

Virtualization and Distribution of the BGP Control Plane

Mihaela Iuniana Oprescu

▶ **To cite this version:**

Mihaela Iuniana Oprescu. Virtualization and Distribution of the BGP Control Plane. Networking and Internet Architecture [cs.NI]. Institut National Polytechnique de Toulouse - INPT, 2012. English. NNT : 2012INPT0126 . tel-04282412v2

HAL Id: tel-04282412

https://theses.hal.science/tel-04282412v2

Submitted on 13 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Réseaux, Télécommunications, Systèmes et Architecture

Présentée et soutenue par :
Mihaela Iuniana Oprescu

le : jeudi 18 octobre 2012

Titre :

Virtualization and Distribution of the BGP Control Plane

JURY
Chadi Barakat
Abdelhamid Mellouk
Philippe Owezarski
Ana Rosa Cavalli
Damien Magoni

Ecole doctorale :
Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :
Orange Labs & LAAS-CNRS

Directeur(s) de Thèse :
Philippe Owezarski
Mickaël Meulle

Rapporteurs :
Ana Rosa Cavalli
Damien Magoni

Acknowledgement

I would like to start with a big THANK YOU that goes to Mickaël Meulle for putting up with my ideal quest of revolutionary research and for the unpaid counseling that he offered during the years at R&D. Taking into account the effort he invested, this thesis belongs to him as much as it belongs to me. Things would not have been possible without the support of Philippe Owezarski who decided to take official responsibility for the work conducted during my days in the Orange Labs and the visits at LAAS-CNRS. Going farther back in time, I am indebted to Joël Lattmann who introduced me to the world of virtualization during my first year. Of course, all of this started because of (or maybe I should say thanks to) Sarah Nataf who managed to trick everyone into thinking that I was the right candidate for the PhD subject she and Franck Carmine had decided to put up.

It was a pleasure working with people like Steve Uhlig, Cristel Pelsser and Olaf Maennel and they proved to be a priceless support for a young researcher trying to fit into the global picture. I also thank the members of the jury, especially Ana Cavalli and Damien Magoni who wrote overwhelmingly positive reports about my manuscript.

My full gratitude goes to the RIV team that offered a really good environment, with people who criticized all possible technical and typography aspects only so that I could improve the work I had taken the courage to present in front of them (which didn't happen very often, I must admit). The biggest encouragement I received was from Bruno Decraene who told me, after reading one of my articles, "oh, it's rather well written and the ideas are not completely insane". Being a part of this team was an advantage and I can say that I have learned a lot, especially during the coffee breaks when subjects would range from the *ascenseur spatial* to the very nitty gritty details of the router testbed and the current debates at the IETF. I would like to point out Fred's disarming cynicism and sometimes funny jokes, Marc's kindness and encouragement (he will always be a *chic type*, so beware), Sovatha and David's helpfulness when I needed assistance and the seriousness and down-to-earth demeanor of Jean-Luc Lutton.

There have been people who showed up along the way, like my sacrificed office mate, Laurent Valeyre and Jaqueline Queiroz with whom I am tied by a special friendly bond. I cannot omit Luiz and Paul – a duo of terror, Adrien and Sidali – *les plus beaux*, Liang who cheered us up, or Greg and Benoît who represented rugby and soccer. Let us not forget the apprentices and interns who made me feel better because they were the only category lower than the PhD student: Miriem, Benjamin, Lamia, Thao, Gaurav and Waqas.

I am also grateful to the line of PhD students (luckily, now they are all PhDs) who came before me, gave me useful advice about `phdcomics.com` and talked me into taking light drugs (chocolate, beer and other such substances): Anthony and Zied. A special thought goes to Marc-Olivier (aka Herr Doktor) who not only took the time to actually read this memoir, but also gave me precious ideas about how to make it better.

And the one person who managed to keep all the team together was Frank Carmine, who believed in me when I didn't and who has been a tremendous help. The list would not

be complete without my current colleagues at NAD [1]: Pierre Combescure has been a role model and a father figure to me since I joined the new team and Slim Gara has constantly urged me to finish my PhD and backed me up in this new professional challenge I faced as a new member of his team. Life would be sad without my colleague Antonio Montani Jimenez, an outgoing and optimistic young man, or without Davide Donato who always has a good story. I would also like to express my consideration for Bruno Morvan, Frédéric Thibord, Jean-Pierre Nicolazzi and Christof Schmitz who are all nice fellows and occasional gardeners, Laurence Bathany, Dominique Delisle and Sophie Nachman-Ghnassia who I am sure are impatient about attending the *pot de thèse*.

There is someone who does not fit in any of the above clusters. Jean-Yves Couty is simply too much to handle, but I would like to thank him for the crazy poems and for the times when he would quietly get out of my office and let me be stressed about my work.

I am certainly letting out a lot of people who have supported my twisted path in the research world, but I cannot forget my friends Răzvan Stănică, Oana Vlăduț, Robert Guduvan, Miruna Stoicescu and Raluca Maria Indre who successfully finished or who successfully still conduct PhD work themselves. I will always remember the painful question "how's your thesis coming along?" from Ștefania Roșu, Paula Borțosu, Diana Stoica, Alexandra Pencea and Ioana Mul.

The one person who was actually more stressed than I was about this work is Guillaume Gaulon. It will be a great relief for you and I would like to let you know how much I appreciate the hours that you have spent with me doing network debugging and getting simulations to work, but in a researchy way because the network is supposed to oscillate and not work properly.

I cannot conclude without thanking my family: my mother Ica, my father Dorel and my brother Bogdan to whom I wish a lean path and good wind for the journey he is on.

My best wishes go to the next generations of PhD students who will likely cross the same state of mind as all the previous PhD students. After sleepless nights and long hours when all hope is gone, do not despair... you will always have chocolate!

1. Network Architecture and Design division of France Telecom

Contents

1 Introduction **1**

1.1 Background 1

1.2 Main Research Contributions 3

1.3 Publications 4

2 Routing in the Internet **5**

2.1 General Background 5

2.2 End-to-end communication 6

2.2.1 What is routing? IGP and EGP 7

2.2.2 IP prefixes 9

2.2.3 Routing versus Forwarding 10

2.3 The Border Gateway Protocol 11

2.3.1 External BGP and Internal BGP sessions 12

2.3.2 Learning routes in BGP 13

2.3.3 Best Path selection 14

2.3.4 BGP policies 17

2.3.5 Route Servers 20

2.3.6 iBGP architectures 21

2.4 Conclusion 24

3 Flaws and fixes in BGP routing **25**

3.1 Current BGP Plagues 25

3.1.1 Scalability 26

3.1.2 Correctness 29

3.1.3 Path diversity . . . 36
3.1.4 Convergence time and path exploration . . . 39
3.1.5 Management and troubleshooting . . . 41
3.2 Routing Platforms to fix iBGP . . . 42
3.3 Summary and Remaining Issues . . . 43

4 The oBGP Solution 47
4.1 Overview . . . 47
4.2 Graph models . . . 49
4.2.1 The IGP graph . . . 50
4.2.2 The iBGP graph . . . 50
4.2.3 The oBGP graph . . . 51
4.3 Design principles . . . 52
4.3.1 Distributed sub-planes . . . 54
4.3.2 Index of Virtual Prefixes . . . 55
4.3.3 Allocation of prefixes to sub-planes . . . 56
4.4 General architecture . . . 59
4.4.1 Network view . . . 59
4.4.2 Sub-plane view . . . 61
4.4.3 Client view . . . 62
4.5 Gain through Design . . . 62

5 Resilient architectures 65
5.1 Redundancy and replication . . . 65
5.2 The 1:1 redundancy scheme . . . 66
5.3 The 1+1 redundancy scheme . . . 69
5.4 Failure cases . . . 70
5.4.1 Node failure . . . 71
5.4.2 Site failure . . . 75
5.4.3 Other failures . . . 76
5.5 Final considerations . . . 77

6 Practical oBGP 79

6.1 The dVirt test platform 79
6.1.1 dVirt Overview 80
6.1.2 dVirt Management Network 80
6.1.3 Virtual Routers and Virtual Ethernets 82
6.1.4 Simulated BGP Network 83
6.2 The oBGP hub 85
6.2.1 The example of a virtualized PoP 87
6.3 Closing remarks 87

7 Evaluation **89**
7.1 Sizing rules 89
7.1.1 Table size 91
7.1.2 Number of Sessions 96
7.1.3 Additional equipment 97
7.2 Convergence time and correctness 98
7.3 Migration scenario 101
7.4 Global assessment 102

8 Conclusion **103**
8.1 Future Work 105

9 Version française abrégée **107**

Appendix **125**

Bibliography **127**

Chapter 1

Introduction

1.1 Background

The Internet is a successful experiment that escaped from the lab. In its beginnings it was meant as a research playground for the American army, but the Internet expanded to encompass a few universities, then it opened up to the wide world through the commercial companies that had grasped its potential. Nowadays, its strong impact on the everyday lives of millions of people is undeniable.

The Internet supports a very diverse range of traffic types and services while at the same time fostering an increasing number of users and networks. In order for billions of connections and seamless information transfers to be possible, one protocol is in charge of holding it all together: the Border Gateway Protocol (BGP). For the many users who need to cross the borders of their local Internet Service Provider, BGP is the universal language allowing all heterogeneous networks to understand each other and be able to relay routing information all the way to foreign destinations.

Although on the surface routing seems to be a trivial graph theory problem, BGP was not designed with a mathematical model in mind, but as a set of rules that give network operators the liberty to express routing policies. This means that real-life BGP offers no guarantees of convergence or correctness form the point of view of routing algebras. On the contrary, many additional standards were issued to solve operational problems as they showed up. Additionally, BGP does not rely on a mere shortest path computation, like internal routing protocols running within a network; BGP offers to the engineering teams the tools for manifesting preferences based on economical interest (e.g., following the logic dictated by peering agreements), for avoiding certain Internet market players (e.g., need to circumvent specific networks due to bad quality of the transit service) or for reinforcing political or government-dictated restrictions (e.g., elude crossing certain parts of the world due to highly sensitive traffic, censorship etc.).

The timeline [1] in Fig. 1.1 attempts to show a schematic view of the efforts put into achieving a global consensus and vision of methods for expressing routing policies. The second half of this evolution pictures predominantly the changes brought to BGP in order to sustain the needs in terms of scalability, correctness, resilience and increasingly diverse network services.

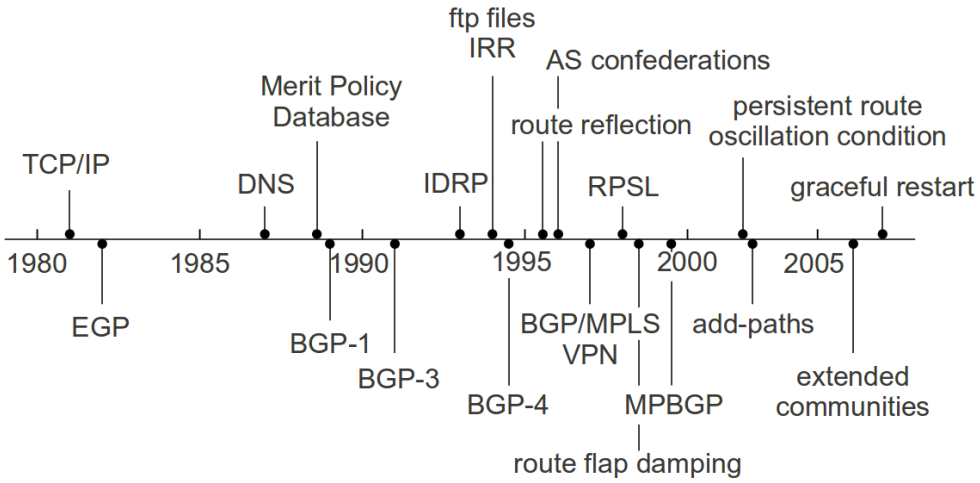


Figure 1.1 – A simplified timeline emphasising BGP policy advancements and the general evolution of the protocol with all its more recent additional features.

Since the Internet would get severed if network operators decided to no longer speak the common language, the major changes in architecture have occurred inside the ISP network, mostly affecting internal BGP (iBGP). The introduction of route reflection and AS confederations seems the right answer to the problem of the growing network size and the inherent full mesh of iBGP sessions that become hard to manage in big networks. However, these two options come with a set of flaws of their own such as persistent routing oscillations, deflections, forwarding loops etc.

