally redirecting the traffic and/or by advertising an alternate nexthop as a community along with the primary nexthop. And finally, Add-Paths [WRC09a] can be used to advertise several paths for a given destination on a liBGP session.

Backup Contact Node

The first alternative to obtain path diversity is to use a second liBGP session with another router of the Contact Group. This simple solution works properly when all paths from the neighbor are equivalent. However, if the paths received from the eBGP neighbor are not of equal quality and one of them is globally preferred by all Contact Nodes, for example because of a higher Local Preference or a lower MED, the Client will receive the same nexthop from its two Contact Nodes.

In figure 7.3, we take the same example as before, but focus only on ASA's Contact Group. Client $R5$ wants to have diversity for the destinations of that AS. It will thus select a primary Contact Node $R1$, and a backup Contact Node $R3$. If Pa, Pb and Pc have the same BGP attributes, $R5$ will learn Pa from $R1$ and Pc from $R3$, and will receive diversity. However, if ASA wants to perform traffic engineering and asks $R3$ to set a lower local-pref on Pc, $R3$ would learn Pa from both Contact Nodes.

A solution to this issue is to configure the second liBGP session of $R5$ as a backup liBGP session. This setting can be specified upon session establishment, along with the specification of the AS for which destinations are required on that session. Instead of advertising its best path, the backup Contact Node will advertise its Best-External path to the client. In figure 7.3, router $R3$ would then advertise its best-external path Pc instead of its best path Pa.

Still, this solution does not guarantee alternate paths in all cases. Indeed, in our example, if $R5$ chooses $R3$ as primary and $R1$ as backup, as the best path of $R1$ is the same is its best external path, $R5$ won't receive an alternate path.

Local recovery

Another possibility is to use the solution presented in chapter 6. Indeed, all routers inside a Contact Group know all alternate nexthops to the common eBGP neighbor. Thus, they are all able to redirect traffic in case of failure of their eBGP link through a tunnel to an alternate nexthop. They can also advertise the alternate nexthop to their client via the specific community $BACKUP\_NEXTHOP$ presented in chapter 6. Thus, in case of failure of the exit point or when the client receives the BGP Withdraw for the primary, traffic can be immediately redirected to the backup nexthop. The advantage of this solution is that no additional session is required, and that memory load for the router is reduced as only one additional community must be stored.

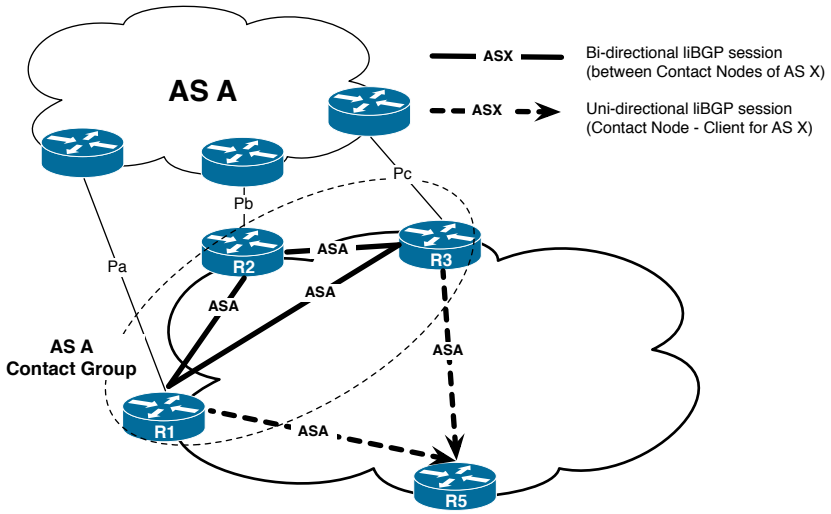


Figure 7.3: Backup contact nodes

Add-Paths

Finally, instead of advertising the alternate path via a community, the Add-Paths encoding can be used to directly advertise two paths for each prefix on Client-Contact Node sessions. Thanks to the Full-Mesh in the Contact Group and the Best-External advertisement, all the Contact Nodes know all the available paths, and each of them is able to provide the primary and the backup path to its Clients. This solution requires a change in the BGP message format, and more memory is needed for the routers to store the additional paths. But its advantage is that the number of paths advertised can be adjusted for example for load balancing purpose.

Depending on the availability of each solution in the routers and on the resources available on them, either the $BACKUP\_NEXTHOP$ community or Add-Paths should be preferred over the establishment of a second Contact Node liBGP session.

7.2 Scalability of the AiBGP organization

The main motivation of our AiBGP organization is the automation of session establishment. Its logical architecture also allows to provide on-demand alternate paths onto routers with an additional mechanism such as the community backup or the Add-Paths feature. But an iBGP organization is also characterized by its cost for the iBGP routers, as well as the routing properties that such organization implies. In this section, we evaluate those characteristics compared to the existing iBGP organizations.

iBGP organization	Max. number of paths in Adj-RIB-Ins
Full-Mesh	$Size_{RT} * (\#Routers - 1)$
Route Reflector	$((\#RRs - 1) + (\#clients)).Size_{RT}$
RR Client	$Size_{RT}.2$
AiBGP	$\sum_{\alpha \in N \mid r \in CG(\alpha)} \mid CG(\alpha) \mid -1).Dest(\alpha)$ $+ \sum_{\alpha \in N \mid r \notin CG(\alpha)} Dest(\alpha)$

Table 7.1: Comparison of iBGP Adj-RIB-In load for different organizations. N is the set of eBGP neighbors, r is the router, $CG(\alpha)$ is the set of routers connected to neighbor AS α (i.e. the Contact Group), $Size_RT$ the number of prefixes in the Internet, and $Dest(\alpha)$ is the set of destinations advertised by α.

7.2.1 Memory load

In chapter 3, we analysed the number of paths that had to be stored in the Adj-RIB-Ins of the routers in the case of a Full-Mesh, and in the case of Route Reflectors. For a Full-Mesh, in the worst case, $N - 1$ times the size of the Internet Routing table is received by an iBGP router, N being the number of BGP routers in the AS. In an organization with redundant Route Reflection, RR clients receive twice the number of prefixes in the Internet. A similar analysis can be performed on the AiBGP organization, as shown in table 7.1.

In this organization, the paths learned by a router come from two types of liBGP sessions. First, it receives the paths from the other Contact Nodes of its Contact Groups. Thus, for each eBGP neighbor, if there are N other Contact Nodes, the router will receive N times the number of paths advertised by this neighbor. Second, the router receives the paths received from the neighbors to which it is not directly connected. Those paths are received on the liBGP sessions with its Contact Nodes. The number of paths received on those sessions is thus the sum of the number of destinations advertised by each non-directly connected AS neighbor.

The size of the routing tables of AiBGP routers depends on several parameters: The size of each Contact Group, the number of destinations advertised by each neighbor AS, and the number of Contact Groups to which each router belongs. It is thus difficult to state whether this organization is more scalable than a Full Mesh or Route Reflection in general.

However, we performed an evaluation of the scalability of iBGP organization in the case of the Tier-1 ISP A using two levels of Route Reflectors already presented in the previous chapters. First, we replay the BGP convergence in the original network with C-BGP based on the paths collected on the top-level Route Reflectors. There are nearly one million of paths to 160,000 destinations. Second, we statically build the AiBGP organization in this network and configure the filters required to obtain the liBGP sessions, and run the same simulation with C-BGP. And third, we compute the Adj-RIB-Ins with a Full-Mesh organization. We then count the number of paths inside each router, and compute the cumulative distribution of the

Adj-RIB-Ins sizes. Results are shown on figure 7.4. The x-axis gives the number of paths in the Adj-RIB-Ins while the Y-axis gives the cumulative percentage of routers. Left-most curves represents organization with smaller Adj-RIB-Ins sizes.

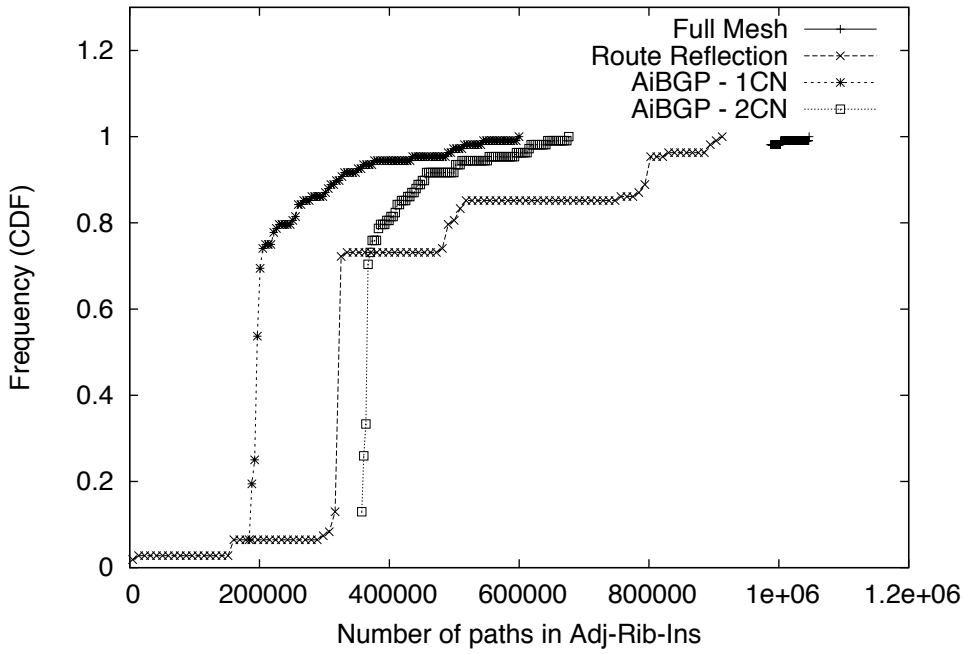


Figure 7.4: Adj-RIB-Ins load of each organization

iBGP organization	Total of Adj-RIB-Ins paths
Full-Mesh	105.72×10^6
Route Reflection	43.7×10^6
One Contact Node	24.67×10^6
Two Contact Nodes	41.85×10^6

Table 7.2: Sum of the numbers of paths stored in ISP A's routers

With a Full-Mesh of iBGP session in this ISP having about 100 routers, all routers have to maintain approximately one million paths in their Adj-RIB-Ins. With Route Reflection, only the top-level Route Reflectors have to maintain that number of paths, 80% of the routers having less than 500,000 paths to maintain. With AiBGP using one contact node, 80% of the routers store fewer than 200,000 paths, while they store fewer than 400,000 paths when using two contact nodes. Notice that the Adj-RIB-Ins load with two contact nodes is similar as the Adj-RIB-Ins load when advertising two paths on a session with one contact node. The more loaded routers have only 600,000 paths with AiBGP compared to 800,000 paths

with Route Reflection. Table 7.2 shows the sum of the number of paths in all Adj-RIB-Ins for each organization. With one contact node, the total memory overhead is 56% of the overhead with Route Reflection, while with two contact nodes, it represents 96% of the Route Reflection overhead. Thus, in this ISP, the AiBGP organization globally reduces the number of paths that have to be maintained in the Adj-RIB-Ins of the routers.

7.2.2 Number of sessions

With Route Reflection, a BGP client only needs to maintain two iBGP sessions with redundant Route Reflectors. Clearly, the AiBGP organization uses more sessions, as, for most routers, at least one session per neighboring AS is needed. In practice, a router needs three types of liBGP sessions: intra-Contact Group sessions, Contact-Node sessions (with its clients) and Client sessions (with its Contact Nodes). Using the topology of the Tier-1 ISP A, we analyzed the number of sessions that are used by the AiBGP organization. In this ISP, there are 91 routers on 108 with eBGP sessions and 217 neighbor ASes. With a Full-Mesh, each router maintains 107 iBGP sessions, and with Route Reflection, there is a total of 363 sessions, which gives an average of about 3 iBGP sessions per router.

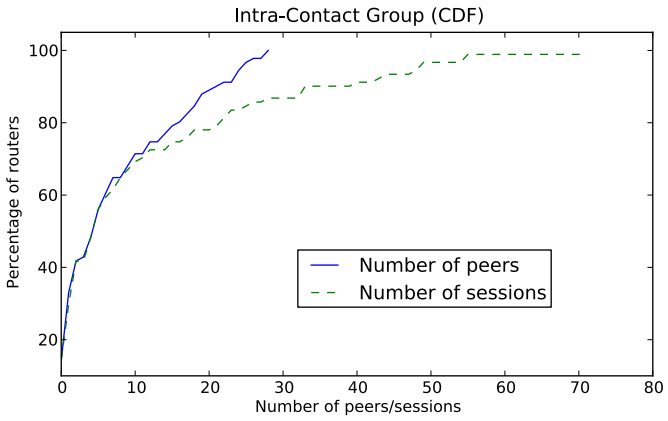


Figure 7.5: Intra-Contact Group sessions

The cumulated distributions of the numbers of liBGP bidirectional intra-Contact Group sessions and peers for each router are shown in figure 7.5. The number of such sessions varies between 0 and 71, while the number of Contact Group peers varies between 0 and 28. The number of bidirectional liBGP sessions depends both on the size of the Contact Group and on the number of Contact Group to which each router belongs. Figure 7.7 shows that 60% of the routers belong to more than two Contact Groups, and 10% belong even to more than 10 Contact Groups and up to 21 Contact Groups. Figure 7.6 shows that 60% of the Contact Groups have a size of 1, such that the corresponding Contact Nodes won't have bidirectional sessions

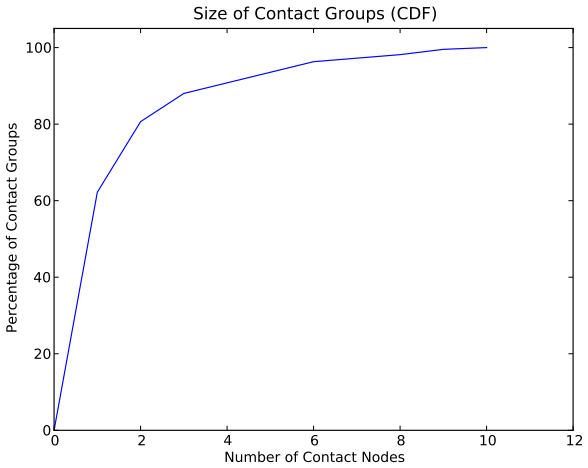


Figure 7.6: Cumulated distribution of Contact Group size

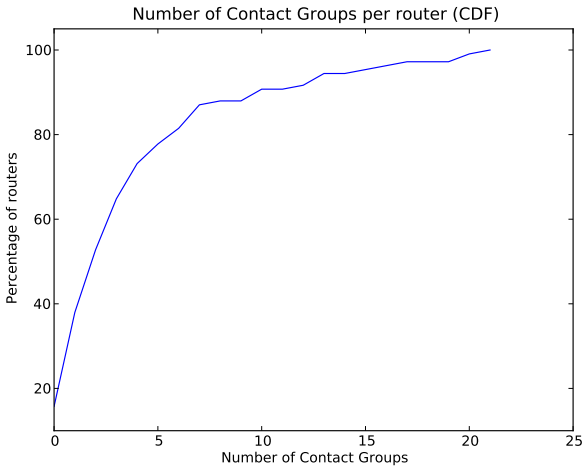


Figure 7.7: Cumulated distribution of number of Contact Groups per router

for them. But the largest Contact Groups can be as large as 10 Contact Nodes, and those will require lots of bidirectional liBGP sessions for their Contact Nodes.

Back to figure 7.5, we notice that half of the routers only maintain less than 5 bidirectional liBGP sessions, with less than 5 peers. Those are the routers belonging to a single Contact Group, such that they have one session with each other Contact Node of this Contact Group. Routers having the largest number of bidirectional liBGP sessions connect to the most connected eBGP neighbors, such that the size of the Contact Group Full-Mesh is larger. For those routers, the number of peers is lower as the number of sessions, which means that there can be several sessions with a same peer. Thus, those routers usually have several Contact Groups in common.

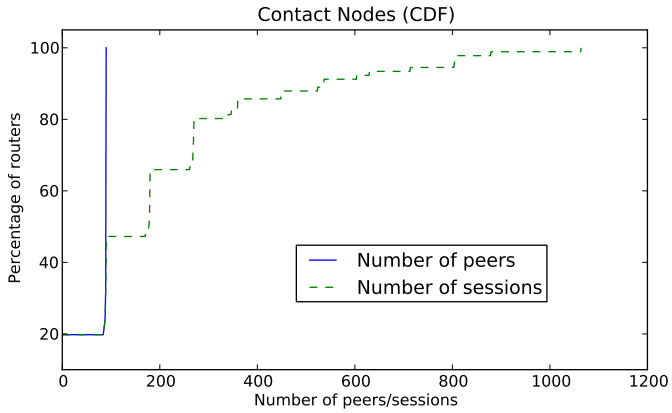


Figure 7.8: Contact Node sessions

Figure 7.8 shows the number of unidirectional liBGP sessions of each router on which it advertises its eBGP paths. A large number of routers have about ninety such peers, which means that most routers choose them as Contact Node for one or several of their Contact Groups. 20% of the routers are not Contact Nodes at all, which means that they are always less preferred than another Contact Node for their Contact Group by potential clients. The number of Contact Node sessions can be as high as 1,000, in the case of a router that is preferred by most routers as Contact Node for several neighboring ASes.

Figure 7.9 presents the number of clients sessions and peers of each router. As a router is typically connected to a few eBGP peers, it will be a client for all other neighboring ASes Contact Group. Here, as we have 217 neighboring ASes, and each router has at least 200 client liBGP sessions. Each router contacts on average 70 different Contact Nodes. Some routers are used as Contact Node for several Contact Groups.

Figure 7.10 shows the total number of liBGP sessions and peers for each router. Clearly, the interconnection of the routers is nearly as dense as for the Full Mesh : Most routers have at least one liBGP sessions with 90 other routers, against 107

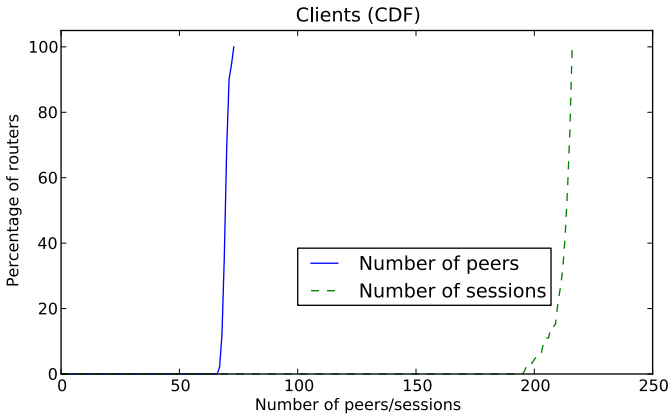


Figure 7.9: Client sessions

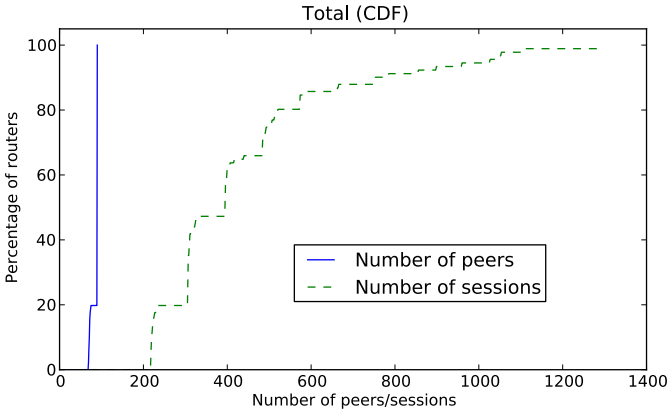


Figure 7.10: All sessions

with the Full-Mesh. The total number of sessions (unidirectional and bidirectional) varies between 200 and 1200. The routers having the largest number of liBGP sessions are those participating to large Contact Groups, or that are Contact Nodes for most other routers for several Contact Groups.

Even though AiBGP could seem even less scalable than the Full-Mesh, those results must be carefully interpreted. Indeed, on each session, only a subset of the paths are exchanged. As shown above, this results in smaller Adj-RIB-Ins in terms of number of paths. Thus, the cost of the liBGP session is mostly a question of maintaining the corresponding number of TCP sessions and computing the BGP messages for each liBGP peer. The first factor depends on router implementations, and we do unfortunately not have information about the scalability of the number of TCP session in routers today. We will see below that the second factor can be reduced by a feature supported by the main router vendors.

Peer groups

We define our liBGP sessions as partial iBGP sessions where the paths from specified ASes are exchanged. If a Contact Node has more than one client for a given AS, it will have to send them the same subset of paths. In that case, it can reduce the amount of computation performed by using the notion of Peer Groups supported by the main router vendors in their BGP implementation [Cis07][Jun06]. A Peer Group is a set of BGP peers that have common characteristics, such as, for example, the same output filters. Using Peer Groups, the paths that have to be propagated to those peers are computed only once by the BGP speaker, and there is only one Adj-RIB-Out maintained for the set of peers belonging to this Peer Group. We measured, on a CISCO 3640, the time required to forward a BGP Update advertising 40 prefixes on 50 iBGP sessions, with and without Peer Groups. Using Peer Groups, the interval between the moment at which the BGP Update was sent and the moment it was received on the last iBGP session was more than 25% smaller than when using normal iBGP sessions. This shows that Peer Groups appreciably enhance the performance of BGP messages processing and forwarding even with a small number of prefixes. This is because the transmission of the BGP Update messages is not as important for scalability as the update of the Adj-RIB-Outs.

Sharing sessions versus sharing peer groups

One question that remains open about liBGP sessions is the following: if two routers have to exchange paths coming from several ASes, do they have to establish a single session, or one session for each AS? This is typically the case when one router is the Contact Node of some Client for several ASes, or when two routers belong together to more than one common Contact Group. Figure 7.7 shows the cumulated distribution of the number of Contact Groups to which routers belong in the Tier-1 ISP A. It shows that 60% of the routers belongs to more than two Contact Groups, so establishing several intra-Contact Group sessions between two routers

should happen frequently. The choice of one solution or the other will influence the number of Peer Groups required, i.e. the number of Adj-RIB-Outs and the amount of memory used on each router. Indeed, in the first case, called *Multiple Sessions* technique, if a router establishes several sessions with a given iBGP router, one session to advertise the paths of each AS, that means that the router sending the paths will have in the worst case one Peer Group for each of its peering AS. In the other case called *Single Session* technique, if a single liBGP session is used for the paths of several ASes, one Peer Group is needed for each different subset of ASes requested by the iBGP peer.

The number of Peer Groups that each router would have in the Tier-1 ISP A using those two techniques is shown in figure 7.11. The maximum number of Peer Groups is 12 for one technique and 11 for the other, while most routers need less than 4 Peer Groups. This is pretty reasonable given the number of routers. For nearly all routers, the *Multiple Sessions* technique needs a few more Peer Groups than the *Single Session* technique. For this particular AS, the *Single Session* technique is better, as it also reduces the number of liBGP sessions by a factor of 10 compared to the *Multiple Sessions* technique.

However, the results in terms of number of Peer Groups depend on the network topology and the location of the eBGP peerings. The choice for one technique or another could be made configurable, for example by using BGP capabilities during session establishment so that a Contact Node can advertise to its potential clients which technique they should use. A Contact Node could even use the "single session" technique with some clients that are many to request the same paths, while using the *multiple sessions* techniques with other clients requesting very different subsets of paths.

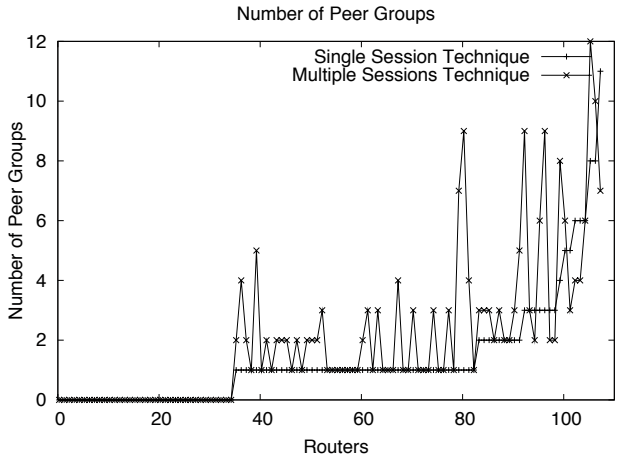


Figure 7.11: Number of Peer Groups per router

Path diversity

With the classical iBGP sessions, the optimal path diversity propagation is obtained when a Full-Mesh of iBGP sessions is used, and Best-External advertisement allow for less-preferred paths to be propagated as backup paths. Route Reflectors greatly reduce this path diversity because Route Reflectors only advertise their best paths, and clients are typically connected to one or two of them.

With AiBGP, the path diversity is under control, depending on the propagation scheme used. It can be propagated only on selected locations for selected customer neighbors. Furthermore, when the goal is to ensure fast recovery, only one backup path can be propagated while with a Full-Mesh using Best-External, all paths are propagated. For example, in the case of a provider neighbor advertising a full routing table on three eBGP sessions, all routers will receive three paths for each prefix with a Full-Mesh compared to two with AiBGP with recovery path propagation. This significantly reduce the load of the Adj-RIB-Ins.

AiBGP controls diversity propagation by providing paths via the same neighbor on-demand, similarly to the solution of chapter 6. Notice however than contrary to the previous solution, Nexthop-AS diversity is also propagated because each client receives the paths from each Contact Group. The exception to this is when a single Contact Node is used for receiving the paths of several Contact Groups.

7.3 Correctness

Route Reflection is known to suffer from routing anomalies, contrary to the Full-Mesh of iBGP sessions. We presented those anomalies in chapter 3, and they mainly consist in Hop-Potato sub-optimality, forwarding deflections and loops and routing oscillations. In this subsection, we analyse our AiBGP organization and show that even though hot-potato cannot be guaranteed, the other routing anomalies can be prevented.

7.3.1 Hot Potato optimality

In AiBGP, a Client chooses for each Contact Group the closest Contact Node in terms of IGP distances. If the Contact Node advertises its eBGP paths, the Contact Node is also the egress router for the traffic to the neighboring AS, and Hot Potato Routing is performed because the Contact Node is the closest egress router to that neighbor. However, if the Contact Node does not select its eBGP path as primary path for a prefix advertised by the neighboring AS, Hot-Potato cannot necessarily be enforced. Indeed, the Contact Node will select another egress router, and that egress router choice is not necessarily the preferred egress router of all its clients for those destinations.

In the example of figure 7.12, three paths are learned from neighboring AS A. The Contact Nodes are configured such that the path received by $R1$ is set a Local-

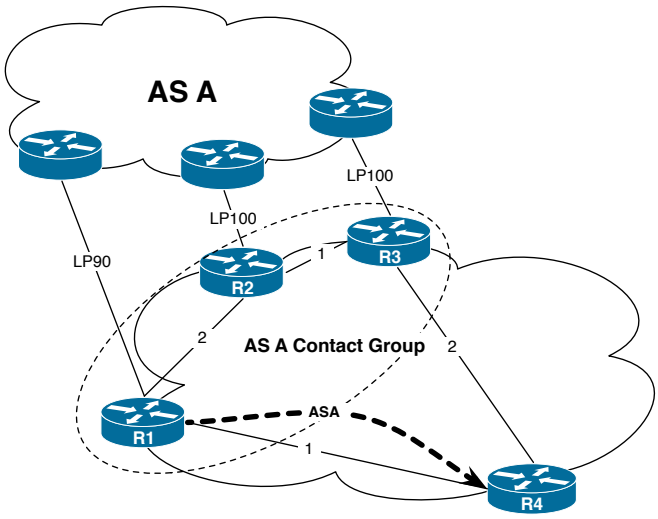


Figure 7.12: Non optimal Hot-Potato Routing with AiBGP

Preference of 90 while the two others have a Local-Preference of 100. Router $R4$ chooses $R1$ as Contact Node for the destinations of ASA, as it is the closest among the three routers. However, as the eBGP paths received by $R1$ have a lower Local-Preference than the other, they will not be advertised by $R1$ to its clients. Instead, it will advertise the paths from the closest egress point between $R2$ and $R3$, i.e. $R2$. But if $R1$ is closest to $R2$ than $R3$, this is not the case for $R4$, which prefers $R3$ over $R2$. Thus, Hot-Potato routing cannot be totally enforced with AiBGP. However, Hot-Potato can be improved when alternate recovery paths are provided with Add-Paths or with a backup Contact Node, as two paths will be made available to the Client.

Deflections

As the propagation path of BGP message is not congruent with the forwarding path, the absence of deflections cannot be ensured in AiBGP. Encapsulation from the ingress interface to the egress interface must be used to prevent any deflection.

MED oscillations

MED oscillations with Route Reflection occur when iBGP paths from a given neighbor have a MED attribute but cannot be compared directly, because their propagation depends on the existence of other paths. In AiBGP, such oscillations cannot occur thanks to the Full-Mesh of liBGP sessions inside each Contact Group on which only eBGP paths of the neighbor are exchanged. All the paths to the same destination received from a given neighbor are thus propagated to all Contact

Nodes. MED values can be compared such that only those with lower MED are advertised outside the Contact Group. Outside the Contact Group, the paths are not propagated further than one hop away, i.e. to the clients. The clients can only compare paths coming from different ASes, and MED values are thus not taken into account outside the Contact Groups.

IGP/BGP oscillations

Oscillations occurring because of the non-congruence of the IGP graph with the iBGP graph cannot occur in AiBGP. The reason is that no path advertisement is dependent on the IGP topology. Indeed, in a Contact Node Full-Mesh, the Best-External eBGP paths are exchanged, and path selection for advertisement only applies between concurrent eBGP paths that are not subject to Hot-Potato. The paths from a Contact Node to a Client are either eBGP paths from the router, or paths preferred over eBGP paths because of lower MED, highest Local-Preference or shortest AS-Path.

7.4 Stability

One concern about the automation of the iBGP organization that we raised in chapter 4 is that the resulting BGP system should be stable. In the case of the AiBGP organization, the establishment of a session between two routers depends on two factors. First, on the location of the eBGP sessions, which is rather stable over time. Contact Groups should then be stable during normal operation. Second, the client-Contact-Node sessions depends on the IGP distances. The stability of the unidirectional liBGP sessions is thus coupled to the stability of the underlying physical topology. In network where this is of concern, it is possible to imagine a variant to the AiBGP organization with a different criteria for choosing a Contact Node. Another possibility is to use timer to limit the propagation of IGP changes on the AiBGP organization. Only durable IGP changes are reflected in the organization to limit the frequency of those changes.

7.5 Conclusion

In this chapter, we proposed a new method to organize iBGP in large ISP networks. For this, we rely on lightweight iBGP sessions. On a liBGP session, a router will only advertise the paths learned from a given neighbor. Compared to existing solutions such as Route Reflectors, this solution offers several advantages. First, it can be completely automated, i.e. iBGP sessions do not need anymore to be manually configured on the routers. This is key to reduce the misconfiguration errors. Second, our organization allows each router to learn two distinct paths. This provides diversity and is key to allow routers to quickly reroute after the failure of one path.

Part III

iBGP 2.0: Add-Paths

Chapter 8

Analysis of Add-Paths Selection modes

In the previous part of the thesis, we explored several solutions to improve iBGP with few modifications to existing routers. In particular, we tried to find alternatives to the advertisement of multiple paths over iBGP sessions. Indeed, even though this is an efficient way to increase diversity, it also forces routers to maintain a lot of additional paths in their Adj-RIB-Ins. Our first proposal prevents BGP Withdraw propagation by notifying the existence of alternate paths to all routers. The second aims at providing a fast recovery mechanism by allowing a router to benefit from novel PIC hierarchical FIB organization. And finally, the third proposal is a new organization allowing iBGP auto-configuration and on-demand fast recovery provisioning.

Still, the *Add-Paths* mechanism allowing the advertisement of multiple paths over iBGP session is undergoing standardization at the IETF [WRC09a] and is being implemented on recent routers operating systems. This solution will thus soon become available to network operators, and as such, deserves attention. In this part of the thesis, we analyse the properties of the *Add-Paths* mechanism.

The standardization activity of *Add-Paths* is focused on the encoding and signaling of such multiple paths. However, the selection of the set of paths to be advertised by routers and Route Reflectors is left to the implementations, depending on the application of *Add-Paths* that they want to support. We call the algorithms to select that set of paths *Add-Paths Selection Modes*.

In this chapter, we explore and analyze several *Add-Paths* selection modes, to provide ground to vendors for making a decision on which modes they want to support, and to operators for which modes they want to deploy in their networks. Our analysis provides insight on what will be the expected impact of these choices. Some performance criteria of these *Add-Paths selection modes* heavily depend on the deployment scenario, i.e., on the connectivity of the AS where it is deployed. Hence, to support our analysis, we will present in the next chapter a tool to perform case-by-case studies for ISPs.

First, we review the main motivations for using *Add-Paths*. They are multiple, and will influence the choice of a given selection mode. Second, we provide an analysis of each *Add-Paths* selection mode. Then, we study the correctness properties of *Add-Paths*, in particular when selection modes are mixed together. Finally, we quickly review the possible deployment of *Add-Paths* in an ISP.

8.1 Motivations for advertising several paths in iBGP

ISPs usually design their networks with resiliency in mind. They tend to multi-home, i.e. connect to multiple providers and peers, and they tend to multi-connect, i.e. to have multiple eBGP links with their neighboring ASes [MdSD+09][FMR04].

Multi-connectivity with the same AS is also often motivated by bandwidth requirements. As a result, multiple in-policy BGP nexthops are often available for each IP subnet in an AS network [UT06].

Nevertheless, as shown in chapter 3, even though multiple paths towards a single IP prefix are available at the borders of an AS, diversity at the router-level tends to be poor [UT06]. This is mainly due to two factors. First, Route Reflectors have normal iBGP sessions with their clients, hence they only advertise one best path to their clients. Furthermore, the two Route Reflectors to which an ASBR connects usually pick the same path, which does not help improving the path diversity on ASBRs. Second, ASBRs having learned external paths do not advertise them to their Route Reflectors when they prefer an iBGP learned path over their external ones. In other words, AS-wide path diversity is usually present for any given prefix at the borders of the AS, but router-level diversity is not ensured in current iBGP designs. In the example of figure 8.1, three paths to destination d are learned by the ISP: Pa, Pb and Pc. As both Route Reflectors prefer the path Pa, path Pb is not advertised to the other ASBRs. Also, due to policies (ex: lower local preference), router $R3$ does not advertise path Pc as it prefers the iBGP-learned path Pa. As a consequence, Pb and Pc are not advertised to other ASBRs, and router $R1$ only learns one path.

Such a lack of router-local diversity can prevent fast recovery when a router or peering link fails. For example, it prevents a fast data-plane activation of alternate nexthops, as provided by the Prefix Independent Convergence feature [Fil07] presented in chapter 6. It also reduces the efficiency of multipath BGP based on BGP nexthop load balancing [Cis]. Furthermore, hidden paths are the source of iBGP routing oscillations caused by MED [BOR+02]. Finally, when a border router fails, an ASBR which does not yet know its post-convergence path must wait for the subsequent iBGP reconvergence. In the meantime, it can trigger the propagation of transient BGP Withdraw messages over its eBGP sessions, leading to transient losses of connectivity [WMW+06][dSFPB09]. Bursts of transient BGP Updates that will eventually be re-updated with the post-convergence paths may also be leaked out over eBGP sessions. This behavior contributes to interdomain routing churn even in cases where the failure could be handled locally by the ISP.

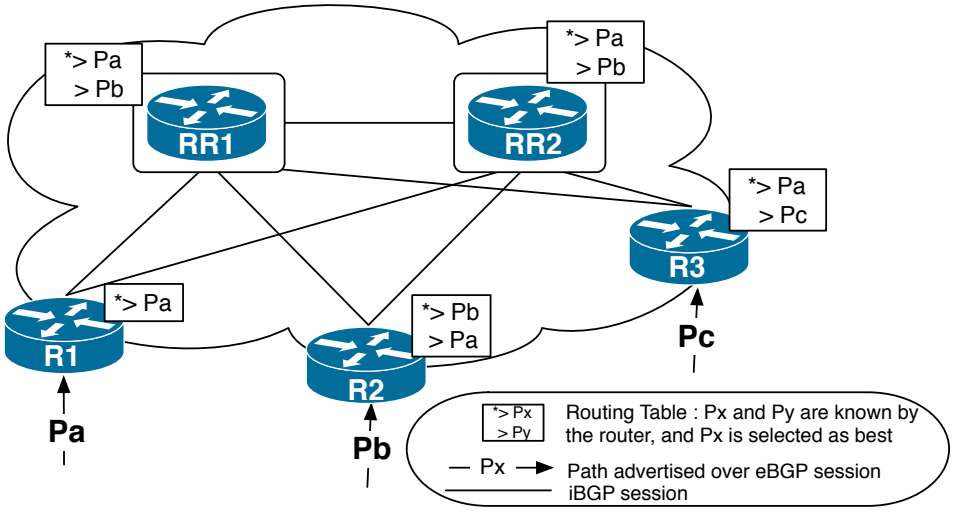


Figure 8.1: ISP using Route Reflection

On Figure 8.1, upon failure of path Pa, router $R1$ cannot reach destination d anymore and will drop packets towards d until the Route Reflectors advertise Pb. Furthermore, $R1$ will also send eBGP Withdraws on its eBGP sessions.

8.2 Properties of Add-Paths selection modes

Depending on how they are selected, the paths advertised by *Add-Paths* have different properties. They will thus unequally meet the different objectives of *Add-Paths* presented in the introduction, and bear different control-plane overheads.

In this section, we will discuss the various evaluation criteria of our analysis of Add-path selection modes. As in previous chapters, we assume that the ISPs use encapsulation (e.g. MPLS) to prevent forwarding loops and path deflection [GW02b]. Most ISPs already use MPLS to support BGP/MPLS VPNs or for traffic engineering purposes.

We discuss in section 8.3 the ability of each mode to provide **nexthop-disjoint alternate paths** to each BGP speaker of the AS, provided that such paths are available at the borders of the network. This property ensures that routers will be able to use multipath BGP [Cis] for load balancing and will be able to use a fast recovery technique such as Prefix Independent Convergence [Fil07] in case of peering links or border routers failures.

We also discuss the ability of each mode to **avoid MED oscillations** [BOR+02]. This was the first motivation for *Add-Paths*. Some of the modes allow routers to learn paths that would have otherwise been hidden to them. This increased diversity can result in a guaranteed iBGP convergence.

Route Reflection is known to sometimes provide sub-optimal routing because Route Reflectors perform an IGP tie-break based on their own IGP distances, which

may differ from the IGP tie-break that the client they serve would perform. By advertising additional paths with *Add-Paths*, **optimal routing** can be re-ensured if the best paths from the perspective of these clients are advertised to them.

By increasing the router-level BGP diversity within an AS, *Add-Paths* reduces the likelihood of propagation of bursts of BGP Withdraw and Update messages outside the AS for a given prefix, which can occur during a BGP convergence following a local link or router failure [WMW+06][dSFPB09]. Indeed, with *Add-Paths*, BGP routers are more likely to already know their post-convergence paths at the time of convergence. We will discuss this property under the term of **eBGP churn reduction**.