Since various services supported by the Internet are highly sensitive to network outages or degraded quality, both network operators and the research community strive to find solutions and workarounds to cope with the demanding expectations of Internet users. Efforts have been put into investigating new architectures or protocols for achieving scalability [Ballani *et al.*, 2008 ; Sarakbi and Maag, 2010 ; Ben Houidi and Meulle, 2010], formal mathematical models for better understanding the BGP behavior [Feamster *et al.*, 2004b ; Metarouting, 2011 ; Vutukuru *et al.*, 2006], more specific insights on protocol correctness [Griffin and Wilfong, 2002b ; Griffin and Sobrinho, 2005 ; Buob *et al.*, 2007 ; Buob *et al.*, 2008] or loss of route diversity [Uhlig and Tandel, 2006 ; Uhlig and Tandel, 2005] and alternative options for improving it [Pelsser *et al.*, 2008 ; Van den Schrieck and François, 2009 ; Bornhauser *et al.*, 2011]. Some of the proposed solutions rely on sets of constraints applicable to the router configuration or consist of new paradigms such as routing platforms

1. Timeline inspired by Susan Hares' presentation "15 years of Policy Routing" at NANOG 30 in 2004

that work in a more centralized manner [Koponen *et al.*, 2010 ; Pelsser *et al.*, 2009 ; Caesar *et al.*, 2005].

1.2 Main Research Contributions

In this work we tackle both scalability and route diversity issues raised in the context of internal BGP routing, while at the same time optimizing the process of BGP route redistribution within a network. The proposed solution is oBGP, an overlay BGP routing platform in charge of collecting all the routes received from the eBGP peers and the internally available routes in order to compute the BGP decision process in a distributed manner. Unlike today's model, when all the routers receive the best routes towards all available destinations, the idea behind oBGP is to give a subset of all the routing information to a subset of the routers. These enhanced routers, called oBGP nodes, are in charge of computing the best routes on behalf of client routers. Marking a difference with other routing platforms, oBGP divides and dispatches the routing data according to a precise mapping, thus reducing the amount of routes on each node. As a consequence, the oBGP nodes can perform best route selections in parallel due to the division of the routing information across several routers. The oBGP platform, as a single entity, gains higher visibility than the classic network routers, so the routes selected on behalf of the platform's clients are more coherent from a global point of view and more adapted to the client's position in the global topology graph.

The contributions of this thesis are outlined as follows:

1. An analysis of current problems that have arisen in BGP internal routing shows that there is still room for improvement when it comes to scalability, correctness and diversity of the propagated routes. Next to a literature survey, Chapter 3 contains an example of the loss of diversity in the case of routes entering the BGP decision process within a real transit network.
2. Chapter 4 presents the oBGP concept and how it manages to handle the propagation of Internet destinations with a new routing platform architecture. The sub-plane and virtual prefix notions are introduced, while the final sections articulate the entire oBGP model as perceived by the clients and by the nodes carrying the different sub-planes or more generally, at the network level.
3. Operational constraints are taken into account in Chapter 5 where two implementation schemes are detailed. Multiple failure scenarios are investigated and their effects on the routes' propagation. Without any major modifications to the existing equipment, a practical deployment is achievable with the help of virtualization techniques and relying on the dVirt platform, as shown in the Chapter 6.
4. Finally, an analytical evaluation of the oBGP solution is provided in Chapter 7. The graphs in this chapter allow for a more precise understanding of the tradeoffs involved in the oBGP design, emphasizing the different results obtained for small, medium and big network topologies.

1.3 Publications

The full text of publications and the posters can be found online at `www.iuniana.ro`.

International conferences

1. Iuniana Oprescu, Mickaël Meulle, Steve Uhlig, Cristel Pelsser, Olaf Maennel, Philippe Owezarski *Rethinking iBGP Routing,* ACM SIGCOMM (Special Interest Group on Data Communications) poster session, New Delhi, India, August 2010.
2. Iuniana Oprescu, Mickaël Meulle, Steve Uhlig, Cristel Pelsser, Olaf Maennel, Philippe Owezarski *oBGP: an Overlay for a Scalable iBGP Control Plane,* IFIP Networking, Valencia, Spain, May 2011.
3. Iuniana Oprescu, Mickaël Meulle, Philippe Owezarski *dVirt: a Virtualized Infrastructure for Experimenting BGP Routing,* IEEE LCN (Local Computer Networks), Bonn, Germany, October 2011.

Workshop presentations

1. *Virtualisation et distribution du plan de contrôle BGP, (in French)* Journées automnales ResCom, Lyon, France, November 2010.
2. *Virtualizarea şi distribuirea planului de control BGP, (in Romanian)* Electronica, Telcommunicaţiile şi Teoria Informaţiei în lume şi în ţară, Bucarest, Romania, September 2010.
3. *Virtualisationdu réseau, (in French)* Journées des doctorants du LAAS-CNRS, Toulouse, France, April 2009.

Patents

1. *Qualité de service dans les réseaux virtuels, (in French)* INPI Patent No. 07501.

Others

1. Mickaël Meulle, Iuniana Oprescu, Joël Lattmann, Jean-Louis Simon *Towards full virtualization of networks?* Orange Labs White Paper, October 2009.

Chapter 2

Routing in the Internet

This chapter sets out to introduce a few basic concepts related to the Internet structure and routing protocols, while concentrating more specifically on the Border Gateway Protocol (BGP). Details are provided about the BGP decision process, emphasizing the differences between a protocol that provides simple routing information and one that can express routing policies.

Further on, the reader can discover a brief presentation of the way that networks interconnect in the global Internet, the tiered hierarchy and the different relationships between the domains connected in Internet Exchange Points. Zooming on the way protocols are organized within a specific network, the last paragraphs are about BGP internal architectures: the full mesh of sessions, confederations and route relfection.

2.1 General Background

The purpose of the Internet is to allow end users to communicate, despite the fact that there may not be a direct connection between the end points. Nowadays, the Internet is a part of everyday life in the modern culture: it supports a wide range of services such as e-mail, web, instant messaging, banking, e-commerce, video on demand, telephony, blogging, social networking and has become a crucial source of information. The Internet has modified the perception we have of computers and information technology in general, making it more accessible at a large scale. It has reshaped the way humans interact and it has strongly influenced communication, being an important vector in events such as the revolutions in Tunisia and Egypt or by playing a role in disseminating information in countries where censorship is enforced.

The success of the Internet can be easily demonstrated by the extended usage and penetration rates, going as far as 1,987.0% growth between 2000 and 2011 in the Middle East region. On March 31 of 2011, 78.3% of North Americans were using the Internet and 30.2% of the world population was connected [Internet World Stats, 2011]. The Internet started as a common platform that would allow a few American universities to exchange research

data in the early 1970's. Facing a continuous growth in the number of users and thus important scalability challenges, the original design and purpose have greatly evolved, leading to what we use today: a worldwide logical network that links a multitude of heterogeneous physical networks.

Regular users will often mistake the Internet for the World Wide Web service, imagining that beyond the web browser used for navigation there is a magic cloud that delivers the vast on-line information. The following sections unveil what the actual structure of the Internet is and how it is possible to achieve end-to-end connectivity.

2.2 End-to-end communication

To obtain an Internet access, consumers usually resort to an Internet Service Provider (ISP) that produces the technical means such as dial-up or broadband connection, satellite or a Local Area Network (LAN). If Alice is connected to a French ISP and she is trying to reach Bob who is connected to a North American ISP, then she needs her French ISP to be in its turn connected to other carriers that will enable the data to travel all the way to Bob.

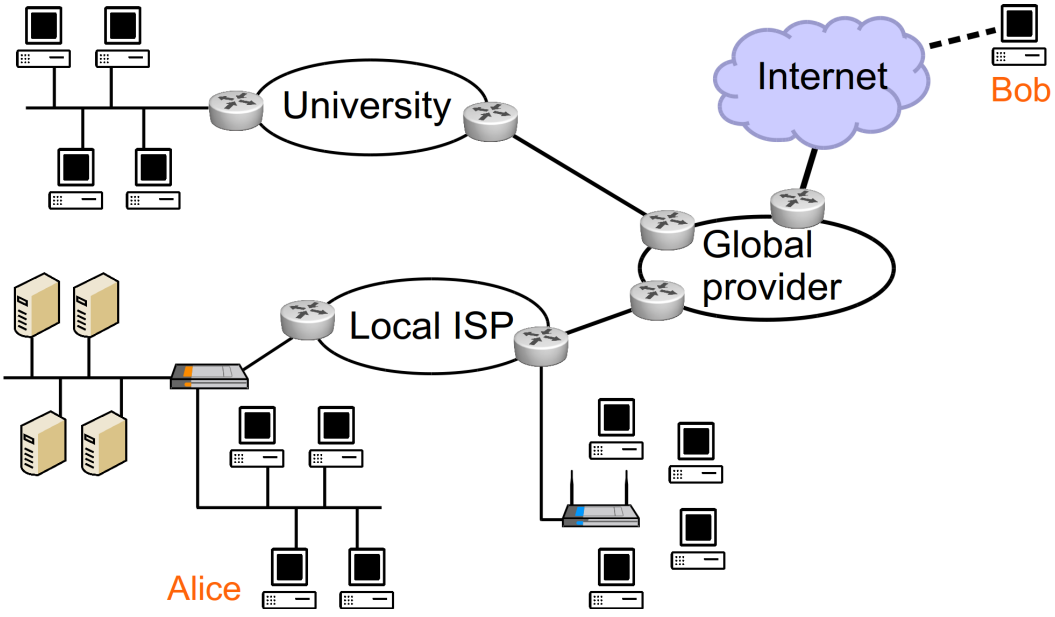


Figure 2.1 – The structure of the Internet: from the end host to the international provider

The Internet is in fact a collection of more than 40,000 [Huston, 2011a] different networks called Autonomous Systems (ASes). An AS is a network controlled by a single administrative entity such as an ISP, a university, a commercial enterprise, an organization or a content provider. An AS is not geographically restricted and there is no real one to one correspondence between an ISP and an AS (e.g., France Telecom is in charge of more

than 150 ASes). AS numbers are 16-bit and more recently 32-bit integers assigned by the Internet Assigned Numbers Authority (IANA) and they uniquely identify a network with a well defined routing policy. To illustrate the concept, let us take a look at some networks belonging to France Telecom:

- AS 5511 OpenTransit (OTIP) is the international carrier that offers transit to other ISPs or content providers. The OTIP network is ranked as part of the top 20 networks of the Internet [Meulle, 2007].
- AS 3215 Réseau Backbone et Collecte Internet (RBCI) corresponds to the residential network federating the ADSL [1] and FTTH [2] clients of Orange.
- AS 25186 Réseau d'Accès des Entreprises à l'Internet (RAEI) is dedicated to business services, such as BGP/MPLS IP VPNs [3].

ASes are interconnected and hopefully, any to any communication is possible. There are no topology constraints and the construction is typically flat, with no hierarchy, building a web of ASes. How can one manage to find and transmit information between two remote hosts located within two of the 40,000 ASes? This is when routing steps in, allowing users located in different networks to get connected and achieve seamless communication.

2.2.1 What is routing? IGP and EGP

A *name* indicates what we seek. An *address* indicates where it is.
A *route* indicates how we get there. – Jon Postel

We know the Domain Name System translates names into addresses. In the same way, routing is in charge of selecting a path in a network to reach a given address. As previously seen, the data packets that cross the Internet from one source to a destination will sometimes travel through several networks. Even within a single ISP's network, there are multiple equipments called routers that are linked together in a topology. This means that information is sent through the network on a **hop by hop** basis. Routing is the process of computing the next hop that a packet should go through in order to reach the desired destination.

Routing can be divided into two categories of protocols that work together to allow for seamless communication across the Internet:

- **Interior Gateway Protocol** (IGP) used for exchanging routing information within an AS. The IGP is visible only to routers within the AS and it does not have any impact on the neighboring ASes. An operator may choose to use any of the available protocols, the most widely deployed being Intermediate System to Intermediate System (IS-IS) [Oran, 1990] or Open Shortest Path First (OSPF) [Moy, 1998].
- **Exterior Gateway Protocol** (EGP) in charge of handling network reachability between distinct ASes. The de facto standard in inter-domain routing is the Border Gateway Protocol (BGP-4) [Rekhter *et al.*, 2006] and it is the only EGP used in the current

1. Asymmetric Digital Subscriber Line
2. Fiber To The Home
3. Border Gateway Protocol/Multiprotocol Label Switching Internet Protocol Virtual Private Network

Internet.
It is worthy of mentioning that BGP relies on results provided by the IGP in order to resolve routes within an AS.

These two types of protocols allow routing packets to internal destinations — done by IGPs — and to external destinations in different ASes — done by BGP. This hierarchical distinction comes from a need to achieve scalability in the Internet and avoid spreading information about the routing policies, the protocols or the topology of a proprietary AS.