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

The Adj-RIB-Ins of BGP routers will contain more paths and thus use more memory than without *Add-Paths*. We will analyze this memory increase under the term **Control-plane load**. Note that the actual memory increase due to the reception of more paths towards the same IP subnet is rendered sub-linear with the number of additional paths thanks to attribute-sharing [ZB03].

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

8.3 Add-paths selection modes

In this section, we review the main *Add-Paths* selection modes that are considered for deployment.

8.3.1 Add-All-Paths

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

This solution gives a perfect path visibility to all routers, thus limiting at best the eBGP churn and transient losses of connectivity in case of nexthop failure, and provides all the paths that a router may consider for actual use with multipath BGP. As no paths are hidden from any BGP router, MED oscillations cannot occur with *Add-All-Paths*.

Also, as no local decision is made by Route Reflectors to not propagate paths to their clients, these have full knowledge of paths and can pick the optimal (hot-potato) one w.r.t. their own IGP distances.

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

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

8.3.2 Add-N-Paths

Add-N-Paths is an intuitive selection mode for *Add-Paths*. It basically provides a configured upper bound N on the number of paths that BGP routers advertise over a single iBGP session.

In this thesis, we consider an implementation where the selection of these N paths is equivalent to the one obtained by a BGP router which first picks its best path, removes all paths with the same nexthop as the best from its Adj-RIB-Ins, then picks its second best on the resulting set of paths, and repeats that process until the resulting set becomes empty, or the N paths have been selected.

Add-N-Paths with $N = 2$ is a very appropriate mode to enable fast recovery with Prefix Independent Convergence [Fil07] as it ensures the availability of at least 2 nexthop-disjoint paths in any BGP router of the AS, provided that there are at least two paths available at the borders of the network. This mode allows for multipath BGP for the same reasons.

From a theoretical point of view, *Add-N-Paths* could be considered as a bad option because it does not provide guarantees in many aspects. First, *Add-N-Paths* does not guarantee that MED oscillations will be avoided when enabled. Under some circumstances, it is even possible that enabling *Add-N-Paths* leaves the iBGP system in a persistent oscillation in the propagation of non-best paths, although iBGP routing was stable without *Add-Paths*, as we will see in section 8.4.1.

Second, routing optimality is not guaranteed but is more likely to be obtained when N is high. Third, even though an ASBR will learn alternate paths towards all prefixes when available, there is no guarantee that it will know the post-convergence path w.r.t. the convergence event. eBGP churn after a local failure may be reduced, but is not necessarily minimized.

Nevertheless, the load and control-plane stress on the routers can be easily

predicted by an ISP, as it is for each router a direct function of the number of iBGP sessions that it maintains, the number of prefixes advertised through the ISP, and the value of N.

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

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

8.3.3 Add-Group-Best-Paths

The main objective of *Add-Group-Best-Paths* [CS04] is to avoid MED oscillations. The idea of this mode is to let BGP routers advertise over iBGP the best path that they know for each neighboring AS. As a result, the lowest-MED paths from each neighboring AS are known to all BGP routers, hence non-lowest MED paths cannot be picked as best, guaranteeing convergence. IGP topology-related oscillations [GW02b] are not prevented by this mode, except if some design constraints on the IGP topology are followed.

This mode provides mitigated benefits for applications other than MED oscillations prevention. It could be deployed as an emergency mechanism to be used when MED oscillations are detected on a prefix, as mentioned in Section 8.5.

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

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

The decision process for *Add-Group-Best-Paths* is relatively simple. The Adj-RIB-In can be optimized by splitting the set of BGP paths according to the neighboring AS from which it was received. The decision process then becomes the usual BGP decision process applied on each of these sets. Upon reception of a BGP Update, a decision is only to be remade on the subset of paths that corresponds to the neighboring AS from which the BGP Update was received.

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

Table 8.1: Summary of selection modes characteristics. The terms "Improved" and "Easier" refer to a comparison with classical iBGP.

8.3.4 Add-AS-Wide-Best-Paths

Another solution focused on the avoidance of MED oscillations has been proposed in [BOR+02]. The solution avoids MED oscillations by design, letting all BGP routers advertise the paths that remain before applying the IGP tie-break rule. Thus, all paths with the highest local preference, shortest AS path length, and lowest MED value per neighboring AS are eligible for propagation. As a result, a router will eventually know all these paths and will no longer select as best a path with a non-lowest MED attribute. This solution also prevents IGP-topology related oscillations without constraints on the IGP topology.

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

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

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

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

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

8.3.5 Add-LP1-LP2-Paths

We propose an additional selection mode that we call *Add-LP1-LP2-Paths*. Its goal is to avoid oscillation and reduce eBGP churn, with a very simple decision process. It also reduces the control-plane churn and load compared to *Add-All-Paths*.

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

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

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

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

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

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

8.3.6 Summary

Table 8.1 summarizes the characteristics of the five selection modes. It shows that both *Add-Group-Best-Paths* and *Add-AS-Wide-Best-paths* are dedicated to MED oscillation prevention and cannot guarantee the existence of at least one alternate path for each prefix. On the contrary, *Add-N-Paths* modes reduce the likelihood but do not prevent MED oscillations. However, they enable fast recovery and limit churn propagation, with bounded costs. *Add-All-Paths* and *Add-LP1-LP2-Paths* both prevent MED oscillations and enable fast recovery, eBGP churn reduction and path diversity for multipath. Compared to *Add-All-Paths*, *Add-LP1-LP2-Paths* has a lower control-plane cost as not all paths are propagated, without losing any of the benefits brought by *Add-All-Paths*.

Whether *Add-N-Paths*, *Add-LP1-LP2-Paths* or *Add-All-Paths* should be preferred when fast recovery and multipath are the target *Add-Paths* applications depends on the network connectivity. It depends on the resources available in the network as well as on its topology and the way it interconnects with other ASes.

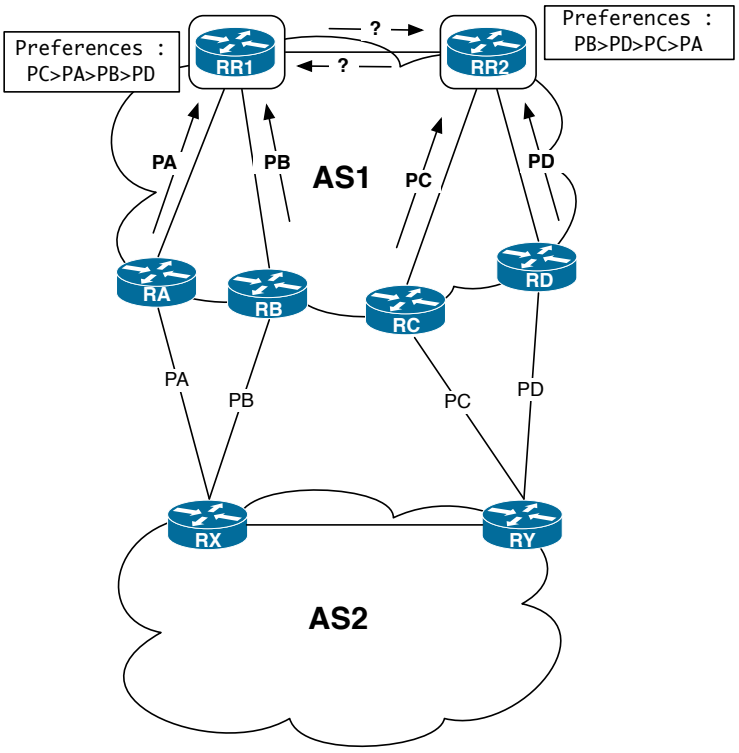


Figure 8.2: Instable BGP system with Add-2-Paths

8.4 Routing anomalies with Add-Paths

Even though one of the goal of Add-Paths is to avoid oscillations, not all selection modes guarantee routing correctness. In this section, we present the correctness issues that can be encountered with some Add-Paths selection modes.

8.4.1 Routing inconsistencies with Add-2-Paths

Similarly to classical BGP, using Add-2-Paths can lead to routing inconsistencies. First, we present several cases of such inconsistencies, then we show that all BGP modes with a fixed number of paths (i.e. Add-N-Paths) can suffer from the same problems when additional paths are learned by the AS.

Topology with multiple solutions

The network described in Figure 8.2 represents two ASes. The edges in the schema represents eBGP or iBGP sessions, depending on whether the routers belong to the same AS or not. $AS1$ has two Route Reflectors, each of them having two clients.

$AS2$ advertises prefix P on four eBGP sessions with $AS1$. We call PA the path to P via RA, PB the path via RB, etc. The IGP links between the routers in $AS1$, not shown in the figure, are such that RR1's preferences on the paths to P are $PC > PA > PB > PD$. Similarly $RR2$'s preferences are $PB > PD > PC > PA$.

If only the best path is advertised, this system converges: $RR1$ chooses and advertises path PA, and $RR2$ path PD. If all paths are advertised, they both know all the available paths, and $RR1$ can choose PC while $RR2$ selects PB.

However, if two paths are advertised (Add-2-Paths selection mode), this topology has two solutions. Depending on which Route Reflector advertises the paths from its client first, two different states can be reached:

- If $RR1$ advertises paths PA and PB before $RR2$, $RR2$ selects PB as its best path and advertises PD as second best. $RR1$ never learns path PC, and keeps PA as its best path.
- If $RR2$ is the first to send its BGP Update, $RR1$ chooses PC as its best path, and $RR2$ never learns PB.

If both Route Reflectors always send their paths to each other together, the system never converges.

Oscillation on the second path

The scenario of figure 8.3 is a modified version of the BAD GADGET case of Griffin et al. [GW02b]. In this figure, $AS1$ has three Route Reflectors, each of them having one client. The IGP links and the corresponding costs, not shown in the figure, are such that each Route Reflector prefers its left neighbor RR path over its own client path, this one being preferred over the path of the right neighbor RR.

Prefix P is advertised by $AS0$ to $AS1$ and $AS2$. Routers of $AS1$ choose the path advertised on the eBGP link with AS0 as their best path, as it has the shortest AS-Path. This topology has thus coherent routing if only one path is advertised. Similarly, if all paths are learned by all Route Reflectors, they are able to choose their left neighbor path as second best path.

However, if only two paths are advertised, there are routing oscillations on the second best path chosen by each RRs. Each of them constantly advertises then withdraws its client path, depending on the right neighbor withdrawing or advertising its own client path. The system being circular, it never converges.

Instable BGP system becoming oscillating with Add-Paths

In Figure 8.4, $AS1$ has three Route Reflectors, each of them having one client. The IGP links and the corresponding costs, not shown in the figure, are such that each Route Reflector prefers its left neighbor path over its right neighbor path, this one being preferred over its own client path.

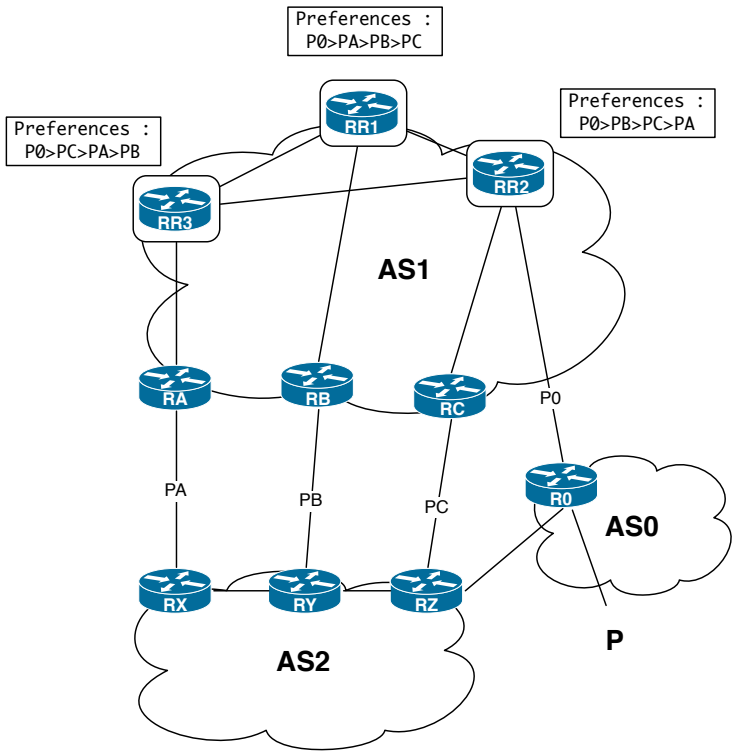


Figure 8.3: Routing oscillation with Add-2-Paths

If only the best path is advertised, the system is already instable: depending on the time on which each Route Reflector advertises the path via its client, it can converge to a common path being chosen by all the RRs. For example, if $RR1$ advertises PA first, $RR2$ and $RR3$ select PA as their best path and don't advertise PB and PC. If they all advertise their client path at the same time, the system diverges: All RRs learn a better path than their own, and they thus simultaneously withdraw then re-advertise their client paths.

If two paths are advertised, it get worse: the system always diverges. Even if the paths advertisements are not synchronized, Route Reflectors are constantly advertising and withdrawing their client path. For example, if PA is advertised first, $RR2$ and $RR3$ choose it as best path, but still advertise respectively PB and PC as their second path. As all Route Reflectors learn two paths better than their own, they withdraw their client path then re-advertise it, and the system oscillates.

This topology is instable even with only the best path advertised, but can still reach a stable state. If the configuration of the Route Reflectors is modified so that they advertise two paths instead of one, routing oscillations appear and the system becomes inconsistent. The only way to have deterministic routing is to advertise all available paths to P.

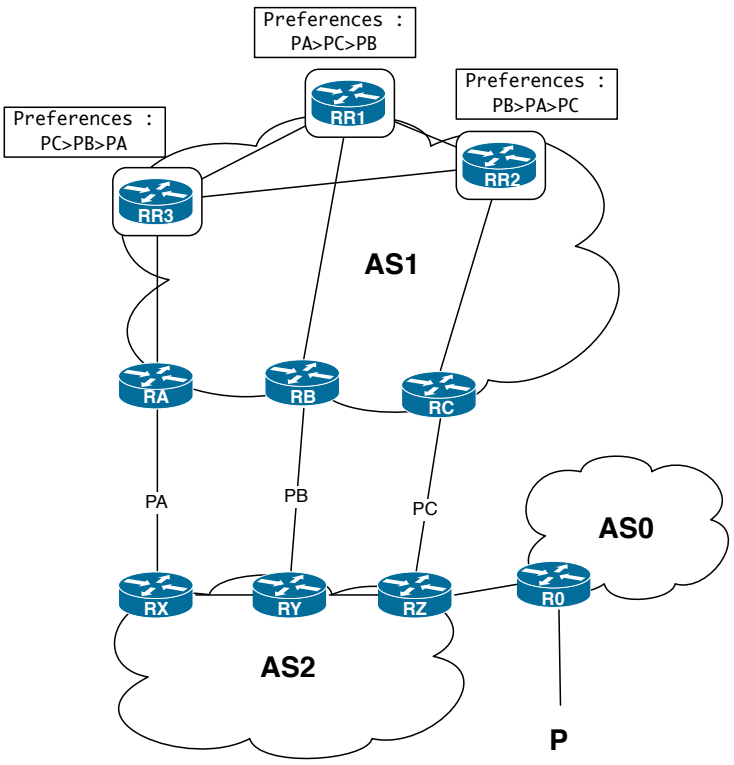


Figure 8.4: Instable BGP system leads to oscillations with Add-2-Paths

8.4.2 Routing oscillations with Add-N-Paths

Even if adding additional paths can solve routing oscillations, other cases of instabilities can occur as long as a fixed number of paths is used.

Consider a routing system with normal BGP that is oscillating between two paths, for example because of MED advertisements. Using Add-2-Paths instead of normal BGP solves the issue, because the two oscillating paths are advertised together, and the system stabilizes. Similarly, in the example of figure 8.3, the system does not oscillate with single path BGP because there is one path that dominates the others (shortest AS-Path length). Advertising one more path leads to routing oscillations on the second path, because the candidates have MED values. The oscillation on the second path can be solved by using Add-3-Paths, such that the oscillating paths are known everywhere.

However, a simple modification to these systems can lead to a new routing oscillation: If a new session is added to a Route Reflector, a new path to the considered prefix could be advertised on that session. If that new path is preferred over all other paths, it replaces one of the paths that was needed to prevent the oscillation and the system loses its stability again.

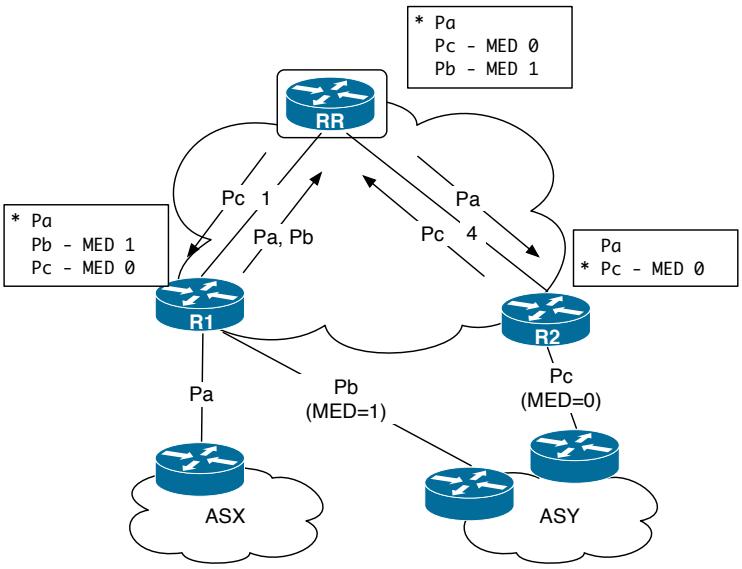


Figure 8.5: MED oscillations solved with Add-2-Paths

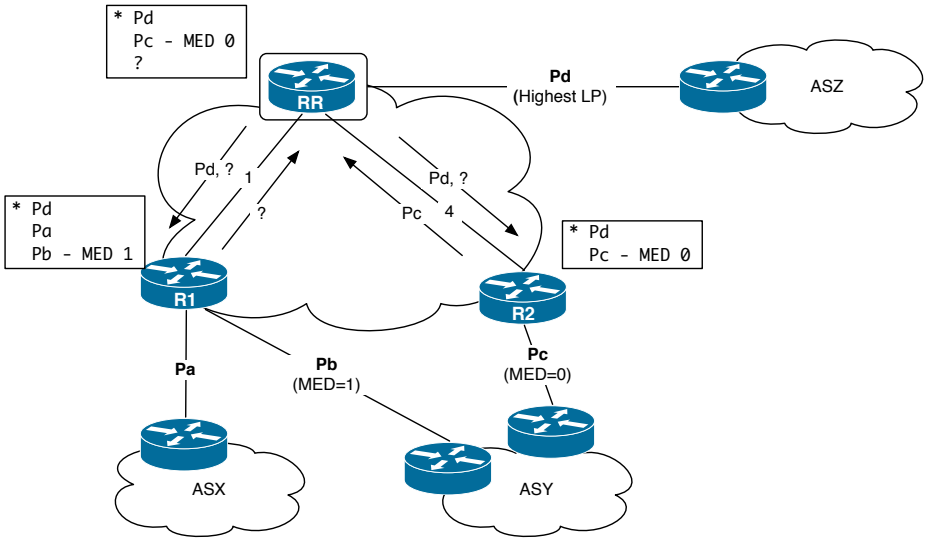


Figure 8.6: MED oscillations with Add-2-Paths

Figure 8.5 is the example of MED oscillations presented in chapter 3. This system was oscillating because the Route Reflector and router $R1$ do not agree on their preferred paths, and constantly exchanged new paths. However, if the routers are configured to advertise two paths instead of one, the oscillation does not occur anymore. As the Route Reflector always advertises the lower-MED path Pc to $R1$, whatever path $R1$ chooses, the system is stable. The stable routing state is shown on figure 8.5.