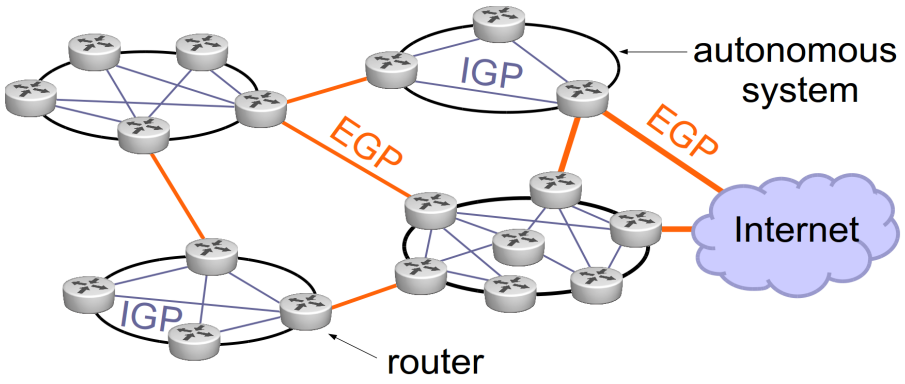


Figure 2.2 – ASes, Routers, IGPs and EGPs working together

IGPs and EGPs have different objectives and resort thus to different methods: an IGP is aware of the entire network topology whereas an EGP hides any detailed information about the internal topology of an AS. As a result, another classification of routing protocols is possible according to the applied algorithms:

- **link-state** routing protocols where each router has knowledge of the full network topology and can independently compute routes to any destination within the network. A node participating in a link-state routing protocol can reconstruct the connectivity map thanks to a flooding mechanism: a router propagates information about its links to all the connected neighbors that will further relay it and so on. Link-state protocols define metrics or costs of links and use a shortest path algorithm such as Dijkstra to determine the shortest path to every other node in the graph.
- **distance-vector** routing protocols where routers exchange reachability information about destinations (next hop routing) and an associated distance metric. This class of protocols uses algorithms such as Bellman-Ford and Ford-Fulkerson.
  In this work we concentrate on BGP which is part of the distance-vector family of protocols. Each BGP router sends to the connected neighbors its current routing table or incremental updates when changes are detected in the topology. Further details about the BGP mechanisms are provided in section 2.3.

2.2.2 IP prefixes

The high adoption rates of the Internet are also due to a common protocol stack, namely the Transmission Control Protocol and the Internet Protocol (TCP/IP), that has enabled billions of users to technically interconnect. The Internet Protocol Suite is based on a layered architecture, with the Network Layer as a thin waist of the hourglass depicting the communication protocols. The Network Layer mainly consists of the Internet Protocol, defining thus the fundamental addressing namespaces: 32-bit numbers for the Internet Protocol version 4 (IPv4) and more recently, 128-bit addresses for Internet Protocol version 6 (IPv6). To render the binary format readable to humans, the IPv4 address is expressed as decimals going from 0 to 255 separated by dots. A valid IPv4 address looks something like 203.0.113.98. An IPv6 address is conventionally expressed using hexadecimal strings, an example could be 2007:cafe::dead:beef.

IP is the primary brick that builds the Internet, allowing to identify and locate hosts in a network with an IP address. Each Internet user enjoys two basic primitives: connectivity and reachability. Connectivity is the property of being able to access the other users and the content of the Internet, whereas reachability is a notion implying that all Internet users and content providers are able to transmit data packets to a given user. To be reachable, a user must be identified by a public IP address that is routable in the Internet.

We have previously seen that each host in the Internet has at least one public IP address and that the routers in the core of the network handle the data packets based on their destination address. A router is generally connected to several neighbor routers, it has multiple links and receives multiple routes to a given destination. A Routing Information Base (RIB) is a mapping table between all reachable destinations and the associated neighbor router. If a router has no input about a specific destination in its RIB, it does not "know" how to handle the routing and the information packet is simply discarded.

To avoid losing packets, it is sufficient for each router to be aware of all the destinations in the Internet. Since every end user and also the network equipment is identified by an IP address, does this mean a router must keep a record for each of the almost 4 billion reachable addresses of the Internet? Hopefully not, since it is possible to aggregate multiple contiguous IP addresses into bigger blocks called *IP prefixes* [Fuller and Li, 2006]. For example, the prefix **203.0.113.**0/24 refers to all the IP addresses starting with the same set of 24 bits: from 203.0.113.0 to 203.0.113.255. There are as many as 400,000 prefixes [Huston, 2011b] seen in the current Internet.

It is equally possible to break these blocks of variable size into smaller prefixes. In the case of an IP address contained by two prefixes, a router will prefer the longest prefix match. Confronted with a packet going to destination 203.0.113.98, if a router has the choice between routing towards **203.0.113.**0/24 and **203.**0.0.0/8, it will select the first route as being the most specific one. A RIB often contains a default route that handles all destinations that do not match any of the existing entries.

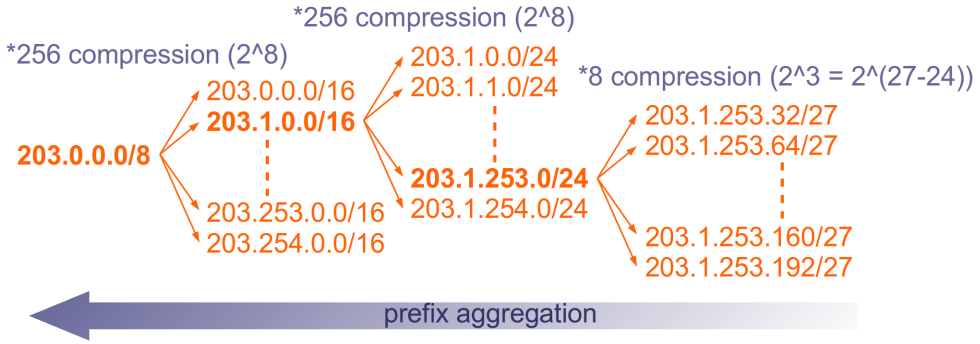


Figure 2.3 – Examples of IPv4 prefix (de)aggregation

2.2.3 Routing versus Forwarding

When a router receives an IP packet, which exit interface to choose towards which neighbor? The answer is stored in a forwarding table or Forwarding Information Base (FIB) that holds a logic correspondence between a destination IP address and a local interface. Based on the available RIB entries, a routing protocol determines the best route and installs it in the FIB. To put it simply, a RIB contains all the routes "known" by a router and a FIB keeps only the routes "in use".

Figure 2.4 – Network topology

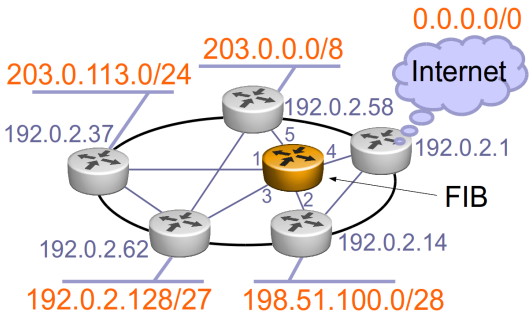


Figure 2.5 – Forwarding table example

destination prefix	next hop	interface
203.0.0.0/8	192.0.2.58	5
203.0.113.0/24	192.0.2.37	1
192.0.2.128/27	192.0.2.62	3
198.51.100.0/28	192.0.2.14	2
0.0.0.0/0	192.0.2.1	4

An important distinction arises between the process of finding the correct route in a network and the action of placing the bits that enter router interface A on interface B directed towards the right neighbor.

There is a fundamental difference between routing and forwarding:

- **routing** (control plane of a router) is in charge of computing the path that data packets will take. It is mainly illustrated by routers exchanging intelligence and individually performing a selection algorithm called the decision process. The decision process ranks the entries of the routing table according to specific criteria (e.g., shortest path, link capacity, financial revenue...) and thus determines one best route that gets installed in

the forwarding table.
- **forwarding** (data plane of a router) is the action of directing data packets between an incoming interface to an outgoing interface. Each router swaps the bits according to its forwarding table.

A third plane exists, called the management plane that is required for remotely commanding and monitoring the equipments. The management plane allows the engineering teams to push configurations and to check the status of the network elements through protocols such as SNMP (Simple Network Management Protocol). These aspects of routing are outside the scope of the work presented here.

2.3 The Border Gateway Protocol

The Border Gateway Protocol [Rekhter *et al.*, 2006] is the actual standard that enables the computation of paths in the Internet. End-to-end communication is possible through the exchange of reachability information. BGP allows a router named a BGP speaker to route traffic towards destinations located in other ASes. In addition, the protocol offers a framework for implementing individual routing policies that are specific to every domain of the global Internet.

Given that BGP assures inter-domain routing in the Internet core, it does not maintain detailed information about every AS that it needs to traverse. Network operators do not want to disclose sensitive data such as topology design, security policies, traffic engineering rules and so on. BGP hides the complexity of the network and the implemented design choices by delivering only reachability information: it is a part of the distance-vector class of protocols.

The messages exchanged for routing take into account two components: *a direction* representing a next hop address associated to an exit interface and *a distance* that quantifies the cost of reaching a certain destination, like the number of hops. The BGP next hop is in fact an AS Border Router (ASBR) that can route the external prefix towards its destination. An important requirement is that the address for the exit point of the AS should be known to iBGP, resolvable through the IGP or statically configured. This means that a BGP router will always reject all routes pointing towards a next hop that is not attainable. As a consequence, all the routes stored in the RIB are correlated with the network topology. Hop by hop, the data packet is routed closer to the exit point until it leaves the AS.

In the example depicted in Fig. 2.6, the AS 64499 announces to the Internet that it can reach a new prefix, 203.0.113.0/24 directly. When the announcement reaches AS 64500, the ASBR advertises in its turn that it can reach the new prefix and it appends its own AS number to the AS path. Further on, AS 64511 informs its neighbors that its new best path for reaching 203.0.113.0/24 is through AS 64500 and AS 64499.

More specifically, BGP extends the distance-vector class to path-vector protocols. Path-vector routing maintains knowledge about the path a packet follows in the network, namely

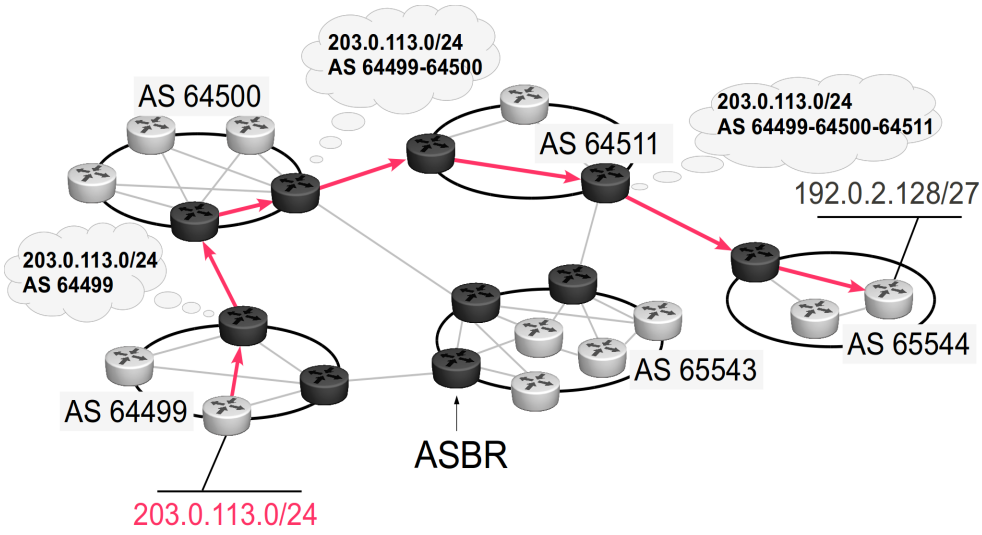


Figure 2.6 – BGP announces an AS path for reaching a destination prefix

the AS path as well as the additional attributes. An advantage of path-vector protocols is the ability to enforce policy-based routing, taking into account criteria beyond the mere destination address and route according to other attributes such as packet source, protocol type, etc. On the other hand, if the shortest path is not consistently preferred or if the distances are arbitrarily adjusted, the Bellman-Ford algorithm does not guarantee convergence. Section 2.3.3 discuses in more details the mechanisms used for loop avoidance.

2.3.1 External BGP and Internal BGP sessions

An ASBR learns a new BGP path towards a given network from its corresponding external peer in the neighboring AS. Usually, an AS contains more than one node, so the entering path has to be propagated to the other border routers within the AS. Since the IGP cannot handle the BGP attributes in the received message, the ASBR establishes an internal BGP connection to all other border routers. These internal connections carry information about external destinations, independently of the underlying IGP. To summarize, there are two operational modes allowing to communicate: external BGP (eBGP) between routers in different ASes and internal BGP (iBGP) between routers of the same AS.