On figure 8.6, we add an additional session on the Route Reflector, on which a fourth path Pd to the same destination is learned. Due to policies, path Pd has a higher local-preference than the three other paths. Thus, it is preferred by all routers. Now, Pd is the best path of all routers, and the routing advertisement about the second path are similar to the oscillating configuration without *Add-2-Paths*.

Thus, using Add-N-Paths is not sufficient to prevent routing oscillation, whatever the value of N, since it is always possible to find a configuration with additional paths that are likely to hide the lowest-MED paths.

8.4.3 Forwarding correctness

With *Add-Paths*, a router can advertise its non-best paths to its iBGP neighbors. Consequently, those neighbors might select those non-best paths as best for themselves. If plain IP forwarding is used, deflections will appear, and packets will often deviate to another exit point than the one selected by the ingress router. Thus, ingress-to-egress interface encapsulation is needed to ensure that packets follow the intended path.

8.5 Deployment options

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

Different deployment schemes could be imagined. An incremental deployment would probably lead to the following scenario. First, deploy *Add-Paths* on Route Reflectors only while enabling *Best-External* on ASBRs [MFCM10]. Such deployment is sufficient to prevent MED oscillations [CS04]. However, for fast recovery and multipath, increased path diversity is needed at the border routers. Thus, *Add-Paths* must be activated on ASBRs as well to benefit from those features, but not all ASBRs need to be upgraded at the same time, and this can be performed router-by-router. Border routers connected to important customers can be targeted first.

Other options are to deploy *Add-Paths* for a given address family (for example, for Internet paths or for VPNs), or even for specific prefixes matching a given access control list (for example, *Add-Group-Best-Paths* for oscillating prefixes).

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

8.6 Mixing modes together

Operators may also imagine mixing modes together, such as using Add-All-Paths on border routers, and Add-2-Paths on Route Reflectors. This would allow to propagate all alternate paths to the Route Reflectors while limiting the memory load of border routers. However, the properties of mixed modes must be carefully investigated before deployment. Typically, using a mode providing path diversity with a mode preventing MED oscillations will result in a network with neither path diversity nor MED oscillation guaranteed.

In this section, we analyse the consequences of mixing modes on two properties: MED oscillations prevention and diversity availability (at least two nexthops per prefix propagated when available). Even though operators would typically use one mode on RRs and another on ASBRs, we do not make any assumption on the deployment itself, i.e. our reasoning applies on all possible combinations of each pair of modes inside an ISP.

8.6.1 Path diversity availability

Evaluating path diversity guarantees upon mixing modes is straightforward. At least two paths are propagated to all routers provided that they all use a mode with at least two paths. Thus, Add-N-Paths, Add-All-Paths and Add-LP1-LP2-Paths are compatible with each other, because they all ensure the propagation of at least two paths. However, Add-AS-Wide-Best-Paths and Add-Group-Best-Paths do not guarantee that two paths are always available. For example, if a destination is learnt only on the eBGP sessions with a single neighbor, Add-Group-Best-Paths RRs would not redistribute those two paths to their clients, and those clients won't be able to perform fast recovery upon failure of their primary path. A similar situation arises with Add-AS-Wide-Best-Paths when the alternate paths have a lower preference than the primary: the Route Reflectors only propagate the single path with the best local-preference.

8.6.2 MED oscillation prevention

Evaluating the properties of an ISP mixing modes when the objective is MED oscillation prevention is less straightforward. The easiest combinations are when some routers use Add-All-Paths, and the other use Add-Group-Best-Paths, Add-AS-Wide-Best-Paths or Add-LP1-LP2-Paths. The set of paths selected by Add-All-Paths is a superset of all the other modes, such that the MED oscillation pre-

	ADD-N-Paths	**Add-All-Paths**	**Add-Group-Best-Paths**	**Add-AS-Wide-Best-Paths**
Add-N-Paths	-	-	-	-
Add-All-Paths	OK diversity / KO MED osc.	-	-	-
Add-Group-Best-Paths	KO diversity / KO MED osc.	KO diversity / OK MED osc.	-	-
Add-AS-Wide-Best-Paths	KO diversity / KO MED osc.	KO diversity / OK MED osc.	KO diversity / OK MED osc.	-
Add-LP1-LP2-Paths	OK diversity / KO MED osc.	OK diversity / OK MED osc.	KO diversity / OK MED osc.	KO diversity / OK MED osc.

Table 8.2: Summary of the properties of combinations of selection modes (symmetrical values are not shown)

vention property is preserved. Similarly, Add-LP1-LP2-Paths is a superset of Add-AS-Wide-Best-Paths, and the MED oscillation prevention is also preserved when mixing those modes.

The analysis of mixing Add-Group-Best-Paths and Add-AS-Wide-Best-Paths is a little bit more complex. For that, we have to identify a minimum set of paths that are needed to prevent MED-oscillations. The paths responsible for best path oscillations are necessarily among the AS-Wide best paths, i.e. those with the best local-preference and the lower MED. Also, among those AS-Wide best paths, all paths with the lowest MED among the sets of paths received from the same AS must be propagated whatever the best-path selection of the router, in order to invalidate the selection of higher MED paths. Thus, each router must propagate the best path for each neighboring AS using MED among the AS-Wide best paths, plus one best path for the AS not using MED. In [FR09], Flavel et al. prove that this set is indeed sufficient to prevent MED oscillations.

The intersection of Add-Group-Best-Paths and Add-AS-Wide-Best-Paths results in a set of paths containing the best path for each neighboring AS among the AS-Wide-Best-Paths. As this is a superset of the set of paths identified above as necessary to prevent MED oscillations, this combination of modes preserves this property.

Add-N-Paths, on the contrary, is not compatible with the other modes when MED oscillation prevention is targeted. Indeed, a router running Add-2-Path could block the advertisement of the AS-Wide best path with the lowest MED among paths from the same neighbor, because it is only the third best path for that router. Thus, MED oscillation prevention cannot be guaranteed.

Results are summarized in table 8.2: For each mode, we show the compatibility with the others w.r.t. oscillations prevention and path diversity.

8.7 Conclusion

In this chapter, we analysed different alternatives to select the set of paths to be advertised when *Add-Paths* is used. For each of those path selection modes, we present its characteristics in terms of its ability to provide alternate paths and to prevent MED oscillations, along with a discussion of the load that routers have to support when using them. In section 8.4.1, we focus on the correctness properties of *Add-N-Paths*: Even though using this selection mode reduces the occurrences of MED oscillations, it cannot totally prevent them. In particular, MED oscillations can appear upon advertisement of additional paths for a given destination, whatever the number of paths advertised. Finally, we analyse the deployment of Add-Paths in an ISP, and discuss the consequences of mixing different path selection modes in terms of diversity and MED oscillation prevention.

Chapter 9

Analysing Add-Paths deployment

The analysis of chapter 8 highlighted that the cost of deploying Add-Paths is network dependent. The size of Adj-RIB-Ins resulting from the propagation of some set of paths are difficult to predict, because they depend on the characteristics of the paths available at the borders of the Autonomous System. Thus, to complement the previous qualitative analysis as well as to provide a tool to allow operators to evaluate the cost of using Add-Paths in their network, we have implemented Add-Paths in a BGP simulator. Along with this simulator, we provide an analyser that is able to extract a set of metrics from simulation output.

In this chapter, we introduce our Add-Paths Analyser, then we present a set of quantitative analyses on synthetic Internet topologies. The goal of those analyses are multiple. First, we provide a simple analysis of the impact of Add-Paths on the propagation of the paths of a dual-connected stub. This simple scenario allows us to validate the tool, and to isolate the benefits of Add-Paths for customers. The second scenario is more global, and takes more diverse paths into account, leading to a more complete evaluation of the cost of Add-Paths on all prefixes of an ISP. Finally, we also use our tool to simulate the deployment of Add-Paths in the global Internet and evaluate the impact of such deployment on the churn in case of failures.

9.1 The Add-Paths analyser

9.1.1 Evaluation tool

The Add-Paths Analyser is based on the SimBGP simulator, originally developed by the BGPVista team [Qiu]. It is an event-driven simulator written in Python that has been used to evaluate BGP extensions that aims at improving the resiliency of interdomain routing [WG09][WG08]. We choose this simulator as a base for our Add-Paths implementation for several reasons. First, and at the difference of the previously used C-BGP, it is based on an ordered queue, allowing to easily take timing into account in the simulation and thus to model the dynamics of BGP convergence. Second, it already included multipath support, which was a good

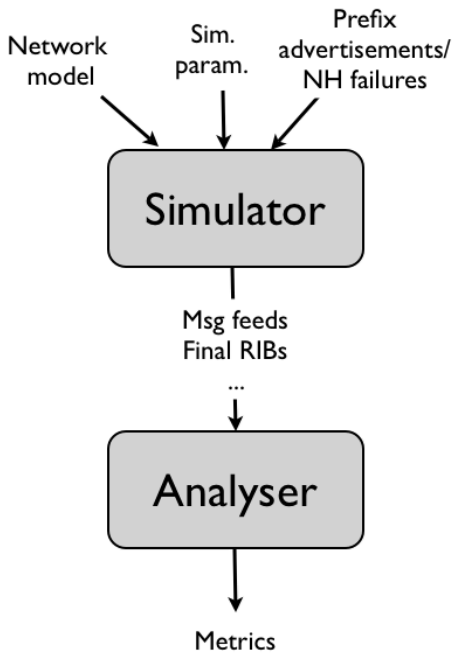


Figure 9.1: Architecture of the Add-Paths Analyser

starting point for an Add-Paths implementation. Finally, the code of the simulator is well structured and easy to modify. In addition to adding Add-Paths support into the simulator, we have also developed a set of test cases to check for simulation consistency upon any implementation change.

In order to extract valuable information from simulations of Add-Paths deployment, we have also developed a set of scripts allowing to analyse the output of the simulator. A complete analysis is performed as shown on figure 9.1: First, the simulator is configured with all network topology information, along with Add-Paths configuration, i.e. which selection mode is to be run on each of the routers, and with the list of events to simulate: Prefix advertisement or withdrawal, or link or session failure. Then the simulation is run, and the BGP messages exchanged between routers are monitored by the analyser. Once the simulation is finished, the Adj-RIB-Ins of the routers are also analysed to quantify their size and the resulting path diversity.

The resulting tool can be used for several purposes. First, it can be used by operators to model and test the deployment of Add-Paths in their network to help them to select the best operational configurations to support this feature depending on their need and resources. To this end, we performed a demo of the tool on the Abilene network, for which data is publicly available, and made the results

available online for interested ISPs[1]. We also used our tool on synthetic topologies, to perform analyses that would otherwise be impossible to carry on due to the lack of reliable operational data. In addition to being available to ISPs, the tool can also be used by the research community.

In this section, we first present the implementation of Add-Paths in the simulator, then we explain the different metrics that are computed by the analyser.

9.1.2 Support of Add-Paths in SimBGP

Modifying the SimBGP simulator mainly implied to follow the execution path of BGP messages from the Adj-RIB-Ins to the Adj-RIB-Outs, and to modify the behavior when the message was learnt or advertised on an iBGP session while letting eBGP messages processing unchanged. During the development process, a set of more important tasks were identified and will be explained in the next paragraphs.

IGP layer

The first modification performed to the simulator does not directly relate to Add-Paths. It consisted in the addition of a more detailed IGP layer to the BGP routing model. In the original simulator, the IGP layer had to be configured as a Full-Mesh of IGP links between BGP routers, and the IGP cost taken into account was simply the cost of the link between a router and its BGP peer. If this model is sufficient for simple convergence cases, or for AS-level simulation, it was clearly limiting for more precise router-level simulation where hot potato routing is playing a key role.

We have implemented an "abstract" IGP layer, which does not model the full IGP protocols but simply relies on IGP graphs shared by all routers of an ISP. Upon failure of a link in the IGP topology of an AS, the shortest paths are recomputed for that AS, and the routers are notified upon changes in the nexthops of their BGP paths. IGP dynamic is not simulated, which results in an IGP convergence of 0 seconds. As BGP convergence is much slower than IGP convergence, this simplification is reasonable.

Path index attribution

In the original SimBGP, a router using multipath BGP can send a BGP Update containing multiple paths for a prefix. If multipath is configured with a limit of N paths, each BGP Update contains the ordered set of the N best paths of the router, each path being identified by its index in the preference list. Thus, upon any change in the ordering of the paths or in any single path of the set, a new BGP Update is generated containing the whole updated ordered set of N best paths.

However, in [WRC09a], it is specified that each path is identified by a Path Identifier, and that paths should be operated similarly to current BGP Updates.

[1] `http://inl.info.ucl.ac.be/add-paths`

When a path for some prefix with a given identifier is received, it implicitly replace the previous path with the same identifier for that prefix. Upon change of a single path for a prefix, the resulting BGP Update should only contain the modified path, and not the other paths of the same prefix. We have modified the update packing algorithm of SimBGP accordingly, such that only the difference between the new set of paths and the set of paths stored in the Adj-RIB-Out of that peer is sent.

Also, path identifier should not have any meaning[MFFR08], so in our implementation, a new paths receives the first unused path identifier for that prefix. In the original version, the path identifier was the rank of the path in the decision process. This modification reduces the number of BGP messages sent to a neighbor if the ranking of the paths is modified while the set of paths remains unchanged.

Support of different selection modes

Once multiple paths advertisement was supported according to the Add-Paths draft, we have modified the process of selecting the paths to be advertised to support the selection modes presented in the previous chapter. For each prefix, each selection mode takes as input the paths available in the Adj-RIB-Ins of the router for that prefix. It then computes the set of paths corresponding to its definition. After that, the resulting set of paths follows the normal processing, i.e. passing through the various outbound filters, before being installed in the Adj-RIB-Out of each peer after computation of the corresponding BGP Update.

Preventing duplicate paths advertisement for loop prevention

When selecting the paths to advertise to a peer, a router must remove from consideration duplicate paths, which are defined as paths identical to a path already in the set except for Route Reflection-related attributes. Indeed, advertising such duplicate paths can result in unstable systems. Considering the example of figure 9.2, using Add-2-Paths without removing duplicates paths results in an unpredictable routing state, which can even oscillate in some cases. In this topology, $RR1$ prefers the paths via $R1.1$ and $RR2$ the paths via $R1.2$. $RR1$ will initially advertise the paths received from its clients. Upon reception of those paths, $RR2$ will choose as its two best paths the paths received from $R1.2$ and the second path advertised by $RR1$. But if $RR2$ sends its initial two best paths before $RR1$, $RR1$ would select the paths via $R1.1$ received from its client and from $RR2$. We thus have two different outcomes, depending on the timing of the Route Reflectors paths advertisements. Furthermore, if the two announcements are synchronized, the system can oscillate between both states. The issue arises because a Route Reflector advertises two best paths that are actually similar regarding all BGP attributes except the iBGP Route-Reflection-related ones (i.e. originator and Cluster-ID list). In this case, the oscillation is prevented if each Route Reflector removes from consideration duplicate paths: $RR2$ would consider only one path via $R1.1$ and one path via $R1.2$

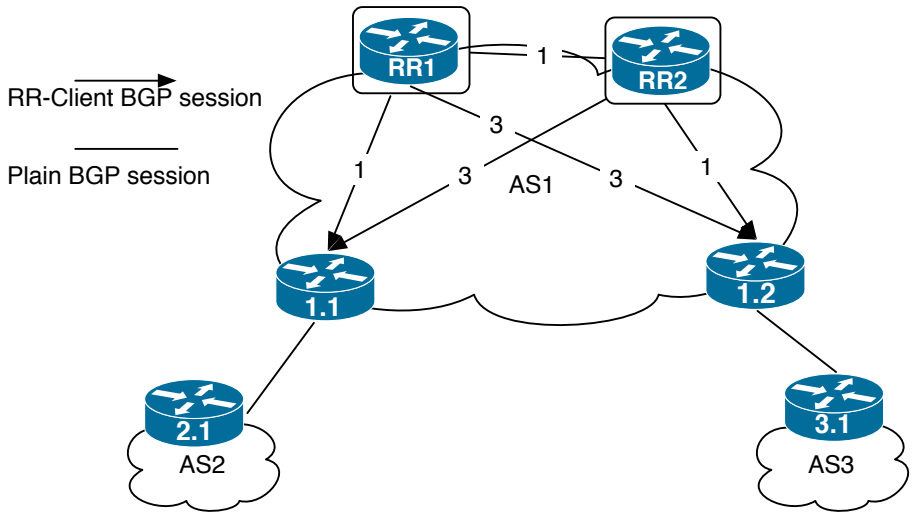


Figure 9.2: Topology with a routing loop because of Route Reflection attributes

9.1.3 Analysis of simulator output

The goal of the analyser is to extract quantitative information about Add-Paths in a given network. For this, we define a set of metrics, and compute them based on the output of the simulator. Those metrics allow to evaluate the load of deploying Add-Paths, but also to compare the different selection modes.

Those metrics are specific to the simulation environment, they should of course not be interpreted as real values. For example, when we talk about convergence time, the corresponding values should not be interpreted as the real convergence times that could be observed in the Internet. Their goal is to serve as comparison basis, for example to evaluate in which cases Add-2-Paths would converge slower than Normal iBGP.

Metrics can be computed with different granularities. They can represent the behavior of the whole Internet, or be computed only on a given relevant part of the network, typically a single ISP.

Convergence time

A first possible characterization of routing behavior in a network is the convergence time upon failure. However, convergence time can be viewed at different levels. First, the **control-plane convergence time** is related to the convergence of the BGP protocol. But even though this reflects the activity of BGP, it is not representative of the reachability of the prefixes. Thus, we also propose the computation of the **dataplane convergence time**, which reflects the availability of a valid path to a destination.

The control-plane convergence time is simply computed as the delay between the first and the last BGP messages of the convergence. For the dataplane convergence time of a router, we analyse the feasibility of the path to the prefix by probing successively the routing table of each nexthop router to the destination. The dataplane convergence time is computed as the time between the event and the time of the last successful probe immediately following a failed probe.

Churn measurement

The metric measuring the **churn** simply counts the number of BGP messages exchanged during a convergence. This metric can be refined depending on the type of the BGP message, as well as on the session on which it was exchanged: Withdraw or Update, and iBGP or eBGP.

When Add-Paths is used, we define a BGP Withdraw as a prefix Withdraw, and not simply as a path Withdraw, in order to keep the meaning of prefix reachability via that router in this metric. Consequently, a BGP Update with Add-Paths denotes a change in the available set of paths advertised, including the reduction of the size of the set of paths.

Propagation of an event

This metric is built on top of the two previous ones. It is computed by counting the number of ASes that are impacted by an event. This metric can be declined in two versions: **reachability impact** and **churn impact**. The first represents the number of ASes that loose reachability during a convergence, while the second reflects the number of ASes that receive BGP messages during a convergence.

Adj-RIB-Ins load

The analyser also analyses the **Adj-RIB-Ins load** of the routers once the convergence is finished. It computes the total number of paths maintained by a router, and the number of paths available for each prefix. This allows to deduce the number of routers having diversity for a prefix, i.e. for which a least two nexthops are available for that prefix.

Routing optimality

With a Full-Mesh of iBGP sessions, all best paths are propagated to all routers. Thus, each router is able to perform its decision process based on its own preference, i.e. the IGP metric to the nexthop. However, with Route Reflection, border routers often only know one or two paths, the ones that were selected as best by their Route Reflectors. Thus, they do not have full path visibility and do not necessarily choose their optimal best path with respect to the IGP distance to the nexthop.