A connection between a pair of routers is a BGP session[4] and it serves to exchange messages over a reliable transport protocol like TCP. The sessions established by iBGP and eBGP designate the BGP signaling graph. Conceptually, the relationship between BGP and the IGP can be represented as the superposition of two planes. As seen in fig. 2.7, the BGP sessions do not map precisely the IGP topology. A session between two adjacent routers is called a monohop session, whereas a multihop session is established

4. A session is a virtual circuit between two routers performing connection-oriented communication.

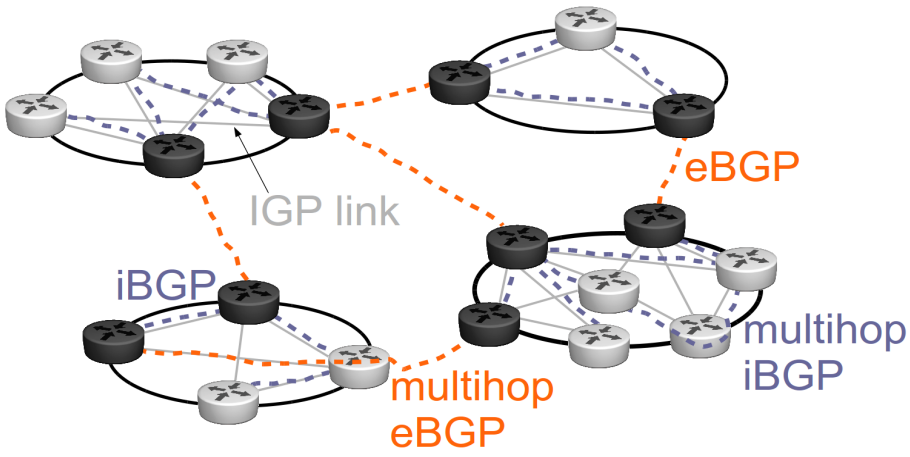


Figure 2.7 – eBGP and iBGP working together, with the iBGP topology on top of the IGP, ASBRs

between two distant routers that are not directly connected.

Two BGP speakers initiate a session over TCP and keep track of the connection through periodical keep-alive messages. Preserving a state allows the two end-points to be aware of spurious network events, such as link or node failure. When the routers cannot reach each other anymore, the session goes down. Once the connection retry timer expires, the session is no longer considered active, ensuing an invalidation of all the routes learned through the session.

2.3.2 Learning routes in BGP

The BGP sessions carry information about paths and each neighbor sends messages about the networks it can reach and the associated attributes. A node receives multiple paths to an IP prefix or Network Layer Reachability Information (NLRI). Roughly, if n is the number of prefixes advertised in the Internet, a BGP RIB will contain about $n * m$ routes in the worst case, where m is the number of neighbors sending their full BGP table.

Fig. 2.8 shows how routers keep a table called an incoming adjacent RIB (Adj-RIB-in) per neighbor. All the routes that pass the inbound filters are stored in the BGP Local RIB. The BGP decision process selects one best route from all the available RIB entries and installs the active route in the FIB. In this example, let us suppose the route advertised by peer 3 is selected as being the best one. The chosen route then goes through the outbound filtering layer and if it does not get discarded, the router advertises it to the BGP neighbors. Note that again, there is a separate Adj-RIB-out table holding the outgoing update messages for each BGP neighbor.

BGP operates in an incremental manner, sending update messages to neighbors when an event has occurred in the network. A router advertises an *announcement* message when a

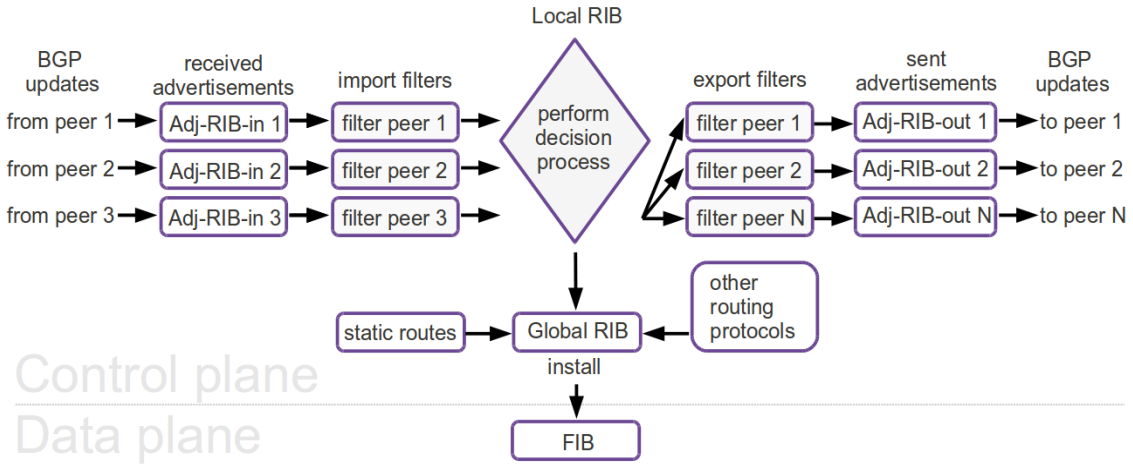


Figure 2.8 – Message processing in a BGP router for 3 concurrent advertisements

new route is available for reaching an NLRI. On the other hand, when the active route to a prefix is no longer available, the router sends a *withdrawal* message to the neighbors.

2.3.3 Best Path selection

When a router disposes of multiple concurrent routes to the same destination, the decision process selects a single best route. A ranking algorithm performs the selection of the best path to a given NLRI based on the attributes of the routes. In practice, attributes are parameters that can be tweaked in order to influence the decision process and enforce AS policies.

Table 2.1 depicts the main attributes of a BGP route and presents some associated values.

Table 2.1 – Main BGP attributes

path attribute	description	example
local_pref	well-known discretionary	100
multi_exit_disc	optional non-transitive	5
origin	well-known mandatory	IGP, EGP, Incomplete
AS path	well-known mandatory	{64497 65500 65508}
nexthop	well-known mandatory	203.0.113.97
community	optional transitive	no-export, no-advertise, internet, local-as, AS:value

The BGP decision process applies a set of rules when determining the best path. The ranking algorithm runs successively on all the concurrent paths entering the decision process and at each step eliminates the least convenient route. The candidate route that makes it all the way through the selection is the active route to be installed in the FIB.

Table 9.1 gives an overview of the BGP decision process. The decision process can be differently implemented by the router vendors and there are several proprietary attributes that can be added to the sequence. The next paragraphs explain in more details how the attributes influence the selection algorithm.

Table 2.2 – BGP decision process

#	preference	consideration
1	highest local_pref	economic relationship
2	shortest AS path	traffic engineering
3	IGP over EGP over Incomplete	
4	lowest multi_exit_disc	
5	lowest IGP metric to BGP egress	
6	lowest router ID	tie break

1. **Highest local preference:**
   The local_pref attribute is an integer value that expresses the preference of a network operator for a given next hop or even a neighbor AS. Note that local_pref is the first attribute taken into account in the decision process and it dictates the financial interest of the AS in question. The network administrator can choose to direct the traffic towards a client AS (equivalent to earning money) rather than a peer AS (no financial compensation, the traffic is mutually transfered) and avoid a provider AS (paying for the traffic sent). Other reasons might compel operators to prefer a certain AS path over others and enforce their choice by manipulating the local_pref.
2. **Shortest AS path:**
   The AS path is a list of AS numbers identifying the ASes the BGP announcement has traversed. The AS path attribute indicates the shortest path and gives an approximate metric in terms of inter-domain routing distance. It cannot reflect the precise cost of reaching a destination because in reality, ASes have different sizes and the number of hops within the AS is unknown to BGP.
   The messages that travel through the network and end up returning to the same AS are discarded, avoiding thus the count-to-infinity problem. Other than loop avoidance, the local domain can use the AS path to modify the length of the path. Adding its own AS number several times makes the path longer and diverts the traffic away. This procedure is called *AS path prepending*.
3. **External BGP over internal BGP:**

If at least one of the candidate routes was received via eBGP, remove from consideration all other routes received from iBGP. The objective is to make the traffic exit as soon as possible by sending it to an ASBR that will transfer it outside the domain. It prevents traffic from crossing the internal infrastructure and taking up resources. This principle is called *hot potato routing*.

4. **Lowest multi-exit discriminator:**
   The multi-exit discriminator (MED) is an integer value associated to each of the multiple links connected to the same neighbor AS. The lowest MED corresponds to the eBGP session that is most preferred by the local AS among the different links that are available.
   In Fig. 2.9 the MED attribute is intended for the neighbor AS 65499, informing it of the optimal entry point for traffic it sends to the local AS 65500. This mechanism allows a form of *cold potato routing* since the neighbor AS 65499 accepts to send its traffic through the next hop specified by AS 65500, at the expense of the packets staying longer in its own network. This is done in exchange for better routing once the packets reach the local AS 65500.

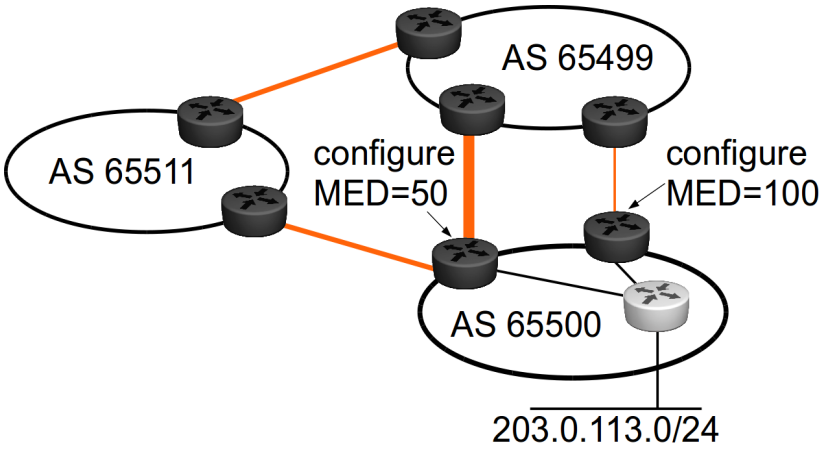


Figure 2.9 – The operator of AS 65500 sets a lower MED on a link with more capacity

MED is relevant to one neighbor AS and it usually does not make sense to compare values received from different ASes, each AS having its own scale. Since the MED is applied on a per neighbor basis, the BGP decision process breaks the rule of independent ranking of routes. Griffin and Wilfong [Griffin and Wilfong, 2002a] point out some unwanted behavior, such as persistent oscillations, due to the MED attribute.

5. **Lowest IGP metric to nexthop:**
   At this step of the decision process, all the remaining candidate routes are received from iBGP sessions. Another way to minimize the cost of traffic crossing the local AS is to send it to the closest exit point. Routers will prefer the nexthop with the lowest IGP metric, reducing the distance the traffic needs to travel inside the AS.

6. **Lowest router ID:**

   If after all the previous steps no best path has been selected, the BGP decision process applies a tie-break rule. The relevant attributes have been all compared and the remaining candidate routes are almost equivalent. To distinguish only one best route, the lowest router ID determines the winner of the selection algorithm.

2.3.4 BGP policies

BGP allows the administrative entity of a network to apply specific decisions regarding the paths to accept, select and propagate. The set of rules that define a customized treatment of paths are called the *routing policy* of an AS. There are two methods for enforcing the preference of a network for a given route, a set of routes, a neighbor AS or a category of ASes: the **attributes** in the BGP paths and the **filters** applied by the routers.

Through the BGP decision process, a network operator can choose to influence the selection of the route that it prefers by setting a high local_pref on an external link. It is also possible to modify the selection of the best route for the other ASes, either by conveying to neighbors the optimal entry point for the incoming traffic or by diverting the traffic away. The local AS expresses its preference for an entry point by tweaking the MED values and thus attracting the incoming traffic on a given link. On the other hand, the BGP routers can perform AS path prepending. Advertising a longer AS path makes the other ASes prefer another path that is shorter, thus deviating the traffic away from the local AS.

The *communities* attribute is another way of triggering policies that take effect based on associated community value. Although absent from the decision process, communities are used as tags that carry information about routes or sets of routes. There are some well-known communities such as *no_export* (do not advertise the routes outside of the local AS), *no_advertise* (do not advertise any route marked with this community), etc. The syntax *AS:value* allows a network operator to apply the policy identified by the value to the traffic exchanged with the designated AS.

The second method consists of applying import filters to the incoming BGP messages or export filters to the advertisements sent to the BGP peers. The filters are in fact instructions that perform a test followed by an action. The test evaluates the features of the routes such as specific values of attributes (e.g., verify that the AS path does not include a given AS number). Based on the result of the evaluation, the router decides to accept, filter or modify the route.

Routing policies support the diverse economic relationships between the ASes that build the Internet. Although sometimes pursuing contradictory goals, the network operators need to preserve a global service for all the users. The next section outlines some key features of the economical model established in the current Internet.

Valley-free model

Inter-domain routing is submitted to economical considerations rather than to technical reasons such as reaching a destination through the cheapest path, ignoring the shortest path. This financial aspect influences many of the routing decisions enforced by ASes.