This optimality is important to respect hot-potato routing. When the diversity is increased with Add-Paths, it is expected that the routing optimality is increased, as border routers can choose their best paths among a larger set of candidate paths.

In order to measure **routing optimality**, we compute in the network the sum on all routers of the IGP distances of each best path to the corresponding iBGP nexthop. The lower the value of this metric is, the more optimal path selection is.

9.2 Analyses of Add-Paths deployment

The analyses that we performed with the Add-Paths Analyser are based on synthetic topologies. We start this section with the description of the topology generation process. The next subsections present the different evaluation scenarios along with the results obtained for each of them.

9.2.1 Generation of synthetic topologies

The topologies used for our evaluations must of course be defined with characteristics as close as possible to those of the Internet, while staying computationally reasonable. The topology generation is a two-steps process: First, we generate an AS-level topology with the business relationships between ASes, then we iterate on all ASes to define the internal structure of each of them as well as the way it interconnects at the router-level with its neighbors.

AS-level topology

The tool we choose for the generation of AS-level topologies is Ghitle [Del]. This tool allows to define Autonomous Systems and the relationships between them. Ghitle builds the topology level per level, starting from a dense core and building successively the ASes of each level. ASes in a given level may have peering links with each other, as well as Customer-Provider links with ASes at the upper levels.

For each level, it is possible to specify the number of nodes and the preference. Levels with higher preferences will be preferred by lower level for customer-provider links. In our topologies, all levels have equal preference. Also, Ghitle allows to specify if the core is meshed and if there are peering relationships between stub nodes.

Peering relationships between domains of the same level are computed as follows: Two ASes are Peers with a probability of b. In our case, b is 0.56. The number of Provider/Customer links per AS can be set as constant, as random between 1 and a, or following the power law $N = \frac{a}{x^u}$. We use the latter distribution, with a equal to 5. The choice of a provider is realized using one of four distributions: Uniform, Barabasi (providers with higher connectivity get better chance to be selected), following the level preferences then Barabasi and following the level preferences then Uniform. The distribution used for our Internet topology is using

level preferences then Barabasi. The chosen values are the default values of the Ghitle generator [Del].

The per-level parameters used for each level are summarized in table 9.1

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

Table 9.1: Parameters of synthetic topologies

Router-level topology

Once the AS-level topology has been defined, we have to generate the internal structure of each AS: Number of routers, IGP topology, iBGP topology and eBGP connectivity. The tool used for this step is iGen [QdSFB09]. The number of routers in each AS is chosen in the range of values specified for each level, as shown in table 9.1.

The methodology used by iGen to build our router-level topology is as follows [QdSFB09]: For each AS, it randomly spreads the routers on a map, based on the specified presence (world or continent, in the latter case a continent is randomly chosen). Routers are then grouped in clusters (or PoP), using the k-Medoids (k is the number of clusters) algorithm. Among those PoPs, two nodes are elected as backbone routers and are connected together. Then all other routers of the PoP are connected to the two backbone routers. Finally, the physical topology is completed by connecting the backbone routers of all PoPs together, using a Delaunay Triangulation. The next step is to assign IGP weights to the physical links. IGP weights are generated based on the geographical distance between the routers.

Above the physical topology, we have to build the BGP topology. If the iBGP organization is a Full Mesh, routers are connected to all other routers of the AS with BGP sessions. Otherwise, Route Reflection is used instead. One iBGP cluster is built in each PoP, with the two backbone routers being the Route Reflectors of the cluster [WMS04]. Each router of a cluster is connected with the two Route Reflectors of that cluster, and all Route Reflectors of the domain are connected with a Full Mesh of BGP sessions.

Finally, eBGP sessions between domains have to be established. The number of BGP sessions between two BGP neighbors is proportional to the number of routers of each AS (N_1 and N_2), using this formula [QdSFB09]:

$$max(1 + \lfloor (MaxLinks - 1).\frac{N_1.N_2}{MaxSize^2} \rfloor, 2) \tag{9.1}$$

$MaxSize$ is the number of routers of the greatest domain in the topology, and $MaxLinks$ is set to 10. Thus, the number of interdomain links is between two and

ten. Our topologies are multi-connected, i.e. there are at least two physical links per interdomain relationship. This ensures path diversity at the borders of all ASes, such that the contribution of Add-Paths can be evaluated more precisely.

9.2.2 Evaluation of Add-Paths selection modes

In chapter 8, we presented a set of selection modes along with their respective characteristics. For some of them, those characteristics are highly network dependent. For example, the memory load of a mode such as *Add-LP1-LP2-Paths* depends on the number of paths having the highest preference, which in turns depends on the policies used by the ISP and on the peerings between the ISP and its neighbors. It might be difficult for a provider to correctly evaluate a-priori the costs and benefits of using a given selection mode.

For this evaluation, we built 10 topologies following the methodology presented in the previous section. The resulting SimBGP configuration files are available online [dS].

Scenario 1: Dual-connected stubs

In this scenario, we randomly pick a total of 90 providers among the 10 synthetic Internet topologies. For each of those providers, we select up to 20 pairs of routers and connect them with a dual-connected stub advertising a prefix, as shown in figure 9.3. The provider is configured to use *Add-Paths*, while the other ASes run vanilla BGP. We parametrized the simulator with no MRAI timer and a processing time of BGP messages between 1 and 10 milliseconds. We then use our tool to compute the metrics during prefix advertisement, then upon failure of one link between the ISP and the stub. For each metric, we compute the ratio between the value for each selection mode and the value for vanilla BGP. The modified metric value represents the gain/overhead of *Add-Paths* compared to vanilla BGP. The value of each modified metric for vanilla BGP is thus always one. We then compute the means of each value. The 95th confidence interval of each mean is below 3% of the mean value.

Having a stub advertising a prefix on two links means that the AS knows two equivalent paths (same local preference in the simulation) to that prefix. With vanilla BGP, as Route Reflectors advertise only their best path to their clients and Route Reflectors of the same cluster are likely to select the same path as best, some routers only learn one path. With *Add-N-Paths*, *Add-All-Paths*, *Add-LP1-LP2-Paths*, backup paths are available by design, while with *Add-AS-Wide-Best-Paths*, both paths are known because they have the same local preference. The metric measuring the number of paths confirms this, as the average number of paths is twice higher with *Add-Paths* than with vanilla BGP. *Add-Group-Best-Paths* does not enforce backup path availability because both available paths come from the same AS, and the control-plane load is thus similar to the one with vanilla BGP.

Upon initial advertisement of the prefix, a control-plane stress overhead is

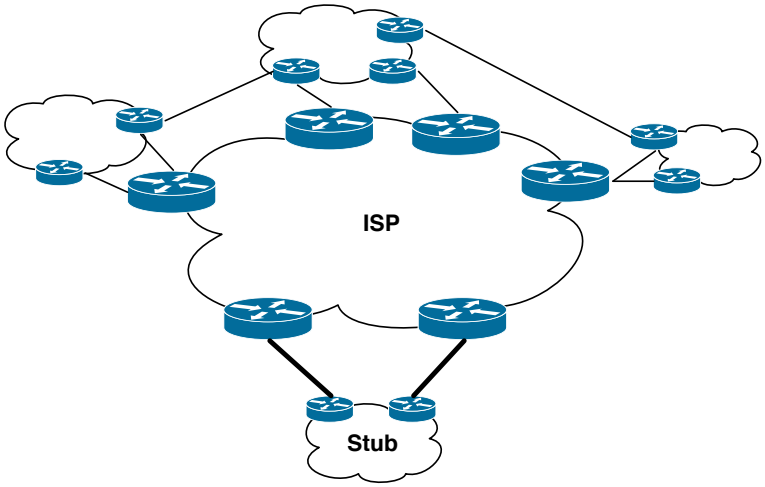


Figure 9.3: Stub dual-connected to its provider

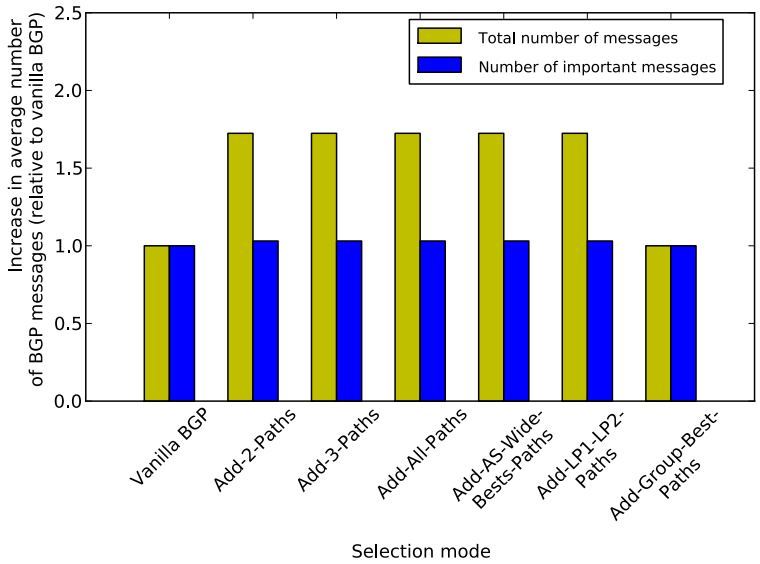


Figure 9.4: Increase in the number of BGP messages exchanged in the provider upon advertisement of a prefix by the dual-connected stub

encountered with the four modes providing backup paths, as more paths are exchanged inside the ISP. We measured in our simulations 70% more BGP messages with those modes than with vanilla BGP and *Add-Group-Best-Paths*, as shown in figure 9.4. However, the number of important messages, i.e. those that change the best path of the router, is roughly the same in all modes. The overhead of using *Add-Paths* upon prefix advertisement is thus mainly due to the exchange of additional paths, and the best path selection is not impacted. This is confirmed by the dataplane convergence time, which is identical in all modes. This suggests that *Add-Paths* does not delay the reachability of a new prefix.

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

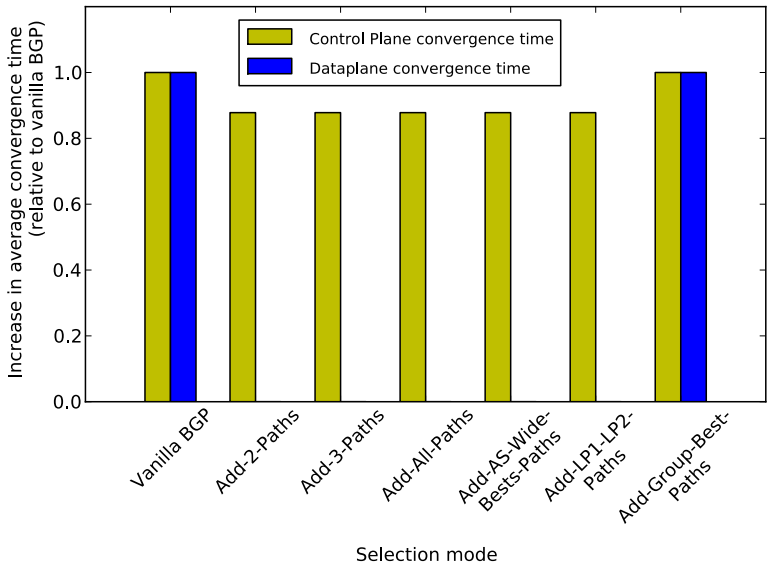


Figure 9.5: Increase in dataplane and control-plane convergence times upon link failure recovery

When a link fails, with vanilla BGP and *Add-Group-Best-Paths*, all routers that use the path via this link and do not know an alternate path encounter dataplane disruptions and send BGP messages outside the AS. Other ASes are then impacted by this local event. With all *Add-Paths* modes except *Add-Group-Best-Paths*, all routers of the provider know both paths. The link failure can be recovered immediately and locally. Figure 9.5 shows that the value of the metric measuring the BGP dataplane convergence time is 0 when backup paths are available. This is also illustrated by the metric measuring the percentage of external ASes impacted : On average, **with vanilla BGP, 12,5% of the ASes in our synthetic Internet learn about the failure, while with *Add-Paths*, no other AS than the provider processes BGP messages about the failure**. *Add-Group-Best-Paths* has the same behaviour as vanilla BGP, as it does not provide an alternate path in this case. This

confirms the adequacy of the *Add-Paths* selection modes advertising at least two paths for providing fast recovery upon link failure.

On the control-plane side, *Add-Paths* converges slightly faster than vanilla BGP when backup paths are available (about 10% in figure 9.5), as only BGP messages about the failure need to be propagated in the AS versus BGP messages about both the failed path and the backup path.

Scenario 2: Prefixes advertised from other providers/peers

Now, we take our 90 providers and for each of them, we advertise one prefix from up to 60 single-connected stubs randomly located in the corresponding Internet topology. The provider under test will thus learn different paths for each of these prefixes depending on its peerings with other ISPs. On the example of figure 9.3, the provider might learn about each prefix from its three neighbors, depending on its policies. If all neighbors advertise the prefix, it will learn up to 6 paths.

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

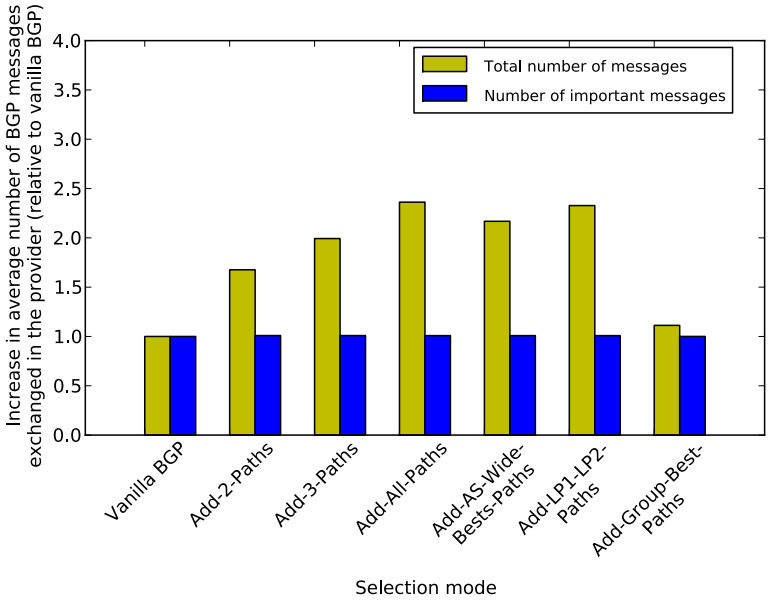


Figure 9.6: Increase in the number of BGP messages exchanged inside a Transit ISP upon initial advertisement of a prefix

Figure 9.6 shows the average number of BGP messages exchanged for a prefix inside Transit ISPs, while figure 9.7 shows the same metrics for T1 ISPS. Similarly to what was observed in the case of the dual-connected stub, advertising the

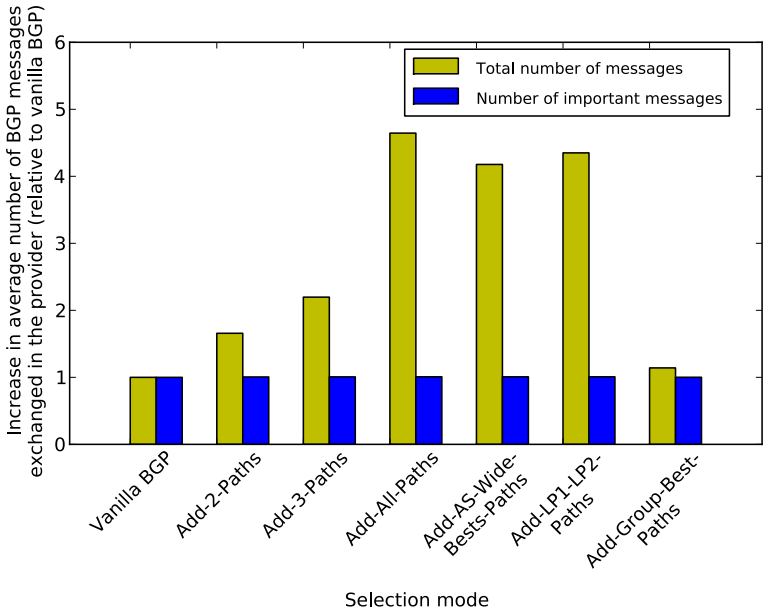


Figure 9.7: Increase in the number of BGP messages exchanged inside a T1 ISP upon initial advertisement of a prefix

additional paths increases the number of BGP messages exchanged inside the AS, but the number of messages impacting the best path selection remains stable. The increase in the number of BGP messages also varies depending on the selection modes. Modes with a bounded number of paths have of course a limited control-plane stress, while the impact of the other modes depends on the topology. For example, routers of a T1 ISP exchange up to 4.5 times more messages with those modes while routers of smaller Transit ISPs exchange 2.3 times more messages than with vanilla BGP. The *Add-Group-Best-Paths* selection mode has the smallest impact among all modes, and is only slightly more costly than vanilla BGP. This is because in our topologies, prefixes are mostly learned on multiple sessions with a single neighbor.

The memory cost of each selection mode is shown on figure 9.8. Roughly, the increase in terms of control-plane load when using *Add-N-Paths* is proportional to the number of paths disseminated, whatever the level to which the ISP belongs. The memory load is bounded by N. However, with *Add-All-Paths*, *Add-AS-Wide-Best-Paths* and *Add-LP1-LP2-Paths*, the number of paths is not bounded, and depends on the number of paths available in the AS. This number of available paths depends itself on the level to which the ISP belongs: Large, highly connected ISPs will have more paths than small providers with a few peering/provider links. In our topologies, routers of T1 ISPs learn on average 9 times more paths than with vanilla BGP, while routers of smaller Transit ISPs learn between 2 and 3 times more paths than with vanilla BGP. We can also notice that on T1 ASes, it is slightly less costly

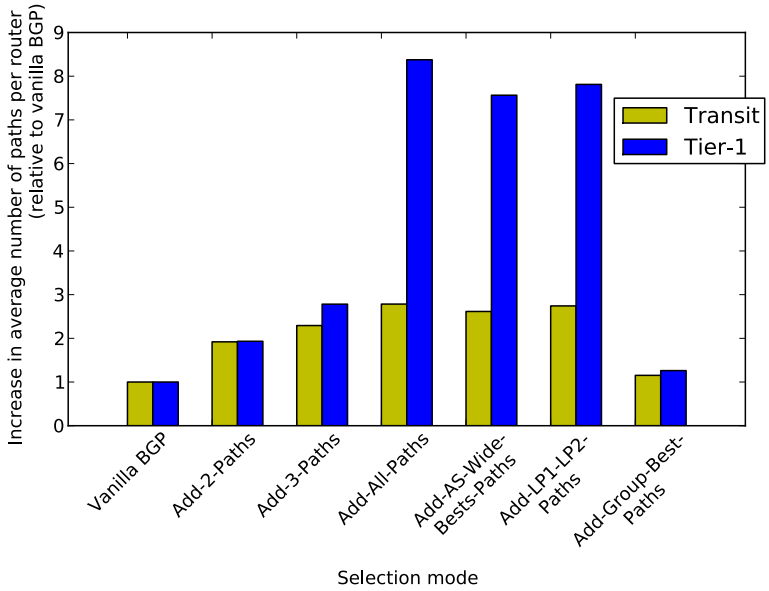


Figure 9.8: Increase in the mean number of paths for a prefix learned by a router

to use *Add-AS-Wide-Best-Paths* or *Add-LP1-LP2-Paths* than *Add-All-Paths*. This is because, among the set of received paths, a few of them have a lower local preference and are thus not advertised by the last two modes. Similarly to what was observed for the control-plane stress, *Add-Group-Best-Paths* has a control-plane load very similar to vanilla BGP. Such a result would encourage an operator only wishing to prevent MED oscillations to use this mode, provided that its IGP topology meets the constraints specified in [CS04]. Otherwise, he should rather use another mode like *Add-AS-Wide-Best-Paths*, at the cost of a higher control-plane stress and load.