A hierarchy settles the rules of global connectivity and complex AS relationships reign the Internet economy. The most common relationships are *customer-to-provider* and *peering*, although in special contexts new relationships appear, such as *paid-peering* or *sibling*.

A provider AS offers connectivity to a customer, allowing the customer to reach all the destinations it knows and accepting all traffic destined to the client. On the other hand, it will try to limit the traffic exchanged with its own provider, because it has to pay for transit. An AS can also have a peering agreement with another AS of equivalent size and they can mutually forward traffic for each other, without any associated cost.

In Fig. 2.10, the AS 65499 has customer-to-provider relationship with AS 65500, meaning that it has to pay for traffic exchanged on the link connecting them. A peering relationship allows it to send traffic for free to all the customers of AS 65498 since AS 65499 itself has to accept incoming traffic meant for its own customer ASes. Luckily, its two clients, AS 65505 and AS 65506, bring revenue each time traffic is forwarded to or from the end hosts.

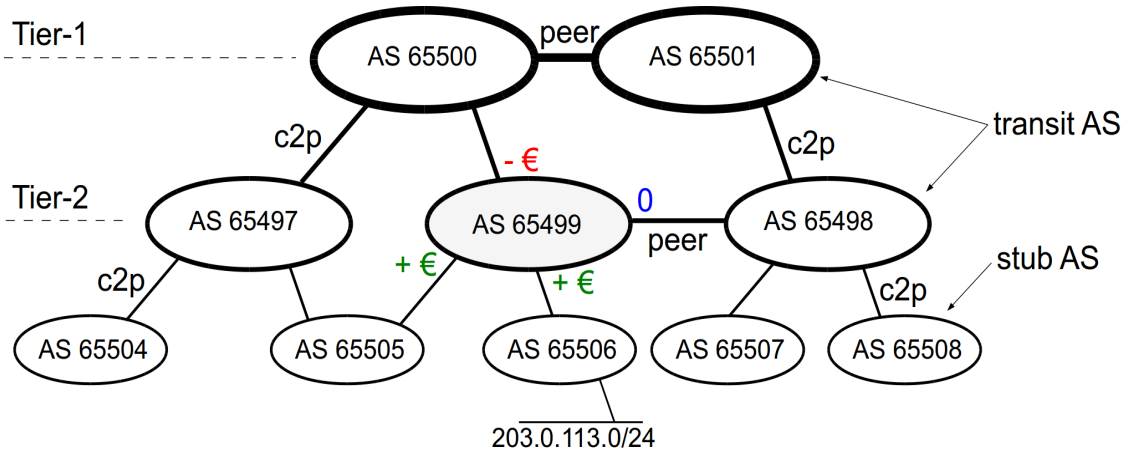


Figure 2.10 – Customer-to-provider and peer relationships within a hierarchy of tiers

Depending on the role they play, domains can be *stub ASes* or *transit ASes*. As their name indicates, stub ASes are the terminal domains that federate the end users, sending and receiving packets destined to the hosts directly connected. A stub AS is said to be *multi-homed* if it is connected to two or more providers. Transit ASes are the bigger domains that sell a transit service to other ISPs wanting to send or receive traffic from remote destinations in the Internet.

According to the Internet's economical principles, traffic follows a hierarchy, going from less important ISPs (called Tier-3, Tier-2) to bigger ASes (called Tier-1). The ASes at the

very top of this hierarchy do not have any providers, meaning that they are able to route all the destinations in the Internet either through their clients or through their peers. The BGP RIB of routers in Tier-1 ASes contains no default route and they are said to perform *full routing*. The clique made of the Tier-1 ASes represents the *default-free zone* (DFZ).

From the business relationships between ASes, it is possible to deduce a pattern in the routing behavior: an AS does not transit traffic for two of its providers because this takes up network resources without bringing any financial compensation. The same reasoning applies for transit between two of its peering ASes, there is no point in forwarding traffic for free. In conclusion, an AS provides transit between two networks only if at least one of them is a client [Gao, 2001]. These business preferences establish the basis of what is known as *the valley-free model*.

Fig. 2.11 shows an example of a path that violates the propagation model because it is valley-shaped. It's an economically invalid path because AS 65505 loses money both for the incoming and the outgoing traffic, none of which is meant for its own hosts. If AS 65505 receives traffic from is provider AS 65497, it has to pay for using the link. If the traffic accepted on the link is not for its end hosts, but is in fact directed to its provider AS 65499, the client has to pay again.

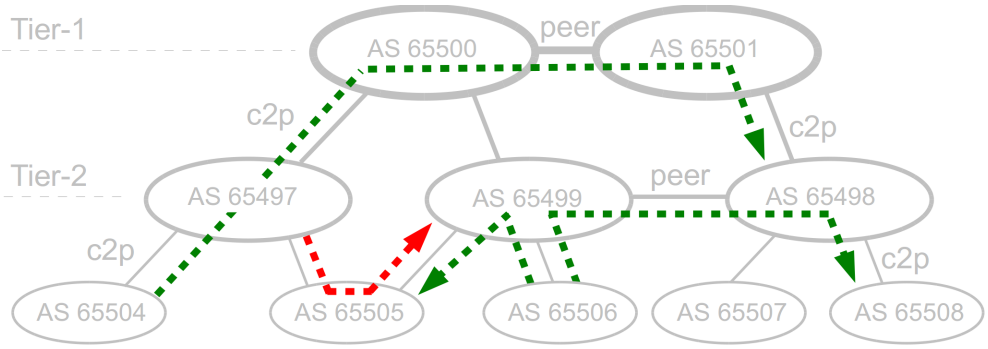


Figure 2.11 – The economically valid and invalid paths in the Internet

The Internet's value is determined by the connectivity. A group of users who can reach only a restricted subset of the Internet population, or even worse, is separated from the Internet, does not enjoy the service they have paid for. As a consequence, the operators are interested in maximizing their revenues while at the same time cooperating to keep the Internet globally stable and connected. The Gao-Rexford conditions [Gao and Rexford, 2000] stipulate that routing oscillations can be avoided if network operators follow the valley-free model when designing their policies.

The ASes that need to interconnect in order to exchange traffic build up a meshed Internet graph. As a consequence, management becomes complex if every AS has to have a physical link going out to each of its customers, peers and providers. The next section shows how Tiers manage to link up in different parts of the world.

2.3.5 Route Servers

Dedicated platforms allow different ASes to interconnect in Internet Exchange Points (IXP). IXPs deploy important hardware infrastructures allowing network operators to interconnect to a plethora of other networks. There are around 120 IXPs in Europe and the most important ones have joined in the European Internet Exchange Association (Euro-IX) [Euro-IX, 2011]. Some of the available exchange points are AMS-IX (located in Amsterdam), LINX (in London), InterLAN and RoNIX (in Bucharest), Equinix etc...

What would have been a mesh of multiple eBGP sessions towards different neighbors becomes a single link to a Route Server [Jasinska *et al.*, 2011] located at an exchange point. Route Servers (RSes) are network equipments that do not forward actual traffic, but determine the paths to redistribute between the connected eBGP peers.

Things are rather simple if all the ASes in the IX have multilateral peering agreements. The implementation of the route server will federate all the information received from the client routers and compute the best paths then advertise the results to each of the participating peers.

What happens if the best route to a prefix is advertised by AS 65507 and AS 65500 does not want to use any of the routes advertised by this AS? Does it mean that AS 65500 will not be able to reach the given destination, even if a second best route is available in the RIB? Suppose AS 65501 is a provider of AS 65499 and it wants to apply a specific route-map [5]... How can the route server handle these requirements?

ASes have different policies and some of the connected clients want to receive preferential update messages from a given AS and ignore routes announced by other ASes. Route servers allow different views of the Internet through multiple RIBs (e.g., one RIB per client). The RS computes the best path to advertise to an AS based on the corresponding RIB. This allows each AS to receive routes that are compliant to its own routing policy.

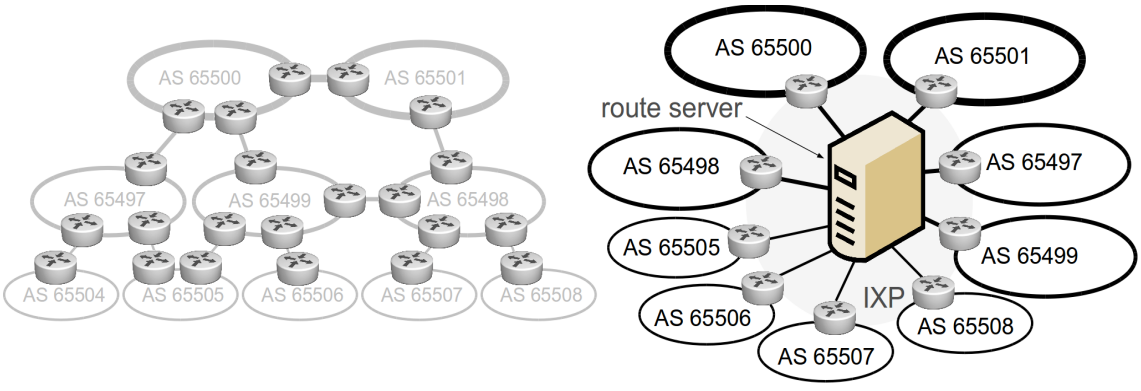


Figure 2.12 – The AS partial mesh versus route server interconnection in IXP

5. A route-map allows a network operator to modify attributes of a route or filter routes based on attributes; it is made of clauses such as "permit" or "deny" followed by statements such as "set" or "match" and associated access lists.

2.3.6 iBGP architectures

We have seen how ASes interconnect to each other, let us now explore the network architectures used inside an AS. The categoric distinction between eBGP and iBGP makes it possible for an ISP to deploy a particular iBGP architecture without any impact on the neighboring ASes. Any of the following setups can be used, with route reflection and confederations that can be simultaneously combined.

Full Mesh

Historically, in the early days of BGP, the number of gateways or routers of an AS was smaller than what we are faced with today. The design constraints of the protocol specify that each iBGP peer has to establish sessions with all the other BGP speakers inside the domain.

The full mesh concept enjoys some appealing features such as maximal route diversity, all routers receive their peers' best route, which can turn out to be quite useful in case a backup route is needed. The management is quite simple, debugging the routing behavior is easy since it is highly predictable. Peers advertise only the paths received on eBGP sessions and if the best route is chosen based on the local_pref, AS path or MED attributes, it should be the same one for the entire AS.

Among other advantages, the full mesh offers fast reconvergence and robustness in case of a network event. Indeed, a full mesh architecture reacts immediately to changes in the network topology and BGP messages are rapidly propagated, all iBGP peers being one hop away.

Although a good solution for ASes with a few BGP routers, the full mesh is subject to some inherent drawbacks. Limitations of the full mesh include memory requirements and performance of the network equipments: each router needs to keep as many Adj-RIB-ins as there are neighbors. Other than that, the memory is wasted since from the total routes received, just few of them will be active. The iBGP peers receive routes from eBGP sessions and also from all the internal routers. If the external routes are preferred, the routers receive, store and handle n copies of the routes to the known destinations and actually use only a fraction.

We've seen that the network quickly reacts to changes, but he reverse of the medal is that the full mesh is verbose. Many messages take up network resources, announcing changes in topologies even to the peers that are not directly concerned by the modifications.

While a full mesh can be reasonably applied in small architectures, this configuration can become a scalability issue if the number of participants increases. The total connections vary with the square number of BGP speakers involved: for a network of n routers, the operator has to set up $n * (n - 1)/2$ iBGP sessions. To avoid the processing overhead induced by the full mesh, the networking community has introduced two alternatives: confederations and route reflectors (RRs).

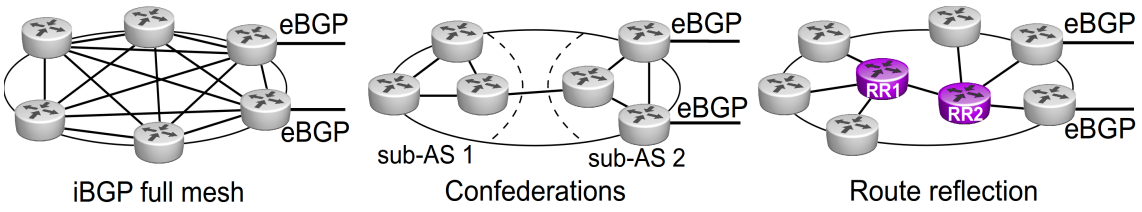


Figure 2.13 – Examples of a full mesh of iBGP sessions, confederations, route reflection

Confederations

Confederations [Rekhter and Li, 1995 ; Traina *et al.*, 2001] are sub-ASes meant to divide a large network into areas of a more manageable size. A network can be split into several smaller routing domains that are seamless to ASes outside the confederation. In Fig. 2.13, sub-AS 1 and sub-AS 2 appear as single AS to the neighbors. The sub-ASes are identified through *private AS numbers* that are stripped out of the AS path at the edge of the confederation. These AS numbers do not play a role in the BGP decision process and are used solely for loop avoidance.

The placement of the ASBRs is important since the two sub-ASes exchange all the traffic through the defined links. Another aspect to keep in mind is the lack of flexibility, since the flows are forced to exit the confederation through next hops that might not be the same as the shortest path out of the sub-AS. The packets remain in the network for a longer time, leading to sub-optimal routing. There is also a penalty on convergence time since the messages need to go through more ASes and be processed on more routers.

Each member of a confederation can use a different IGP and this can trigger some unwanted interactions between different metric spaces on the border routers, as pointed out in [Le *et al.*, 2010]. Another limitation is the fact that handling sub-divisions of a network increases the complexity of management and debugging.

BGP confederations can be used as a preliminary step for integrating heterogeneous IGPs into one network. With confederations, it is possible to take into account geographical topology constraints, such as peering agreements that try to avoid carrying traffic over transoceanic links. Another advantage is the the opportunity to enforce BGP policies based on physical or political boundaries.

Route Reflection

Route reflection [Bates *et al.*, 2000] is a method that renews route redistribution in an AS, by using a hierarchical topology instead of the flat full mesh. Route Reflectors (RRs) are special routers that federate the learned routes and propagate the BGP updates to other routers called *clients*. Clients subscribe to the BGP announcements advertised by one route reflector or sometimes more, for redundancy purposes.

The changes brought in the iBGP architecture currently define two new types of BGP

peers that a route reflector can have: *client peers* and *non-client peers*. A route reflector (or a set of route reflectors) and the associated clients form a route reflection *cluster* as seen in Fig. 2.14. Similar to the AS path attribute, a *cluster-list* holds information about the clusters that a message has traversed. This mechanism is appropriate for a form of path-vector routing within the AS.

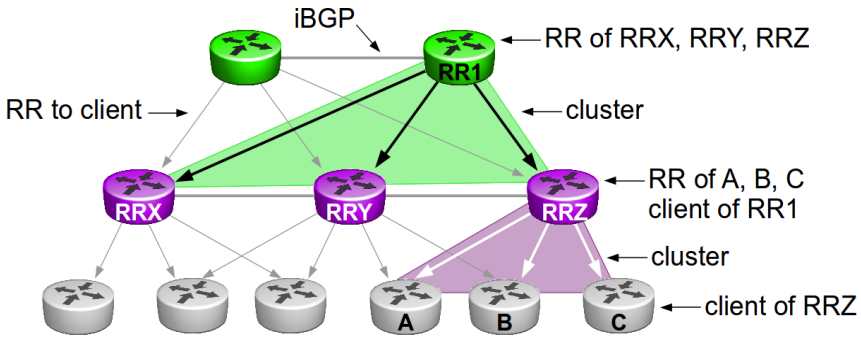


Figure 2.14 – Examples of route reflection

Breaking the classical propagation rule of the full mesh, route reflection provides the means to reduce the number of sessions in an AS, thus offering more scalability.

The rules of route advertisement are relaxed, allowing BGP speakers to send iBGP learned routes to other iBGP peers. Loop avoidance is done with cluster lists, but how can route reflectors avoid creating reachability problems? In the hierarchical organization, RRs propagate BGP updates according to a new set of rules that make route reflection work properly. Table 2.3 specifies the propagation patterns applied by route reflectors:

Table 2.3 – Route reflection rules

received from	reflect to
a client	all iBGP neighbors
a non-client	all clients

Route reflection is commonly used in large networks because of scalability considerations. Some operators have managed to install intricate RR hierarchies, with several levels of route reflection where an RR becomes a client of another RR of higher level.

Just as any other router in the network running a BGP decision process, an RR will select a single best route and then decide to propagate it. When joining route reflection to the BGP selection algorithm, the diversity of paths is greatly reduced in the core network. These aspects are later investigated in Chapter 3.

Route reflection can be used jointly with confederations, depending on the goals of the network operator. BGP architectures respond to specific needs and the design must take

into account particular implications. Table 2.4 points out the characteristics of the three architectures in terms of number of sessions and the number of paths received. Refer to the Juniper application note [Juniper Networks, 2002] for a more thorough comparison between route reflectors and confederations.

Table 2.4 – Number of sessions and the maximum number of paths in the Adj-RIB-Ins

	# of sessions	maximum paths in Adj-RIB-Ins
full mesh	$\#routers - 1$	$RIB_{size} * (n - 1)$
route reflector	$\#RRs - 1 + \#clients$	$RIB_{size} * (\#RRs - 1 + \#clients)$
RR client	usually 2	$RIB_{size} * 2$

2.4 Conclusion

This chapter has confronted the issues of end-to-end communication in the Internet, passing through routing protocols, router structure and AS interconnection with BGP. The reader is now familiar with the path attributes compared in the BGP decision process and routing policies. Section 2.3.4 laid out some of the economical reasons that shape inter-domain routing into the valley-free model.

Finally, the last section contained an overview of the three main iBGP architectures currently used in networks: the full mesh, confederations and route reflection. Recall that full mesh is adequate in small size ASes, whereas confederations and route reflection offer a tradeoff between scalability and complexity. Route reflection is most largely deployed and most popular solution in Tier-1 and other large networks. The rest of this dissertation concentrates on this specific architecture, unless otherwise stated. The following chapter elaborates a taxonomy of the problems that can be encountered in the current BGP, as well as some proposed solutions.

Chapter 3

Flaws and fixes in BGP routing

This chapter presents some of the shortcomings that network operators might encounter while managing a BGP network. The issues that "plague" BGP, and inherently the Internet, are widely studied in the literature and some more or less adequate solutions have been proposed. Section 3.1 displays an overview of the causes that have led to the current situation, with an emphasis on the evolution of BGP. The following subsections focus on aspects such as scalability, correctness, path visibility in route reflection architectures and path diversity at the router level.

A taxonomy of existing solutions is delivered in Section 3.3 and the concluding paragraphs make a case for the main research work presented in the body of this dissertation.

3.1 Current BGP Plagues

As summarized by Jennifer Rexford [Rexford, 2011], one of the main causes that have led to the current status of BGP is the absence of underlying models, with a protocol designed without having in mind the decision process or a specific policy language. BGP models such as Stable Path Problem (SPP) and Stable Path Vector Protocol (SPVP) or policy languages like Routing Policy Specification Language (RPSL) came along much later.

BGP has been around for a long time, with version BGP-4 operational and in use since 1994. Over the years, BGP has endured many additions, including new route attributes, decision process steps or even router structure (e.g., the introduction of the Adj-RIB-Out in order to reduce BGP churn). The incremental evolution of BGP has not allowed a rethinking of the general design, leading to a rather complex and (still) mysterious system. The plethora of standards and patches related to BGP can be monitored at [Inter-Domain Routing Workgroup, 2011].

The rapid expansion of the Internet is salient in the datasets that confirm the sustained increase in the number of users during the past 20 years. This growth has resulted also in many more ASes and a more intricate topology. The opportunities that the Internet

provides have not been overlooked by the business industry and soon, competition and antagonistic interests have populated the relationships between the operators. Complicated routing policies and security concerns came along as a natural consequence.

During the last decade, the research community has reached a better understanding of the side-effects appearing from protocol interactions or under specific conditions. The following sections focus on problems related to intra-domain architectures, bringing forward a review of these topics and the related solutions.

3.1.1 Scalability

The most important dimension of scalability inside an AS is the size of the routing table.

The Size of the Routing Table

The numbers presented in Section 2.1 demonstrate that the Internet has come a long way in terms of number of users. The scaling limitations have been tackled with a *divide et impera* approach, first with the use of ASes and later with Classless Inter-Domain Routing [Fuller and Li, 2006] that allowed better prefix management.

More people talking to each other means bigger networks, translating to more prefixes and at the same time more ASes in the Internet. Geoff Huston's website, `potaroo.net` [Huston, 2011b] is a valuable source of information when it comes to monitoring the general trends of the expansion. Fig. 3.1 reproduces a graph depicting the evolution of the number of advertised ASes, now going beyond the 40,000 threshold.

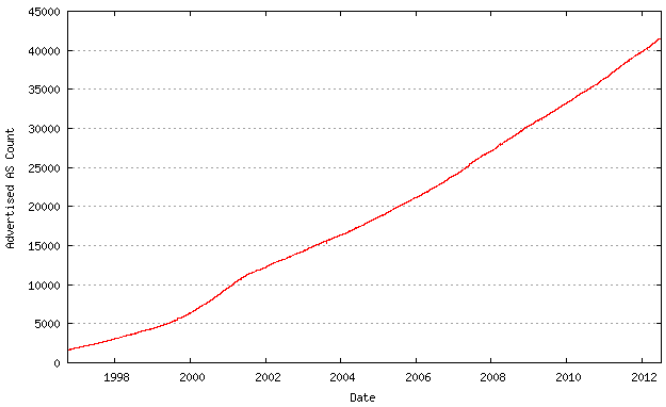


Figure 3.1 – Total of advertised ASes. [potaroo.net]

There are as much as 400,000 prefixes in the current Internet DFZ, compared to the figures from eleven years ago, when 100,000 prefixes covered all destinations. Fig. 3.2 is a popular graph that illustrates the evolution of the set of active BGP prefixes, measured from several vantage points.

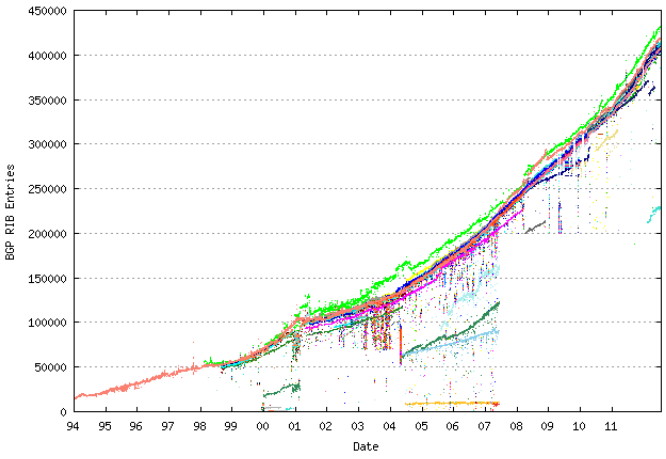


Figure 3.2 – The number of BGP active prefixes (FIB entries). [potaroo.net]

Some of the causes leading to the BGP table explosion are related to the natural growth of the number of users and machines in general that are on the Internet, the use of provider-independent prefixes, multi-homed ASes and prefix de-aggregation for traffic engineering or security purposes. Similar to the Internet scenario, the origins of route explosion in the case of BGP/MPLS IP VPNs are yet to be fully understood, but a preliminary study exists in [Ben Houidi *et al.*, 2007]. An extra constraint on VPN routes is the fact that they cannot be aggregated because each route identifies a given client.

Scalability in the Internet is one of the struggles of the research community and the standardisation bodies. There has been tremendous effort to insure that the Internet could keep up the pace with the increasing demands. The current trend of the routing table indicates continuous growth of the Internet and we expect future evolution to be similar, especially during the transition to the apparently inexhaustible IPv6 space and to new features such as *add-paths* [Walton *et al.*, 2011]. Several solutions exist for achieving scalability, the following section shows some mechanisms used when dealing with the routing table size.

Achieving scalability

Nowadays, scalability can be partially achieved thanks to two important features inherent to the Internet Protocol: **hop by hop** IP routing and IP **longest match** forwarding. These properties allow network operators to perform filtering and aggregation.

Some of the messages posted on the NANOG [1] mailing list are quite straight forward, compelling the engineers and network administrators to take action: “Filter, people, filter!”. Some of the means to reduce the size of the BGP routing table is to automatically discard prefixes that are longer than a /24, meaning that networks that advertise more specific

1. North American Network Operators’ Group.

/25s, /26s and so on are no longer accepted. Inbound filters allow an implementation of this mechanism and help save memory on routers by filtering the long prefixes received on the eBGP sessions. This scheme is limited in application because it deteriorates reachability for perfectly legitimate prefixes when there is no shorter prefix that contains the filtered destinations.

Network operators may also decide to filter prefixes based on other criteria. One option is to drop all invalid prefixes, known as *bogon* prefixes. Bogon prefixes are not allocated [2] by the registries, so they do not correspond to any authorized networks. Security considerations might require the filtering of *martian* prefixes, i.e. private and reserved pools of addresses. All these filtering actions aim to reduce the number of prefixes stored in the RIB (Adj-RIB-Ins and Adj-RIB-Outs) and optimize the memory consumption.