9.2.3 Conclusion of the evaluation

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

9.3 Analysis of eBGP churn reduction upon Add-Paths deployment

9.3.1 Motivation and intuition

In chapter 5, we proposed to reduce the propagation of BGP Withdraws upon local failure by propagating the notification of the existence of an alternate path. With Add-Paths, provided that a selection mode enabling path diversity is used, the same objective is automatically fulfilled. But in addition to stopping BGP Withdraws propagation, Add-Paths might also help reducing BGP Update propagation: If the paths are sufficiently propagated across routers, path exploration is reduced. When fewer routing states are explored, there are also fewer BGP messages propagated. In this section, we study the impact of Add-Paths on the churn resulting from an interdomain link failure in the Internet. First, we study the conditions necessary to prevent eBGP Withdraws and eBGP Updates leaking upon failure, and show that they are not applicable in practice. However, we deduce two operational criteria that allow to minimize eBGP churn upon failure. Then, we use our simulator to evaluate the eBGP churn reduction that could be obtained by running Add-Paths on a fraction of the Autonomous Systems in our synthetic Internet topologies.

9.3.2 Preventing BGP messages leaking upon failure

Churn-Minimizing AS

In chapter 3, we defined the Withdraw-Blocking property of an AS as its ability to stop the propagation of BGP Withdraws to its neighbors upon failure. The condition to prevent BGP Withdraw propagation was to provide a valid export-policy compliant path to all routers inside the AS. As some Add-Paths selection modes enable the propagation of several additional paths, the intuition suggests that in addition to preventing the propagation of BGP Withdraws, those alternate paths could as well reduce the propagation of BGP Updates. Similarly to the concept of *Withdraw-Blocking AS*, we define the *Churn-Blocking* property for an AS as follows:

Definition 9.3.1. An AS is said to be **Churn-Blocking** for a destination D if that AS advertises D on at least one eBGP session and does not propagate any BGP message to an eBGP neighbor not advertising D itself, upon failure of its primary path towards that destination.

While the Withdraw-Blocking property aims at stopping dataplane convergence, the Churn-Blocking property aims at stopping control-plane convergence.

In practice, a BGP Update is sent outside an AS during failure recovery when an eBGP router changes its best path, and the attributes of this new best path are different from the previous one from the point of view of the eBGP neighbor. Typ-

ically, a BGP Update is sent when the AS-Path of the path changes[2]. During the whole recovery process, the router can change its mind several times and transiently select different paths, depending on the best paths currently selected by other routers. This is typical BGP path exploration, and may result in sending several unnecessary eBGP messages.

In order to suppress the propagation of BGP Updates upon link failure, a router must first avoid transient states, and thus switch immediately to the path it will prefer after the convergence. We call this alternate path the *post-convergence alternate path*. This path is by definition valid, because if it is used after the convergence, it cannot be impacted by the failure. That path must be the second preferred path. If it is not the case, that means that the second preferred path is invalid, and thus that the router transiently select it as best until it is withdrawn, before switching to the final post-convergence path.

Second, in order to completely prevent the sending of an eBGP message, that post-convergence alternate path should be via the same neighboring AS as the primary one. Thus, when the router switches to the backup path, nothing needs to be sent on eBGP sessions because the new path is similar to the old one from the point of view of the neighboring ASes.

Thus, the following condition is necessary for an AS to be *Churn Blocking*:

Theorem 9.3.1. *An AS is Churn-Blocking for a destination if and only if all routers of the AS know their post-convergence alternate path prior to the failure, this alternate path is their second preferred path, and is via the same neighboring AS as the primary.*

Proof. Upon failure of the primary path to the destination, each router switch to its second best path. As this path is the post-convergence path, the router does not perform any path exploration, and won't change its best path selection after that. As it is via the same AS as the primary, the eBGP path attributes are the same and there is no need to send a BGP Update for this prefix to the eBGP neighbors. Thus, no BGP messages is sent outside the AS by any router of the AS. □

Ensuring the churn-blocking condition for all prefixes in an AS is difficult, for three reasons. First, the ISP must be multi-connected to all neighbors to have an alternate path via the same neighboring AS. If each neighbor advertises its paths consistently, both paths will have the same AS-Path.

Second, the post-convergence alternate path must be known a-priori by all routers. Using an ad-hoc Add-Paths selection mode (i.e. either Add-All-Path or Add-LP1-LP2-Paths) can guarantee that such path is available to all routers, provided it has already been advertised in eBGP.

Finally, the post-convergence alternate path must be via the same neighboring AS as the primary. The latter condition cannot be guaranteed, as illustrated on

[2]MED or community attributes changes could also provoke the sending of eBGP messages, but this will often result from traffic engineering practice. In our analysis, we focus on failover events, which typically imply AS path changes

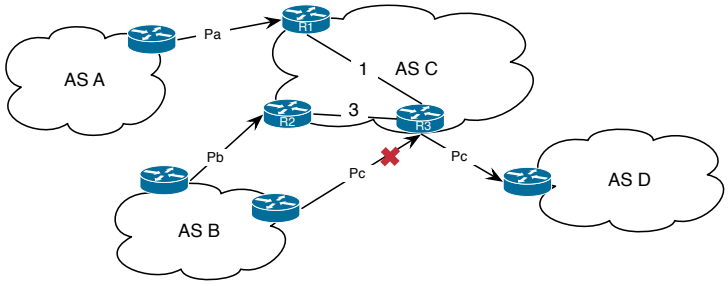


Figure 9.9: Post-convergence path via another AS

figure 9.9. In this network, ASC receives three paths for a destination, one via ASA and two via ASB. Add-Paths is used with a selection mode such that router $R3$ knows the three available paths. Upon failure of its primary path Pc, it will switch on its post-convergence alternate path Pa, which is not via the same neighboring AS as its primary, even if path Pb was available with that characteristic. In this case, the IGP distance to the nexthop lead to a change in preferred exit AS for router $R3$, and that change in the routing state must be notify outside the AS. An eBGP message containing the new AS-Path for that destination will be sent to ASD.

As the churn-blocking property is dependent on the IGP topology and on hot-potato routing, it is difficult to guarantee the prevention of the propagation of any BGP messages even when the failure is local. However, deploying Add-All-Paths and ensuring that an alternate path via the same AS as the primary is available to all routers is the optimal approach to *minimize* the propagation of churn upon failure. We call an AS having those two properties *Churn Minimizing AS*.

Definition 9.3.2. An AS is said to be **Churn-Minimizing** if all its routers use *Add-All-Paths*, and all its eBGP neighbors are multi-connected to him and advertise their prefixes consistently on all their eBGP sessions.

Importance of local failure recovery

Both the *Withdraw-Blocking* property and the *Churn-Minimizing* property require the validity of the alternate path. The path used for failure recovery should of course not be impacted by the failure. We already mentioned in chapter 3 that it is difficult to ensure the validity of a path when the failure is not directly adjacent to the AS. In the example of figure 9.10, ASA advertises two paths to ASB. Due to local-preference settings, ASB only propagates the path Pa to its eBGP neighbors ASC and ASD. Both are multi-connected, and ASD is also dual-homed to ASB and ASC. Consequently, ASD receives four paths, via two different ASes. As it runs Add-All-Paths, all its routers are aware of the four nexthops for destination $10/8$. Thus, ASD is correctly designed to minimize churn propagation upon failure of this destination. However, due to the way ASB propagates its iBGP paths,

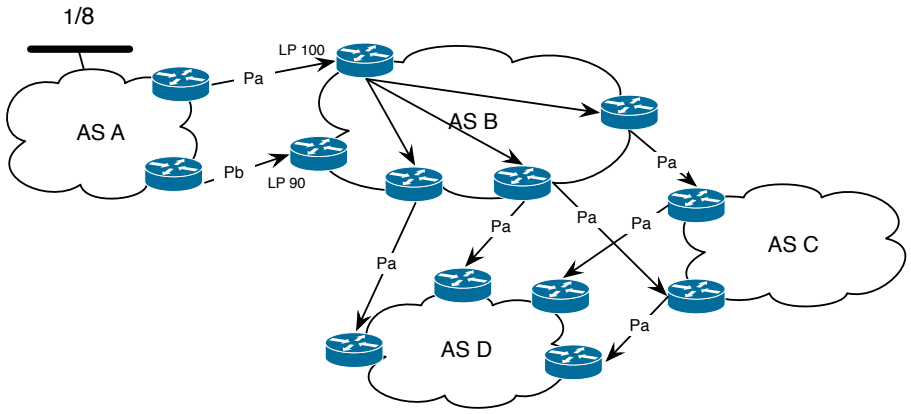


Figure 9.10: Invalid alternate paths

only Pa is known outside this AS, and the alternate paths of ASD are not valid upon failure of the link between ASA and ASB on which Pa is advertised. The same issue is raised for ASC, as well as for all ASes downstream ASC and ASD that also depends on an unique path Pa.

The prevention of churn propagation upon distant failure being difficult to ensure, it is thus of the utmost importance to solve the failure locally, and to prevent the propagation of BGP messages outside the AS(es) directly adjacent to the failure. Indeed, in the example, if ASB propagates Pa AND Pb to all its routers, no BGP messages is sent outside ASB upon failure of the link on which Pa is learned.

The deployment of Add-Paths in the Internet should thus help to reduce global Internet churn upon failure. Indeed, there should have more and more Churn Minimizing ASes as Add-Paths is incrementally deployed. All failures for which both ASes are Churn Minimizing will be solved as locally as possible, while for the others, the overall propagation of eBGP churn could still be reduced in a more limited fashion.

9.3.3 Churn simulation on synthetic Internet topologies

In order to evaluate the impact of Add-Paths on eBGP churn upon failure as well as to evaluate whether *Add-2-Paths* or *Add-All-Paths* has the most positive effect, we re-used our synthetic topologies for a second simulation campaign. Multi-connectivity is ensured for all ISPs due to the way the topology is generated. Due to the computational complexity of those simulations, we limit ourselves to one Internet topology among the ten that were generated. We define five scenarios of Add-Paths deployment: An initial deployment in 5% of the ASes of the synthetic Internet, then in 20%, 50% and finally, a complete deployment in 100% of the ASes. We also evaluate the impact of deploying Add-Paths only in the Tier-

1 ASes. For each scenario, we test both *Add-2-Paths* and *Add-All-Paths*. *Add-2-Paths* should have a large positive impact on BGP Withdraw propagation and reachability upon failure, and should help reducing global churn. We also analyse *Add-All-Paths*, such that all ASes using it are Churn-Minimizing ASes. We will thus explore to which extent this property limits the propagation of BGP messages, depending on the number of ASes that fulfill it, compared to what was obtained with *Add-2-Paths*. We also run a simulation with no Add-Paths deployment to use as reference.

The methodology is as follows: First, we let the 162 lower-Tier ASes in the topology advertise one prefix each. Then, we successively fail 300 links spread over six different levels of the hierarchy: intra-T1, T1-T2, intra-T2 etc. In each level, 50 links are thus randomly selected for failure. We start by analysing the impact of those failures without using Add-Paths, then focus on isolation improvement and churn reduction when using different scenario of Add-Paths deployment.

Analysis of failure propagation without Add-Paths

First, we look at the propagation of the failure in the topology, both from the dataplane and control-plane viewpoints. The metrics for each failure are displayed under the form of a boxplot for each scenario. The line at the center of the box represents the median, while the upper and lower sides of the box represent respectively the upper and lower quartiles of the distribution. Thus, the content of the box represents half of the population. The length of the whiskers are 1.5 times the interquartile range. Other displayed points are the outliers, i.e. points that deviates markedly from other measurements. In this specific case, the outliers are the failures with the largest impact on the churn. Note that each distribution represented contains a population of 300 values, one metric for each link failure.

Figure 9.11 displays the number of ASes encountering control-plane convergence upon occurrence of each failure, i.e. the number of ASes receiving at least one BGP message about the failure. The distribution of the churn impact metric is large with the base scenario (classical BGP), with a lot of outliers. In order to better understand why some failure have such a large impact while other are relatively local, we analysed the topology of the ASes implied in the failures having the largest impact.

On this figure, the failure with the largest impact with normal BGP is a failure implying a stub originating a prefix and one of its providers. In the provider, the iBGP topology is such that one path to the customer is preferred by all routers except one. This is because both paths are learned by routers belonging to the same PoP, and both Route Reflectors of that PoP prefers the same path. This is the worst case of **Route Reflection Path Loss**. Thus, upon failure of the primary link, few routers have an alternate path via the same provider as the primary, and the provider sends BGP messages to most of its neighbors. As the failure is close to the origin of a prefix, lots of ASes in the topology use a path via that provider for that prefix, and they will thus learn about the failure.

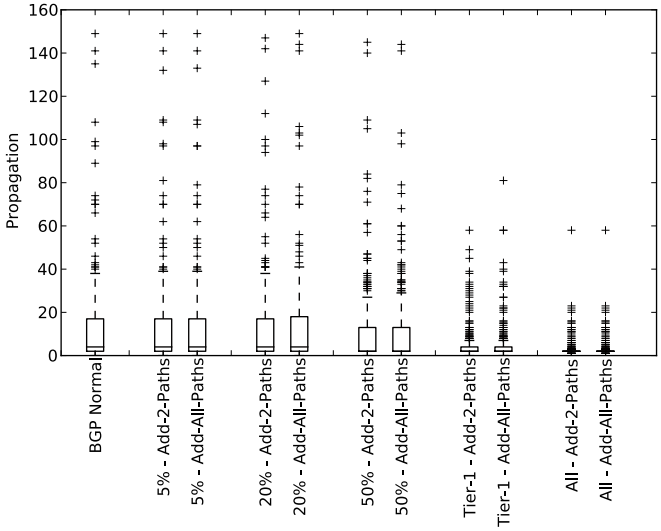


Figure 9.11: Propagation of control-plane convergence

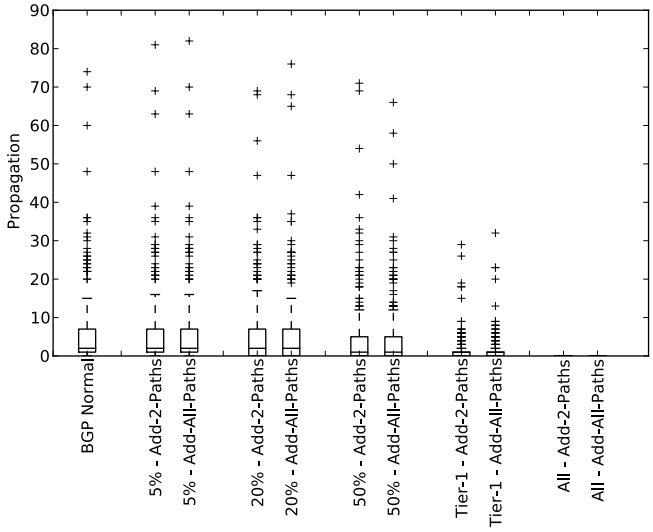


Figure 9.12: Propagation of dataplane convergence

Failure propagation is thus maximal when the failure occurs close to the origin of a prefix, and when one or both ASes adjacent to the failure have a very bad iBGP propagation.

On figure 9.12, we display the number of ASes encountering reachability issues (i.e. loss of path availability for a prefix) upon occurrence of each failure. Clearly, failure propagation on the dataplane is smaller than on the control-plane, because there exists alternate paths with a different AS-Path that prevent dataplane disruption but have no effect on the control-plane propagation.

However, the presence of outliers shows than in some extreme cases, several tens of ASes might encounter reachability issues. Those outliers have the same characteristics than for the control-plane propagation, i.e. bad iBGP diversity and closeness to the prefix origin, but in addition, those failures impact prefixes for which lots of ASes do not have an alternate path with a different AS-Path.

Analysis of churn upon failure without Add-Paths

In addition to failure propagation, we also analyse the number of eBGP messages exchanged during the BGP convergence following a link failure. The results are displayed on figure 9.13. For half of the failures, there are up to 150 eBGP messages exchanged, but the most extreme outlier failure creates up to two thousands eBGP messages.

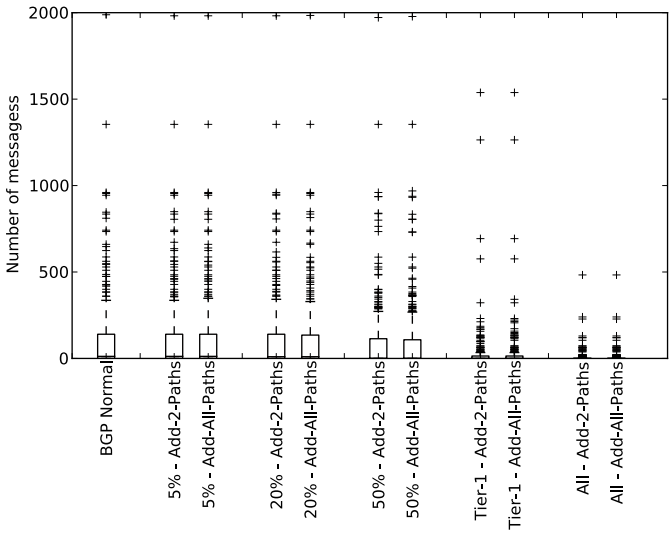


Figure 9.13: Number of eBGP messages during re-convergence

While the propagation of a failure is influenced by the proximity to the failure, the raw eBGP churn depends mainly on the number of prefixes exchanged on the failed link. Indeed, the more prefixes are impacted by the failure, the more BGP

messages will be created by impacted routers. Thus, the most extreme outlier failure of figure 9.13 is not the same as the one of figure 9.11. This time, the failure occurs between a Tier-1 and a Tier-2. The Tier-2 ISP propagates BGP messages about all prefixes learned from its provider to all ASes downstream, while the Tier-1 ISP propagates BGP messages about all prefixes originated downstream its customer.

Thus, the impact in terms of number of eBGP messages depends of course on the iBGP diversity of adjacent ASes, but also on the number of prefixes learned on the failed link.

Reduction of failure propagation with Add-Paths

Based on figure 9.11, we now focus on the metrics computed for the different Add-Paths deployment scenarios. A first observation about using Add-Paths to reduce failure impact is that *Add-All-Paths* does not perform better than *Add-2-Paths*, which means that the latter mode is actually good at propagating the post-convergence alternate path to iBGP routers.

Second, upon full deployment of Add-Paths, the churn propagation upon failure is not totally suppressed. This is not surprising, given that the Churn-Blocking property cannot be enforced totally. The most extreme outlier is a failure implying a router of a Tier-1 ISP and a router of a Tier-2 ISP. The Tier-2 is dual-homed, and upon failure, a large number of its routers switch on a path from the other provider. Thus, ASes downstream the Tier-2 receive Update messages with the new AS path. Also, some routers of the Tier-1 ISP select as alternate path a path from another customer, and propagate Update messages to their eBGP neighbors.

However, on figure 9.11, the range of the distribution of failure propagation for a full deployment is twice shorter than without Add-Paths, and for the majority of the failures, only a few ASes are implied against nearly twenty originally. The improvement was the most significant for the outlier failure with normal BGP analysed in paragraph 9.3.3 : the failure was originally propagated to 159 ASes, but that propagation was reduced to two ASes when using Add-Paths

Thus, failure isolation is obviously improved when Add-Paths is totally deployed in the Internet.

This improved isolation is however not perceptible upon homogenous, incremental deployment of Add-Paths (5% or 20%). Only when half of the ASes have deployed it is there a small improvement. But if Add-Paths is deployed on Tier-1 ISPs, isolation improvement is very similar as the one obtained with a full deployment.