A less obvious way to reduce the size of the routing table is performed within the routers: equipment vendors optimize the code in order to reduce the size of the memory required to store the routes. However, this optimization does not entirely fix the problem since the increasing connectivity and the tweaking of routes for traffic engineering keep inflating the size of the BGP routing table.

Aggregation is widely deployed, it has good properties that enable the hiding of "remote" details. In practice, there is no real aggregation in the core network, but only in the source ASes. It is not a perfect solution because there is no (financial) incentive for the AS performing the aggregation and it would require a coordination between the real Internet topology and the address allocation policy, i.e. same hierarchies. Another shortcoming is the recent IPv4 public address scarcity that leads to smaller address blocks being used, hence de-aggregation.

Other projects advocate the idea of downsizing the routing table: ViAggre (Virtual Aggregation) is a configuration-only method for shrinking the size of the forwarding table in the Internet default-free zone. It proposes a "dirty slate" technique for distributing routing within an ISP network so that routers maintain only a part of the global routing table. A level of indirection is added for when there is no match in the local RIB. Routers have to forward the packets to another router that is aware of the path to the incoming prefix. One of the negative impacts of ViAggre [Ballani *et al.*, 2008] is a stretch imposed on traffic, diverting it from the native shortest path. Another inconvenient is the difficulty of the configuration. This same approach is advanced in [Francis *et al.*, 2011] and X. Zhang et al. elaborate similar work in [Zhang *et al.*, 2006], but Core-Router Integrated Overlay (CRIO) seems to bring more benefit to VPN routing.

The need to overcome scalability issues in BGP/MPLS VPN networks has become a reality in large provider networks as presented in [Ben Houidi, 2010]. One of the alternatives is a rethinking of the entire design for VPN architectures and the questionning of whether BGP itself is appropriate for the desired goal [Ben Houidi and Meulle, 2010]. The second option is to adapt the current network construction so that it can face the evolution in

2. The Internet Assigned Numbers Authority (IANA) has exhausted its IPv4 pool of addresses, but the Regional Internet Registries (RIRs) still have some remaining prefixes that have not been yet delegated to the Local Internet Registries (LIRs).

the number of VPN sites, sessions and the increasing table size. Operators will opt for the latter approach, as proves the VPN partitioning architecture from [Cazaux *et al.*, 2009].

In order for the BGP control plane to evolve, new nodes can be added, or more powerful equipments can replace older ones. The capacity of an RR is typically a concern for an ISP and a good method to differentiate the router vendors. There is a constant race for more RR capacity in order to accommodate the growth in the number of prefixes, in the number of sessions and in the number of routing messages.

M. Dobrescu *et al.* revisit the problem of router scalability in [Dobrescu *et al.*, 2009] and propose a system called RouteBricks. The presented software router architecture parallelizes the functionality of a router across multiple servers and across the cores within a single server. The prototype router built from commodity hardware achieves performance comparable to specialized routers from the low-end range.

Since vertical scaling is limited by the equipment capacity, it may be easier to favor other schemes, such as introducing another layer of route reflection. In [Vissicchio, 2012], the authors present an algorithm that allows a BGP architecture to evolve while respecting valid signaling paths. The migration to a final iBGP topology guarantees a safe passage, with no unwanted effects such as transient black holes, forwarding loops or suboptimal routing and with minimal interference on the eBGP sessions. On the other hand, the limitation of this proposal is related to the lack of flexibility: it requires an initial fm-optimal topology and it also needs all intermediate topologies to be fm-optimal. In addition, some spurious sessions appear, traffic shifts among egress points and there might be unlikely eBGP updates.

3.1.2 Correctness

A network topology is said to be correct if there are no anomalies that can cause the protocol to diverge or to have other unexpected behavior such as undeterministic outcomes or deflections in the path of packets. In [Griffin and Wilfong, 2002b], the authors prove that it is NP-hard to determine whether an iBGP configuration is correct, but manage however to provide a set of "simple sufficient conditions on network configuration that guarantee correctness".

Correctness issues are a consequence of information masking in route reflection topologies. The lack of visibility of all the routes available in the AS causes some wrong or unexpected decisions from the part of routers. Although *hot potato* routing is often the desired goal, the outcome deviates from the intended routing policy.

[Bornhauser *et al.*, 2010] cites a list of the possible causes leading to anomalies in iBGP. The work of Griffin, Sobrinho, Wilfong and others reveals a large panel of issues due to iBGP construction and analyzes the undesired effects of route reflection topologies. This section presents some of the problems that arise when a route reflector topology is not well chosen, thus leading to possible side-effects such as:

- suboptimal routing

- non-deterministic convergence
- deflections possibly degenerating into forwarding loops
- routing oscillations resulting in instability

Suboptimal routing

Suboptimal routing happens when traffic follows a longer route than the shortest path, taking up more resources within the network. This effect appears when a route reflector picks a route that it considers best from its point of view, but who can turn out to be different from what the client router would have chosen when confronted to the same set of candidate routes. Remember that an RR makes a choice based on its own position in the IGP graph, disregarding the IGP metrics between the AS exit point and the client router, that will in fact be using the route.

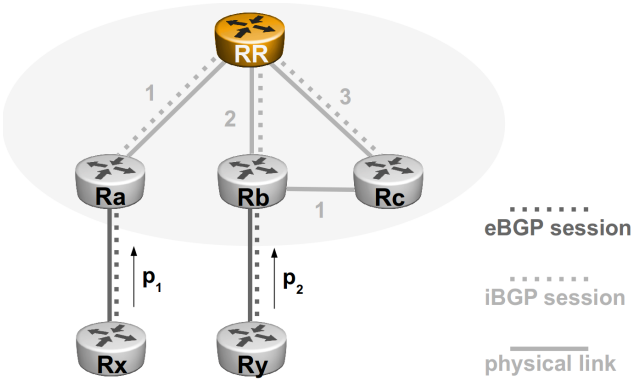


Figure 3.3 – Suboptimal routing: Rc does not learn about the path through Rb, although shorter from the standpoint of the IGP metric.

In the example depicted by Fig. 3.3, the route reflector RR receives two almost equivalent candidate routes to the same destination and prefers the route through Ra because of its smaller IGP metric. All the clients receive the route through Ra which leads to suboptimal routing for Rc. Indeed, Rc is not aware of the other route through Rb which would have been preferable from an IGP standpoint.

Non-deterministic behavior

Ideally, the routing decisions should not be influenced by the arrival order of the announcement messages. The situations depicted by Fig. 3.4 show that BGP is not guaranteed to converge to a unique, stable solution. In these cases, we are dealing with a bi-stable solution where the network converges to a state where the first announcement is preferred.

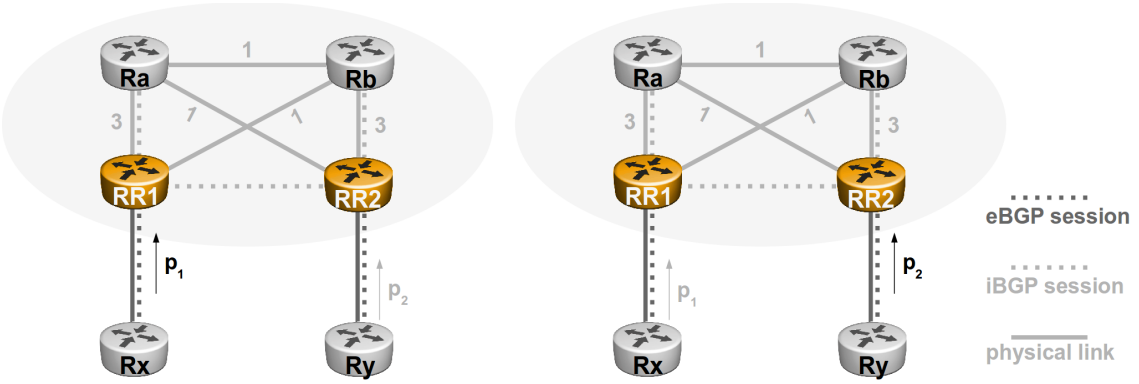


Figure 3.4 – Nondeterministic routing: depending on the moment of arrival of the update messages, the network can converge on two different states.

Deflections and forwarding loops

Deflections occur when the path of a packet is changed by one of the routers along the way, diverting it thus from the initial exit point. A simple deflection will cause the traffic to exit the AS through a different nexthop router. When multiple deflections appear, a loop can be created and the packets trapped in the loop will never reach their destination.

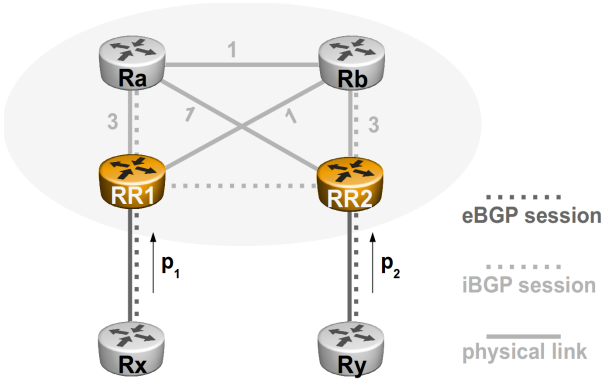


Figure 3.5 – Deflective routing: the traffic gets caught up in a loop between Ra and Rb.

Such is the case for the setup in Fig. 3.5 where eBGP peers Rx and Ry both advertise concurrent but almost equivalent routes to the same destination prefix. Both RR1 and RR2 choose the external routes over any other route advertised by iBGP, so RR1 picks the route advertised by Rx and announces it to Ra, its client; RR2 selects the route through Ry and sends it in its turn to its client Rb. Each of the client routers knows only one exit route which is not the optimal one from the IGP point of view (Ra cannot exit through RR2 and Rb is not aware of the route through RR1).

For forwarding the traffic, the client Ra uses RR1 as a next-hop, but the shortest IGP path goes through Rb. The traffic reaches Rb whose exit point is in fact RR2, so Rb decides

to reach RR2 using the path with the lowest IGP cost, which happens to be through Ra. The situation repeats and the traffic is trapped in a loop between Ra and Rb.

This phenomenon is related to the ability of the routers along the path to question the decision of the upstream router. Multi-protocol Label Switching (MPLS) gets rid of this problem by setting up a tunnel between the source and the destination. Routing no longer dynamically controls where the packets go, as this is decided by a set of labels that determine the path in the network.

Routing oscillations due to the IGP metrics

Fig. 3.6 depicts a classical example of a "no solution" topology that permanently oscillates. This particular scenario is reproduced from [Griffin and Wilfong, 2002b].

Consider a route to a given prefix P advertised by Rx, Ry and Rz. The border routers Ra, Rb and Rc will prefer their direct eBGP path. Due to the specific topology and the IGP metrics on the links, the route reflectors RR1, RR2 and RR3 will never reach an agreement about the best path to P.

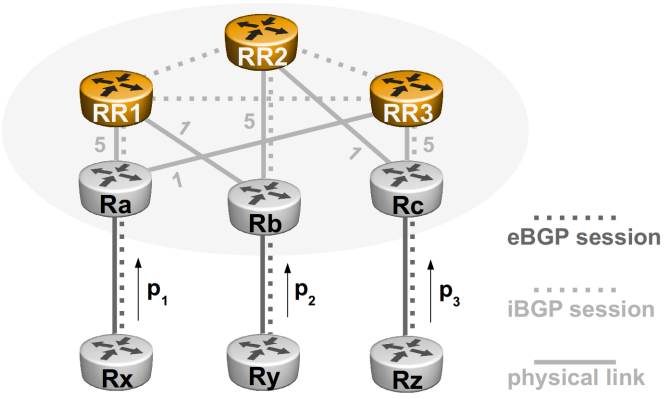


Figure 3.6 – Network topology with routing oscillations due to the IGP metrics

Indeed, each of the RRs has one client, but the IGP configuration makes RR1 prefer RR2's client, Rb; at the same time RR2 prefers RR3's client, Rc; and RR3 prefers the client of RR1, Ra due to lower metrics. Initially, all RRs know the route advertised by their own client and they advertise it to their peers. But as soon as they each receive the routes from their peers, they select as best path the one advertised by the peer and hence they each withdraw their own path. Simultaneously, the neighbor does the same and withdraws in its turn the path it had advertised, so the current best path becomes unavailable. Every RR switches back to its own client route and the situation continues indefinitely.

The routing oscillation depicted in Fig. 3.6 can be observed in practice with the dVirt [Oprescu *et al.*, 2011a] simulation tool presented later in more detail in Chapter 6. An analysis of the messages exchanged between the routers shows that RR3 keeps updating and withdrawing its advertised routes as seen in Fig. 3.7. On the y axis, the BGP message

type can be observed: for any given prefix, it can be either *withdraw* or *announce* and the x axis represents the time. It can be noticed that for a prefix, the messages oscillate continuously between the two states. The same behavior can be observed for the other RRs in the topology.