To explain this, we have classified the failure in function of the level to which the ASes adjacent to the link belong, and analysed the distribution. A first result is that a Tier-1 deployment helps logically when the failed link is adjacent to a router of a Tier-1 AS. Of course, when the link is between two Tier-1 ISPs, there is no difference when Add-Paths is deployed in the whole Internet or when it is deployed only in the core, because failure propagation is stopped at the location of the failure

and other ASes do not participate at the event.

Second, failures implying Tier-1 ISPs usually have a larger impact than other failures, i.e. more ASes are affected. Thus, deploying Add-Paths in the core improves the isolation of the most visible failures, which explains why deploying Add-Paths in the core is so efficient on the results of figure 9.11.

The dataplane propagation measurements displayed on figure 9.12 show that the improvement induced by Add-Paths deployment is similar to what was observed for the control-plane propagation, except that with a complete deployment, dataplane convergence is immediate because alternate paths are available to all routers. The benefit in terms of reachability is maximal.

Churn reduction with Add-Paths

Figure 9.13 shows similar trends for the reduction of churn upon Add-Paths deployment scenarios as with the propagation metric earlier: no improvement between *Add-2-Paths* and *Add-All-Paths*, and efficiency of a deployment on all Tier-1 ASes, for the same reasons. But this time, the improvement factor is even larger than for failure isolation. Except for three outliers, all convergences need less than 200 eBGP messages with a full deployment, instead of one thousand and a half without Add-Paths.

We have observed that running *Add-All-Paths* is not more efficient than running *Add-2-Paths* from an eBGP point of view. However, with *Add-All-Paths*, all modifications of any path must be sent to all routers of an ISP, even if that path is not selected as best by anyone. In contrast, with *Add-2-Paths*, only the changes of the two best paths are propagated in the AS. Thus, *Add-All-Paths* is probably more costly in terms of iBGP messages than *Add-2-Paths*. On figure 9.14, we plot the total number of messages exchanged, both eBGP and iBGP. Clearly, *Add-All-Paths* implies much more BGP messages exchanges than *Add-2-Paths*. The total number of messages with a full deployment of *Add-All-Paths* is actually worse than without Add-Paths, while *Add-2-Paths* compensates additional iBGP messages by a reduction of eBGP messages.

9.3.4 Summary of the churn analysis

In this section, we have analysed the properties of an ISP in terms of BGP messages propagation upon link failure. The factors influencing the impact of the failure are of course the iBGP diversity, but also the location of the failure and the number of prefixes exchanged on the failed link.

We have shown that, in theory, to optimize the isolation of a failure in eBGP, *Add-All-Paths* was needed along with parallel links. However, our simulations reveal that this theoretical property might not be needed in practice, and that using *Add-2-Paths* could be sufficient to obtain a good churn reduction. On the contrary, while not improving failure isolation compared to *Add-2-Paths*, *Add-All-Paths* also implies a non negligible increase of iBGP churn.

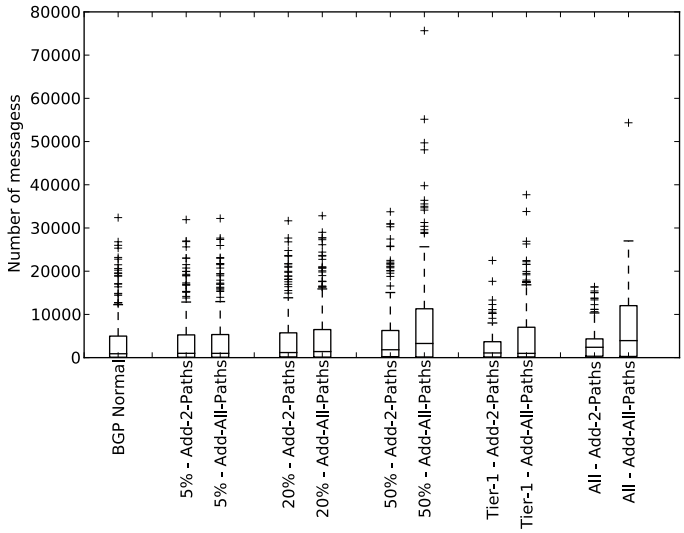


Figure 9.14: Number of BGP messages during re-convergence (eBGP and iBGP)

Simulations also confirm that Add-Paths is sometimes not as efficient as it could be to reduce eBGP churn. Indeed, when the alternate path chosen after the failure is not via the same AS as the failed path, it is necessary to send BGP messages with the new AS-Path.

Finally, our simulations showed that, in contrast to an incremental homogenous deployment of Add-Paths (5%, 20% and 50% scenarios), a core deployment is efficient to sensibly reduce churn upon failure. This is interesting, because the number of Tier-1 ISPs is limited. Such deployment should then result in a global eBGP churn reduction upon link failure.

9.4 Conclusion

This chapter presents the tool that we developed to evaluate the deployment of Add-Paths in ISPs. This tool simulates the BGP convergence of a topology with Add-Paths activated on some routers, and extracts a set of metrics allowing to quantitatively evaluate any Add-Paths deployment scenario.

The tool can be used by an ISP considering the deployment of Add-Paths in its infrastructure, to evaluate the cost/benefits tradeoffs. But it can also be used for large-scale simulation with several ISPs, as shown in the second part of this chapter.

Based on synthetic Internet topologies, we provide two analysis. First, we focus on individual ISPs and evaluate the memory and churn increase due to Add-Paths for customers destinations and for destinations learned from any neighbor. Add-Paths overhead for customer destinations is predictable, but the same over-

head for destinations learned from any neighbor is more difficult to predict, and depends on the size and connectivity of the ISP.

The second analysis focus on the interdomain churn upon failure. Indeed, even though Add-Paths increases iBGP churn, it also improve failure isolation through a better path diversity. The results of this analysis shows that interdomain churn upon failure should be reduced with Add-Paths deployed in the core of the Internet.

Conclusion

The objective of this thesis was to improve iBGP locally to increase the stability of BGP globally. Indeed, our work shows that improving iBGP routing has a positive impact on interdomain routing stability, through a careful provisioning of alternate paths. We have thus proposed to increase the availability of such paths, by adding dedicated iBGP sessions or by advertising multiple paths over iBGP sessions. Alternate paths allow for faster recovery and better isolation upon failure. Improved iBGP path diversity also prevents the occurrence of routing anomalies. Those local improvements have a global impact, since they reduce connectivity losses during failures, allow for shorter convergence times, and reduce the interdomain churn.

With the work performed in this thesis, we have increased the range of solutions for improving internal BGP routing. The panel of available mechanisms is large and varied, and the lack of path diversity is no more a fatality, resulting in better stability and reduced failure impact in the Internet. Among those solutions, Add-Paths is particularly promising, as it constitutes a complete and adaptable mechanism, thanks to the diversity of selection modes available. With an expected deployment of Add-Paths in the next years, there is a real hope that the weaknesses of iBGP will not impair Internet evolution. Thanks to our analysis and our tool dedicated to the evaluation of the deployment of Add-Paths inside an ISP, we are able to help network operators with the adoption of this promising solution.

Detailed contributions

The thesis can be summarized by the following three contributions. The first one is an analysis of iBGP that highlights the importance of alternate paths for routing performance, and shows that current iBGP organizations often prevent the propagation of useful alternate paths. This analysis is described in chapter 3. The second contribution is a set of solutions that improve iBGP without changing the single path advertisement, while the third one explores the new opportunities brought by the advertisement of multiple paths over iBGP sessions. They are presented respectively in part II and part III.

Even though they can all improve iBGP isolation, each of the three solutions proposed in part II has its own characteristics. The `PATH_DIVERSITY` community solution of chapter 5 is incrementally deployable and can be transparently

transported by legacy routers that simply ignore it. It limits the propagation of BGP Withdraws, thus limiting the global recovery time to the local control-plane convergence time. However, this first solution does not improve the local convergence time inside the AS, and packets are still lost as long as the alternate path is not known.

The liBGP sessions proposed in chapter 6 can be deployed on-demand, to protect selected eBGP neighbors against a failure of their interdomain links. These additional sessions provide alternate nexthops that can be installed as backup nexthops in the PIC hierarchical FIB [Fil07]. Combined together, the two mechanisms allow fast dataplane recovery in case of failure. This solution is deployable on top of an existing iBGP organization with a limited overhead.

The main characteristic of the third solution is that it proposes an iBGP organization that is automatically configurable. This limits the risks of humain errors in configuration, errors that are known to contribute to Internet instability [MWA02]. Given the way it is designed, this organization can support failure isolation and fast recovery quite easily and on selected locations, by using Add-Paths or the proposed $BACKUP\_NEXTHOP$ community. The Adj-RIB-Ins load of the routers with this organization is reduced compared to Route Reflection. The overhead of the solution is function of the total number of sessions that have to be maintained. As only a subset of the paths are advertised on each session, and as Peer Groups can improve the scalability of Update packing, this overhead is mostly function of the number of TCP connections that a router is able to maintain.

In table 10.1, we have included our three solutions in the taxonomy of iBGP mechanisms of chapter 4. This table shows that our solutions span a large range of requirements, and increase the set of possible mechanisms that can be used by ISPs to improve their iBGP routing.

In the third part of the thesis, we focus on the Add-Paths mechanism. The main challenge with this solution is the scalability of the Adj-RIB-Ins, as routers must store additional paths. In our analysis, we expose the trade-offs of the different selection modes, and show that network operators must carefully choose their Add-Paths configuration depending both on their infrastructure and on the applications for which they deploy Add-Paths. To help this process, we developed an analyser that is able to model such deployment and guide operators in their operational choices. Finally, we use this tool to simulate a global deployment of Add-Paths, and show that using an Add-Paths mode that improves path diversity improves failure isolation and reduces Internet Churn. The combination of Add-Paths and PIC allows routers to immediately recover from a failure, thus minimizing the dataplane convergence. The provisioning of alternate paths also reduces the path exploration in case of failure, and contributes to a reduction of interdomain churn in the global Internet. We have updated table 10.1 with each Add-Paths selection mode. The strength of Add-Paths is that some selection modes individually cover a wide range of requirements, and are thus really interesting solutions for ISPs that can afford the additional load.

Throughout this document, we have also focused on the correctness properties

Requirements	**iBGP solution**
Scalability	Route Reflectors
Routing optimality and Forwarding Correctness	Full-Mesh, Double Encapsulation, Vutukuru et al. [VVKB06], Buob et al. [BUM08], Musuruni et al. [MC04], Intelligent Route Reflectors [BUQ04], Routing Control Platform [FBR+04][CCF+05], **Add-AS-Wide-Best-Paths** [BOR+02], **Add-All-Paths** [MFFR08], **Add-LP1-LP2-Paths**
Routing Correctness	Full-Mesh, Flavel et al. [FR09], **Add-Group-Best-Paths**[WRC09b], **Add-AS-Wide-Best-Paths** [BOR+02], **Add-All-Paths** [MFFR08], **Add-LP1-LP2-Paths**, **AiBGP organization**
Path Diversity	Best External [MFCM10], Protection tunnel [BFF07], Nexthop-diverse iBGP topology [PQU+10], **Add-All-Paths** and **Add-N-Paths**[MFFR08], **Add-LP1-LP2-Paths**, **Additional liBGP sessions**, **AiBGP with fast recovery support**
Failure Isolation and reduction of Internet churn	Nexthop-diverse iBGP topology [PQU+10], **Add-All-Paths** and **Add-N-Paths**[MFFR08], **Add-LP1-LP2-Paths**, Protection tunnel [BFF07], **`PATH_DIVERSITY` community**, **Additional liBGP sessions**, **AiBGP with fast recovery support**,
Automatic Configuration	iBGP auto-mesh [RAM03], Auto-discovery of protection tunnel[BFF07], **Additional liBGP sessions**, **AiBGP organization**
Robustness	Redundant Route Reflectors, Reliable RR topology [XWN03] [BUM08], **AiBGP with two Contact Nodes**
Simplicity	Full-Mesh, **Add-All-Paths** [MFFR08]
Incremental Deployment	Best External, Protection tunnel [BFF07], Nexthop-diverse iBGP topology [PQU+10], **`PATH_DIVERSITY` community**, **Additional liBGP sessions**,

Table 10.1: Taxonomy of iBGP solutions, including Add-Paths and our proposals

of iBGP with each studied mechanism. Ensuring the forwarding correctness can be easily performed thanks to encapsulation from the ingress router to the egress interface, but routing correctness is more difficult to guarantee. However, several solutions are able to prevent routing oscillations. The AiBGP organization is built in such a way that oscillations cannot occur, and several Add-Paths selection modes can also solve these issues. Oscillations prevention also contributes to the global Internet stability.

Perspectives and further work

In this thesis, we have proposed and studied several mechanisms for improving iBGP behavior. Our analysis of those mechanisms was based on available routing data and simulations. Obtaining data for research purpose is difficult, because operators are reluctant to give information about their network. Collecting routing data is also challenging. Route Views data provide an interesting insight on eBGP paths, but obtaining information about iBGP routing requires to monitor iBGP routers internally, or to collect iBGP paths directly at some or all iBGP routers. This last method was the one used by ISPs to collect the data we obtained, but the set of collected paths is not complete, as only the Route Reflectors were probed. Thus, the analyses based on these data probably underestimate the available diversity, as paths with lower preferences were not collected because they were not advertised to Route Reflectors. We used simulations on synthetic topologies when this lack of paths was expected to have an impact on our analysis results. This was the case when analysing the behavior of Add-Paths selection modes. As a result, it would be interesting to complement our analysis by looking at complete routing data from real ISPs. For the ISP itself, such analysis would allow to identify the causes of diversity losses, and the destinations impacted by the lack of alternate paths. This would help operators to choose the proper iBGP mechanism to improve the service to their customers.

The analysis of the memory load required to store the paths in the Adj-RIB-Ins could also be refined by taking attribute sharing into account. Such data structure optimization allows router implementations to minimize the actual memory footprint of the paths by sharing common attributes such as the AS-Path [ZB03]. In particular, this would give a more accurate view of the scalability of Add-Paths selection modes. However, this analysis requires input from router vendors about the details of the data structures they use.

The interest of IETF participants on Add-Paths allows us to expect deployment of this solution by ISPs in the near future. Such deployment will certainly raise new research challenges, and leave room for several improvements. We have already mentioned in this thesis the possibility to mix different selection modes together inside a given ISP. This analysis can be deepened by looking at more specific scenarios, as for example the combination of *Add-All-Paths* on border routers and *Add-2-Paths* on Route Reflectors.

There are also optimization challenges with Add-Paths. First, the implementation of routers can be improved to support a given selection mode. The data structure of the Adj-RIB-Ins could be organized to pre-classify the paths for the decision process. For example, if *Add-LP1-LP2-Paths* is used, having the paths pre-ordered based on their local-preference would help the selection of the paths with the best local preference. Second, the advertisement of several paths per prefix will have an impact on the way BGP Updates are built. Indeed, similarly to Attribute Sharing in routers data structures, several prefixes sharing the same BGP attributes can be packed in a single BGP Update message. This packing could be optimized to take the specificities of Add-Paths encoding into account.

Finally, our Add-Paths Analyser still deserves attention, and could be upgraded to support potential operator demands. First, its usability could be increased to ease the construction of a network model, for example through the support of vendors configuration languages. Second, additional features could be added, as for example the support of Virtual Private Networks. VPN services are indeed candidates to benefit from Add-Paths deployment, as fast recovery is a critical requirement in this context.

Bibliography

[ACK03] S. Agarwal, Chen-Nee Chuah, and R.H. Katz. OPCA: Robust Interdomain Policy Routing and Traffic Control. In *IEEE Conference on Open Architectures and Network Programming, 2003*, pages 55 – 64, 2003.

[ALD+05] J. Abley, K. Lindqvist, E. Davies, B. Black, and V. Gill. IPv4 Multihoming Practices and Limitations. RFC 4116 (Informational), July 2005.

[AS05] C. Appanna and J. Scudder. Multisession BGP. Internet draft, draft-ietf-idr-bgp-multisession-01.txt, work in progress, October 2005.

[BBAS03] A. Bremler-Barr, Y. Afek, and S. Schwarz. Improved BGP Convergence via Ghost Flushing. In *INFOCOM 2003*, volume 2, pages 927–937 vol.2, 2003.

[BBGR01] S. Bellovin, R. Bush, T. Griffin, and J. Rexford. Slowing routing table growth by filtering based on address allocation policies. June 2001. http://www.research.att.com/jrex/.

[BCC00] T. Bates, R. Chandra, and E. Chen. BGP Route Reflection - An Alternative to Full Mesh IBGP. RFC 2796 (Proposed Standard), April 2000. Obsoleted by RFC 4456.

[BFCW09] Hitesh Ballani, Paul Francis, Tuan Cao, and Jia Wang. Making routers last longer with ViAggre. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 453–466, Berkeley, CA, USA, 2009. USENIX Association.

[BFF05] O. Bonaventure, C. Filsfils, and P. Francois. Achieving sub-50 milliseconds recovery upon BGP peering link failures. In *Co-Next 2005*, Toulouse, France, October 2005.

[BFF07] O. Bonaventure, C. Filsfils, and P. Francois. Achieving sub-50 milliseconds recovery upon BGP peering link failures. *IEEE/ACM Trans. Netw.*, 15(5):1123–1135, 2007.

[BGT04] Tian Bu, Lixin Gao, and Don Towsley. On characterizing BGP routing table growth. *Comput. Netw.*, 45(1):45–54, 2004.

[Bha03] M. Bhatia. Advertising Equal Cost Multi-Path (ECMP) routes in BGP. Internet draft, draft-bhatia-ecmp-routes-in-bgp-00.txt, work in progress, May 2003.

[BOR+02] A. Basu, Chih-Hao Luke Ong, A. Rasala, F. B. Shepherd, and G. Wilfong. Route oscillations in I-BGP with Route Reflection. In *SIGCOMM '02*. ACM, 2002.

[BQ03] O. Bonaventure and B. Quoitin. Common utilizations of the BGP community attribute, June 2003. Work in progress, draft-bq-bgp-communities-00.txt.

[BUM08] Marc-Olivier Buob, Steve Uhlig, and Mickael Meulle. Designing optimal iBGP route-reflection topologies. In *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, volume 4982 of *Lecture Notes in Computer Science*, pages 542–553. Springer Berlin / Heidelberg, 2008.

[BUQ04] O. Bonaventure, S. Uhlig, and B. Quoitin. The case for more versatile BGP Route Reflectors, July 2004. Work in progress, draft-bonaventure-bgp-route-reflectors-00.txt.

[CCF+05] Matthew Caesar, Donald Caldwell, Nick Feamster, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. Design and Implementation of a Routing Control Platform. In *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, May 2005.

[CDZK05] J. Chandrashekar, Z. Duan, Z.-L. Zhang, and J. Krasky. Limiting Path Exploration in BGP. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2337–2348 vol. 4, March 2005.

[CGG+04] Don Caldwell, Anna Gilbert, Joel Gottlieb, Albert Greenberg, Gisli Hjalmtysson, and Jennifer Rexford. The cutting EDGE of IP router configuration. *SIGCOMM Comput. Commun. Rev.*, 34(1):21–26, 2004.

[Cis] BGP Multipath. Cisco online documention.

[Cis07] Cisco Systems. IOS 12.4. `http://www.cisco.com`, May 2007.

[Con10] Internet Systems Consortium. Interdomain Survey Host Count, July 2010.

[CR08] E. Chen and Y. Rekhter. Outbound Route Filtering Capability for BGP-4. Internet draft, draft-ietf-idr-route-filter-17.txt, work in progress, June 2008.

[CS04] Enke Chen and Naiming Shen. Advertisement of the Group Best Paths in BGP. Internet draft draft-chen-bgp-group-path-update-02, September 2004.

[CTL96] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. RFC 1997 (Proposed Standard), August 1996.

[cym] Team Cymru bogon route server project.

[DB08] Benoit Donnet and Olivier Bonaventure. On BGP Communities. *ACM SIGCOMM Computer Communication Review*, 38(2):55–59, April 2008.

[Del] Cedric Delaunois. Ghitle : Generator of Hierarchical Internet Topologies using LEvels.

[dS] Virginie Van den Schrieck. Topologies used for simulations.

[dSFPB09] Virginie Van den Schrieck, Pierre Francois, Cristel Pelsser, and Olivier Bonaventure. Preventing the Unnecessary Propagation of BGP Withdraws. In *Proceedings of IFIP Networking*, 2009.

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

[EKD10] Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. BGP churn evolution : A perspective from the core. In *INFOCOM 2010. The 29th Conference on Computer Communications. IEEE*, 2010.

[EMS+07] William Enck, Patrick McDaniel, Subhabrata Sen, Panagiotis Sebos, Sylke Spoerel, Albert Greenberg, Sanjay Rao, and William Aiello. Configuration management at massive scale: system design and experience. In *ATC'07: 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*, pages 1–14, Berkeley, CA, USA, 2007. USENIX Association.

[FB05] Nick Feamster and Hari Balakrishnan. Detecting BGP configuration faults with static analysis. In *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, pages 43–56, Berkeley, CA, USA, 2005. USENIX Association.

[FBR03] Nick Feamster, Jay Borkenhagen, and Jennifer Rexford. Guidelines for interdomain traffic engineering. *SIGCOMM Comput. Commun. Rev.*, 33(5):19–30, 2003.

[FBR+04] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In *Future Directions in Network Architecture*, August 2004.

[FDPF10] P. Francois, B. Decraene, C. Pelsser, and C. Filsfils. Graceful eBGP Session Shutdown. Internet draft, draft-ietf-grow-bgp-gshut-02, work in progress, October 2010.

[FFEB05] Pierre Francois, Clarence Filsfils, John Evans, and Olivier Bonaventure. Achieving sub-second IGP convergence in large IP networks. *ACM SIGCOMM Computer Communication Review*, 35(3):33–44, July 2005.

[Fil07] Clarence Filsfils. BGP convergence in much less than a second. Presentation at Nanog 40, June 2007.

[FJB05] Nick Feamster, Jaeyeon Jung, and Hari Balakrishnan. An empirical study of "bogon" route advertisements. *SIGCOMM Comput. Commun. Rev.*, 35(1):63–70, 2005.

[FKMT04] Anja Feldmann, Hongwei Kong, Olaf Maennel, and Er Tudor. Measuring BGP pass-through times. In *Passive and Active Measurement Workshop (PAM*, pages 267–277, 2004.

[FL06] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632 (Best Current Practice), August 2006.

[FMR04] Nick Feamster, Zhuoqing Morley Mao, and Jennifer Rexford. BorderGuard: Detecting Cold Potatoes from Peers. In *Internet Measurement Conference*, Taormina, Italy, October 2004.

[FR09] Ashley Flavel and Matthew Roughan. Stable and Flexible iBGP. In *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, 2009.

[Gao01] Lixin Gao. On inferring Autonomous System relationships in the Internet. *Networking, IEEE/ACM Transactions on*, 9(6):733–745, Dec 2001.

[GGRW03] J. Gottlieb, A. Greenberg, J. Rexford, and J. Wang. Automated provisioning of BGP customers. *Network, IEEE*, 17(6):44–55, Nov.-Dec. 2003.

[GH05] T. Griffin and G. Huston. BGP Wedgies. RFC 4264 (Informational), November 2005.

[Gil06] Vijay Gill. Perspectives on Network Routing: Lessons Learned and Challenges for the Future – An Operational Overview. Presented at Infocom 2006 BGP Panel, http://www.ieee-infocom.org/2006/panelist/infocom-panel2-vijay.pdf, April 2006.

[GP01] T.G. Griffin and B.J. Premore. An experimental analysis of BGP convergence time. In *Network Protocols, 2001. Ninth International Conference on*, pages 53–61, Nov. 2001.

[GR01] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. *IEEE/ACM Trans. Netw.*, 9(6):681–692, 2001.

[GW99] Timothy G. Griffin and Gordon Wilfong. An Analysis of BGP Convergence Properties. In *in Proc. ACM SIGCOMM*, pages 277–288, 1999.

[GW00] T.G. Griffin and G. Wilfong. A Safe Path Vector Protocol. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 490–499 vol.2, 2000.

[GW02a] T. Griffin and G. Wilfong. Analysis of the MED oscillation problem in BGP. In *ICNP2002*, 2002.

[GW02b] T. Griffin and G. Wilfong. On the correctnes of iBGP configuration. In *SIGCOMM'02*, pages 17–29, Pittsburgh, PA, USA, August 2002.

[HA06] Geoff Huston and Grenville Armitage. Projecting future IPv4 router requirements from trends in dynamic BGP behaviour. In *In Australian Telecommunication Networks and Applications Conference (ATNAC)*, 2006.

[Hug04] D. Hughes. PACNOG list posting, December 2004.

[Hus] Geoff Huston. BGP Routing Table Analysis Report.

[Hus01] Geoff Huston. Analysing the Internet's BGP Routing Table. *Internet Protocol Journal*, 2001.

[Jac95] V. Jacobson. mrinfo, 1995. see `http://cvsweb.netbsd.org/bsdweb.cgi/src/usr.sbin/mrinfo/?only_with_tag=MAIN`.

[Jun06] Juniper. Junos 7.6. `http://www.juniper.net/techpubs/software/junos/junos76/index.html`, May 2006.

[KBC+06] D. Karrenberg, Randy Bush, Brett Carr, Niall O'Reilly, Ondrej Sury, Nigel Titley, Filiz Yilmaz, and Ingrid Wijte. IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region. RIPE-387, 2006.

[KFR06] J. Karlin, S. Forrest, and J. Rexford. Pretty Good BGP: Improving BGP by Cautiously Adopting Routes. In *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, pages 290–299, Nov. 2006.

[KKK07] Nate Kushman, Srikanth Kandula, and Dina Katabi. Can You Hear Me Now?! It Must be BGP. In *Computer Communication Review*, March 2007.

[KKKM07] Nate Kushman, Srikanth Kandula, Dina Katabi, and Bruce Maggs. R-BGP: Staying Connected in a Connected World. In *NSDI'07*, Cambridge, MA, April 2007.

[KLS00] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, April 2000.

[LABJ01] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet routing convergence. *IEEE/ACM Transactions on Networking*, 9(3):293–306, 2001.

[LBU09] Anthony Lambert, Marc-Olivier Buob, and Steve Uhlig. Improving internet-wide routing protocols convergence with mrpc timers. In *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 325–336, New York, NY, USA, 2009. ACM.

[LCC+09] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. A brief history of the Internet. *SIGCOMM Comput. Commun. Rev.*, 39(5):22–31, 2009.

[LCR+07] Karthik Lakshminarayanan, Matthew Caesar, Murali Rangan, Tom Anderson, Scott Shenker, and Ion Stoica. Achieving Convergence-Free Routing using Failure-Carrying Packets. In *SIGCOMM '07:*

Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, 2007.

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

[LGW+07] Jun Li, Michael Guidero, Zhen Wu, Eric Purpus, and Toby Ehrenkranz. BGP routing dynamics revisited. *SIGCOMM Comput. Commun. Rev.*, 37(2):5–16, 2007.

[LMIJ09] Craig Labovitz, Danny McPherson, and Scott Iekel-Johnson. Internet Observatory Report. NANOG 47, 2009.

[LMJ97] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Internet Routing Instability. In *SIGCOMM '97: Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 115–126, New York, NY, USA, 1997. ACM.

[LMJ99] Craig Labovitz, Gerald Malan, and Farnam Jahanian. Origins of Internet Routing Instability. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, 1999.

[MC04] R. Musunuri and J. A. Cobb. A complete solution for iBGP stability. In *IEEE International Conference on Communications*, 2004.

[MdSD+09] Pascal Merindol, Virginie Van den Schrieck, Benoit Donnet, Olivier Bonaventure, and Jean-Jacques Pansiot. Quantifying ASes Multiconnectivity using Multicast Information. In *Proc. ACM USENIX Internet Measurement Conference (IMC)*, November 2009.

[Mey06] D. Meyer. BGP Communities for Data Collection. RFC 4384 (Best Current Practice), February 2006.

[Mey08] Dave Meyer. The Locator Identifier Separation Protocol (LISP). *The Internet Protocol Journal*, 11(1), March 2008.

[MFCM10] P. Marques, R. Fernando, E. Chen, and P. Mohapatra. Advertisement of the best-external route in BGP. Internet Draft, draft-marques-idr-best-external-01, work in progess, February 2010.

[MFFR08] P. Mohapatra, R. Fernando, C. Filsfils, and R. Raszuk. Fast Connectivity Restoration Using BGP Add-path. Internet draft draft-pmohapat-idr-fast-conn-restore-00, September 2008.

[MGVK02] Z. M. Mao, R. Govindan, G. Varghese, and R. Katz. Route Flap Damping Exacerbates Internet Routing Convergence. In *ACM SIGCOMM'2002*, 2002.

[Mil84] D.L. Mills. Exterior Gateway Protocol formal specification. RFC 904 (Historic), April 1984.

[mul] BGP Multipath. Cisco online documention.

[MVdSD+09] Pascal Mérindol, Virginie Van den Schrieck, Benoit Donnet, Olivier Bonaventure, and Jean-Jacques Pansiot. Quantifying ASes Multiconnectivity using Multicast Information. In *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 370–376, New York, NY, USA, 2009. ACM.

[MWA02] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfigurations. In *ACM SIGCOMM 2002*, August 2002.

[MZF07] D. Meyer, L. Zhang, and K. Fall. Report from the IAB Workshop on Routing and Addressing. RFC 4984 (Informational), September 2007.

[nan97] Wow, as7007! Nanog discussion, 1997.

[Ng04] James Ng. Extensions to BGP to Support Secure Origin BGP (soBGP). Internet draft draft-ng-sobgp-bgp-extensions-02.txt, April 2004.

[OZP+06] R. Oliveira, B. Zhanf, D. Pei, R. Izhak-Ratzin, and L. Zhang. Quantifying Path Exploration in the Internet. In *Internet Measurement Conference*, Rio de Janeiro, Brazil, October 2006.

[PAMZ05] Dan Pei, Matt Azuma, Dan Massey, and Lixia Zhang. BGP-RCN: Improving BGP Convergence through Root Cause Notification. *Computer Networks*, 48(2):175 – 194, 2005.

[PJL+10] Jong Hang Park, Dan Jen, Mohit Lad, Shane Amante, Danny McPherson, and Lixia Zhang. Investigating occurence of duplicate updates in BGP announcements. In *Proc. Passive and Active Measurement*, 2010.

[Pos81] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349.

[PQU+10] Cristel Pelsser, B. Quoitin, S. Uhlig, T. Takeda, and K. Shiomoto. Providing scalable NH-diverse iBGP route redistribution to achieve sub-second switch-over time. *Computer Networks*, 2010.

[PVdM06] Dan Pei and Jacobus Van der Merwe. BGP convergence in Virtual Private Networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 283–288, New York, NY, USA, 2006. ACM.

[QdSFB09] Bruno Quoitin, Virginie Van den Schrieck, Pierre Francois, and Olivier Bonaventure. IGen: Generation of Router-level Internet Topologies through Network Design Heuristics. In *Proceedings of the 21st International Teletraffic Congress*, September 2009.

[Qiu] Jian Qiu. SimBGP : Python Event-driven BGP simulator.

[QPBU05] B. Quoitin, C. Pelsser, O. Bonaventure, and S. Uhlig. A performance evaluation of BGP-based traffic engineering. *International Journal of Network Management (Wiley)*, 15(3), May-June 2005.

[QU05] B. Quoitin and S. Uhlig. Modeling the routing of an Autonomous System with C-BGP. *IEEE Network*, 19(6), November 2005.

[QUP+03] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with BGP. *IEEE Communications Magazine Internet Technology Series*, 41(5):122–128, May 2003.

[RAM03] R. Raszuk, C. Appanna, and P. Roque Marques. IBGP Auto Mesh. Internet draft, draft-raszuk-idr-ibgp-auto-mesh-00.txt, work in progress, June 2003.

[RIS08] RIPE NCC RIS. YouTube Hijacking: A RIPE NCC RIS case study, 2008.

[RLH06] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006.

[Rou] University of Oregon Route Views Project. http://www.routeviews.org/.

[RWXZ02] Jennifer Rexford, Jia Wang, Zhen Xiao, and Yin Zhang. BGP routing stability of popular destinations. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 197–202, New York, NY, USA, 2002. ACM.

[SF07] G. Siganos and M. Faloutsos. Neighborhood Watch for Internet Routing: Can We Improve the Robustness of Internet Routing Today? In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1271–1279, 2007.

[sid] Secure Inter-Domain Routing (SIDR). IETF Working Group.

[SMS06] Wei Sun, Z.M. Mao, and K.G. Shin. Differentiated BGP Update Processing for Improved Routing Convergence. In *ICNP '06. Proceedings of the 2006 14th IEEE International Conference on Network Protocols, 2006.*, pages 280–289, Nov. 2006.

[SP06] Philip Smith and Christian Panigl. RIPE Routing Working Group - Recommendations on Route-flap Damping. may 2006.

[TMS01] P. Traina, D. McPherson, and J. Scudder. Autonomous System Confederations for BGP. RFC 3065 (Proposed Standard), February 2001. Obsoleted by RFC 5065.

[TSGV04] Renata Teixeira, Aman Shaikh, Tim Griffin, and Geoffrey M. Voelker. Network sensitivity to hot-potato disruptions. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 231–244, New York, NY, USA, 2004. ACM.

[UT06] S. Uhlig and S. Tandel. Quantifying the impact of route-reflection on BGP routes diversity inside a tier-1 network. In *IFIP Networking 2006*, Coimbra, Portugal, May 2006.

[VC07] Q. Vohra and E. Chen. BGP Support for Four-octet AS Number Space. RFC 4893 (Proposed Standard), May 2007.

[VCG98] C. Villamizar, R. Chandra, and R. Govindan. BGP Route Flap Damping. RFC 2439 (Proposed Standard), November 1998.

[VGE00] Kannan Varadhan, Ramesh Govindan, and Deborah Estrin. Persistent Route Oscillations in Inter-Domain Routing. *Computer Networks*, 2000.

[VPB08] Laurent Vanbever, Grégory Pardoen, and Olivier Bonaventure. Towards validated network configurations with NCGuard. In *Proc. of Internet Network Management Workshop 2008*, Orlando, USA, October 2008.

[VQB09] Laurent Vanbever, Bruno Quoitin, and Olivier Bonaventure. A hierarchical model for BGP routing policies. In *PRESTO '09: Proceedings of the 2nd ACM SIGCOMM workshop on Programmable routers for extensible services of tomorrow*, pages 61–66, New York, NY, USA, 2009. ACM.

[VVKB06] M. Vutukuru, P. Valiant, S. Kopparty, and H. Balakrishnan. How to Construct a Correct and Scalable iBGP Configuration. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[WG08] F. Wang and L. Gao. A Backup Route Aware Routing Protocol - Fast Recovery from Transient Routing Failures. *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 2333–2341, 13-18 April 2008.

[WG09] Feng Wang and Lixin Gao. Path Diversity Aware Interdomain Routing. In *INFOCOM 2009, IEEE*, pages 307–315, April 2009.

[WGWQ05] Feng Wang, Lixin Gao, Jia Wang, and Jian Qiu. On Understanding of Transient Interdomain Routing Failures. In *ICNP '05*, pages 30–39, Washington, DC, USA, 2005. IEEE Computer Society.

[WMRW05] Jian Wu, Zhuoqing Morley Mao, Jennifer Rexford, and Jia Wang. Finding a needle in a haystack: pinpointing significant BGP routing changes in an IP network. In *NSDI'05*, pages 1–14, Berkeley, CA, USA, 2005. USENIX Association.

[WMS04] Russ White, Danny McPherson, and Srihari Sangli. *Practical BGP*. Addison-Wesley, 2004.

[WMW+06] Feng Wang, Zhuoqing Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. A measurement study on the impact of routing events on end-to-end internet path performance. In *SIGCOMM '06*, pages 375–386, New York, NY, USA, 2006. ACM.

[WRC09a] D. Walton, A. Retana, and E. Chen. Advertisement of Multiple Paths in BGP. Internet draft, draft-walton-bgp-add-paths-05.txt, work in progress, 2009.

[WRC09b] D. Walton, A. Retana, and E. Chen. BGP Persistent Route Oscillation Solution. http://tools.ietf.org/html/draft-walton-bgp-route-oscillation-stop-02, 2009.

[WSGP07] Lan Wang, M. Saranu, J.M. Gottlieb, and Dan Pei. Understanding BGP Session Failures in a Large ISP. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 348–356, May 2007.

[WWGQ09] Feng Wang, Jia Wang, Lixin Gao, and Jian Qiu. On Understanding Transient Interdomain Routing Failures. *Transactions on Networking*, 17(3):740–751, June 2009.

[WZP+02] Lan Wang, Xiaoliang Zhao, Dan Pei, Randy Bush, Daniel Massey, Allison Mankin, S. Felix Wu, and Lixia Zhang. Observation and analysis of BGP behavior under stress. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 183–195, New York, NY, USA, 2002. ACM.

[XGF07] J. Xia, L. Gao, and T. Fei. A Measurement Study of Persistent Forwarding Loops on the Internet. *Computer Networks*, 51(17):4780–4796, December 2007.

[XWN03] Li Xiao, Jun Wang, and K. Nahrstedt. Reliability-aware IBGP route reflection topology design. In *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on*, pages 180–189, Nov. 2003.

[ZB03] R. Zhang and M. Bartell. *BGP Design and Implementation : Practical guidelines for designing and deploying a scalable BGP routing architecture*. CISCO Press, 2003.

[Zin02] Alex Zinin. *Cisco IP Routing: Packet Forwarding and Intradomain Routing Protocols*. Addison Wesley Professional, 2002.

[ZJP+07] Changxi Zheng, Lusheng Ji, Dan Pei, Jia Wang, and Paul Francis. A light-weight distributed scheme for detecting IP prefix hijacks in real-time. *SIGCOMM Comput. Commun. Rev.*, 37(4):277–288, 2007.

[ZKL+05] Beichuan Zhang, Vamsi Kambhampati, Mohit Lad, Daniel Massey, and Lixia Zhang. Identifying BGP routing table transfers. In *MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 213–218, New York, NY, USA, 2005. ACM.

[ZMZ04] B. Zhang, D. Massey, and L. Zhang. Destination reachability and BGP convergence time. In *IEEE GLOBECOM*, December 2004.

[ZPW+01] Xiaoliang Zhao, Dan Pei, Lan Wang, Dan Massey, Allison Mankin, S. Felix Wu, and Lixia Zhang. An analysis of BGP multiple origin AS (MOAS) conflicts. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 31–35, New York, NY, USA, 2001. ACM.

[ZZH+08] Zheng Zhang, Ying Zhang, Y. Charlie Hu, Z. Morley Mao, and Randy Bush. Ispy: detecting ip prefix hijacking on my own. In *SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pages 327–338, New York, NY, USA, 2008. ACM.

[ZZHM07] Zheng Zhang, Ying Zhang, Y. Charlie Hu, and Z. Morley Mao. Practical defenses against BGP prefix hijacking. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2007. ACM.