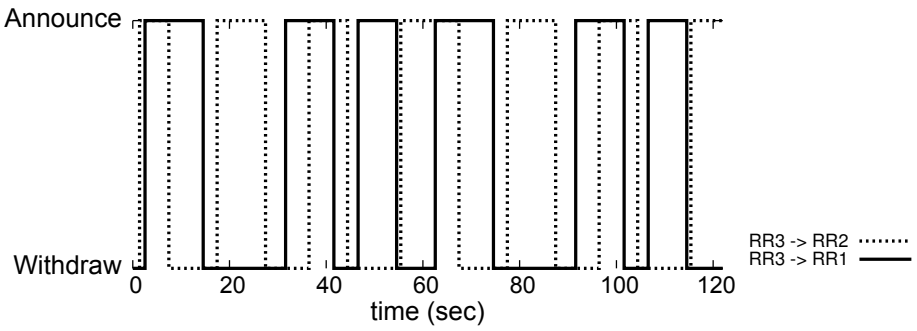


Figure 3.7 – BGP messages oscillating between announcement and withdrawal of routes

Recent standardization efforts [Raszuk *et al.*, 2011] acknowledge the fact that RR deployment thwarts the ability to achieve hot potato routing. The IETF proposes two solutions that support route reflection based on the client's position in the AS: "best path selection for BGP hot potato routing from customized IGP network position" and the second one is "angular distance approximation for BGP warm potato routing".

Optimal route reflection is a step forward, improving the propagation of routes according to a more suited view of the network. Now that the route reflectors will be able to take into account the client's place in the topology, one can expect more accurate routing decisions.

MED-induced routing oscillations

Routing oscillations provoked by the MED attribute are presented in [McPherson *et al.*, 2002]. This type of oscillation is caused by the fact that MED intervenes in the BGP decision process and violates the simple ranking of routes. When MED is used, a route's rank can vary according to the presence or absence of other routes to this same destination.

The system FRIED-POTATO in Fig. 3.8 is taken from [Griffin and Wilfong, 2002a]. Suppose the router R_a receives two paths P_1 and P_2 to reach prefix P. Path P_2 from AS 65499 has a MED attribute set to 1 and a lower metric [3], making it the winner for R_a.

As seen in the figure, R_b receives in its turn another path, P_3, from the same AS 65499. The RR receives the routes from the clients and compares them: P_3 is selected because it has a lower MED than P_2. After the decision process, the RR reflects its best route, P_3, to its client R_a. When finding out about P_3, R_a stops advertising P_2 because of the higher MED. However, R_a does not select P_3 as its best, but changes to P_1 instead. At this point, the RR is aware of both P_1 and P_3 and thus switches to P_1 because of its lower

3. The metric used here reflects the preference in case of tie-breaking rules and can be different from IGP metrics that are not systematically used on the eBGP links.

IGP metric. RR withdraws the advertisement of P_3 to R_a; now that R_a no longer has P_3, it goes back to using P_2, since there is no competing route with a lower MED anymore. The system has reached the initial situation and the whole process happens again.

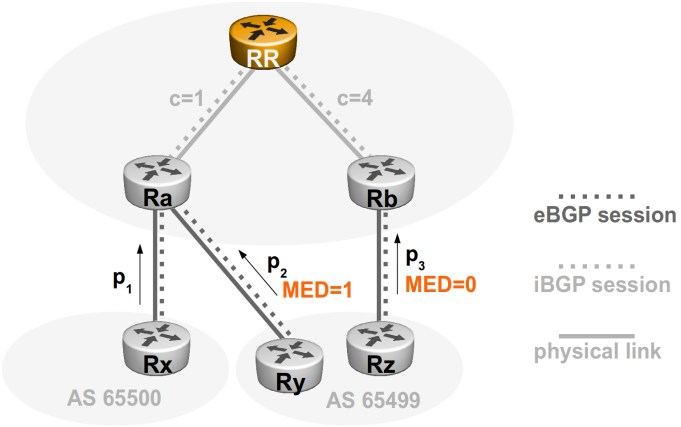


Figure 3.8 – Routing oscillation due to the MED attribute

The oscillations due to the MED attribute can be fixed by adding a new type of session called a lightweight BGP session (liBGP) between the ASBRs, as suggested by the authors in [Van den Schrieck *et al.*, 2006]. A theoretical approach allows Flavel and Roughan [Flavel and Roughan, 2009] to construct routing algebras per neighboring AS and their idea is to propagate one route per neighbor, thus avoiding undesired MED effects.

Another way to avoid MED oscillations is to set one of the following options on the routers that are potential victims of this kind of anomaly:

1. *always-compare-med:* compare MEDs even for candidate routes coming from different neighbor ASes.
2. *set-deterministic-med:* for a given prefix, choose the best route per neighboring AS. Then compare each of these best routes to select the final best route to a given destination.

Guaranteeing correctness

As pointed out in [Rawat and Shayman, 2006], there are two straightforward approaches for tackling the routing issues due to MED or iBGP topology. The first one consists of modifying the protocol, which can be a challenging task because it requires a wide deployment, and the second one relies on an intelligent manner of configuring the routers in the AS such that all anomalies are absent.

Foreseeing all the use cases can be quite complex. It is hard to guarantee a good behavior in RR networks, but they should enforce general architecture rules. [Feamster *et al.*, 2004b] propose a model for valid signaling paths in iBGP topologies. The propagation model can be summarized by the regular expression $(up) * (over)?(down)*$. This means each path

contains several steps during which it is climbing up the RR hierarchy, one single hop across followed by zero or more *down* edges towards the clients.

Further theoretical validation has been proposed in [Buob *et al.*, 2007] where a method allows for checking that a topology is full mesh optimal (fm-optimal), meaning that the RR hierarchy produces the same output as a full mesh configuration. The authors suggest later in [Buob *et al.*, 2008] a way to design optimal route reflection topologies that are compliant with the fm-optimality criteria. Finally, [Buob, 2008] gives details about iBGPv2, a modified and improved version of iBGP where BGP routers interconnect according to the IGP topology. This method is based on a simple propagation rule: a BGP peer announces a route to its neighbor only if the latter is interested in discovering this new information. A computation of Dijkstra's algorithm for a limited number of neighbors allows the router to decide whether to forward the BGP announcement or not.

BGP Skeleton [Sarakbi and Maag, 2010] presents itself as an alternative to route reflection while at the same time overcoming the inherent routing anomalies. It proposes to use a BGP signaling graph that is a subgraph of the physical graph. Using this method, the concept of clusters is eliminated and all Skeleton nodes have the ability of reflecting routes, but the solution does not address any scalability issues other than the number of sessions handled by the BGP routers.

A similar objective is achieved in [Vutukuru *et al.*, 2006] where the authors test an algorithm for determining the suitable meshing for the iBGP network. Their BGPSep implementation allows for correctness by conveniently splitting the network graph and assigning the route reflector hierarchy. Note that it is a constructive algorithm and again, the only issues that are tackled concern correctness guarantees and the reduced number of iBGP sessions comes as a beneficial side-effect.

[Griffin and Sobrinho, 2005] defines the Routing Algebra Meta-Language (RAML) that allows a network administrator to build a large family of routing algebras and from which correctness conditions can be derived for the chosen routing mechanisms. This work is a part of the larger Metarouting project [Metarouting, 2011] whose goals are to define a high-level declarative language called a *routing metalanguage*, an abstract formalism based on a theoretical foundation. The routing metalanguage can be used for specifying the policy components of routing protocols captured by the associated algebras. The authors "envision a world in which routers do not implement any routing protocols but rather come with a **routing metalanguage compiler** [sic]".

The solutions presented above rely on a formal validation of topologies and are usually funded on a fixed view of the network graph. When resorting to such static validation, it is crucial to take into account the lack of flexibility in such algorithms. Some solutions are adapted for handling single failure cases, but do not take into account the actual evolution in the life of a network when equipments are added or removed and the general architecture needs to adapt.

3.1.3 Path diversity

We have seen that the lack of visibility in route reflection architectures can cause a series of anomalies. The correctness issues are basically due to routing decisions based on an incomplete set of routes, meaning that anomalies appear because the total AS-wide knowledge is not propagated to every router. This section shows another consequence of information hiding, but this time the problem is related to the normal functioning of the protocol that restricts the number of routes propagated by delivering only the single best one.

The BGP design offers limited flexibility when it comes to path selection, enforcing single-path routing per destination prefix. The main goal of the protocol is to offer at least one path for each destination, which is enough in nominal conditions. On the other hand, in case of failure of a link or router, reachability is interrupted and traffic can get lost while waiting for failure recovery. It seems reasonable to desire alternative routes ready to be used in case of a network event, but these paths are not always visible.

Considering an AS as one entity, the amount of routes available for reaching a destination is enough for ensuring redundancy. Usually, a prefix can be reached through several neighbor ASes. The received paths can have a shared segment in the global Internet graph, but to the local AS they are all distinct because in the AS path, the first AS number is different. This type of variety in the paths is referred to as *next-hop-AS diversity*.

Within the ISP network, the ASBRs represent the exit point towards the destinations in the other ASes of the Internet. Prefix advertisements are often consistent on the links to the same neighbor AS, meaning that the advertised paths are similar, except for the next-hop attribute that designates the precise border router sending the BGP update message. This is quantified by the term *next-hop-router diversity*.

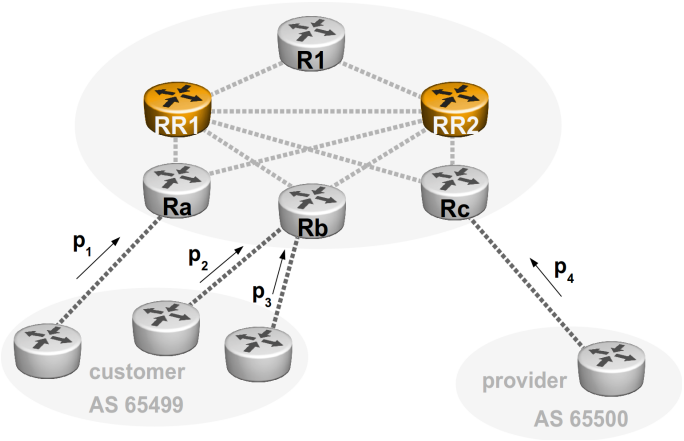


Figure 3.9 – Next-hop-router and next-hop-AS diversity for a given destination

In Fig. 3.9, next-hop-router and next-hop-AS diversity can be observed, as seen globally by an AS. The local AS receives route advertisements to the same destination from multiple

neighbors, in the present example from AS 65500 and AS 65499, so two next-hop-AS diverse paths are available. When looking at the next-hop-router diversity, four exit points can be used for sending traffic, one located in the provider AS 65500 and three in the customer AS 65499. If the next-hop-self [4] option is used, the diversity is reduced to three paths going out through the ASBRs R_a, R_b and R_c.

According to [Uhlig and Tandel, 2006 ; Uhlig and Tandel, 2005], some routes received on the eBGP sessions are never selected as best by any of the internal routers, but they are known only by the ASBRs that have directly received them. Among the several routes that manage to pass the selection algorithms, a very limited number are popular and thus get selected by the majority of the routers, stopping the propagation of many other routes inside the AS. The diversity loss becomes even more acute towards the core of the network, where the repeated decision process drastically reduces the availability of candidate routes. The same observation stands in the case of BGP/MPLS IP VPNs, as highlighted in [Pei and Van der Merwe, 2006] where the authors evaluate the problems caused by route invisibility.

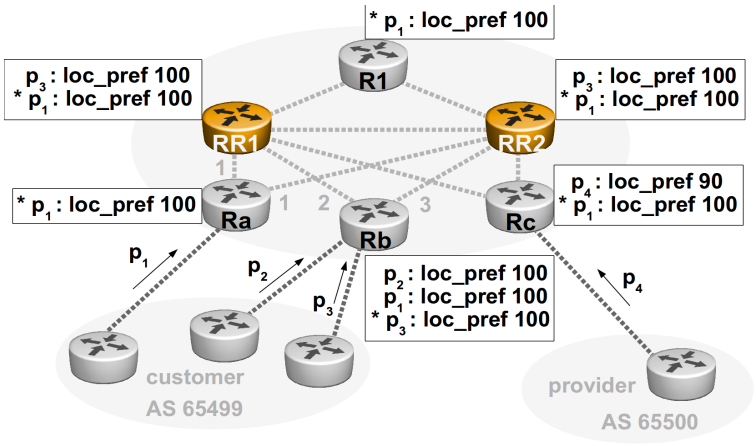


Figure 3.10 – Diversity loss within an ISP's network

There are several causes leading to a loss of diversity in the available paths. To illustrate the phenomenon, let us examine the output of the BGP decision process for the routers in Fig. 3.10. For example, R_b receives two paths, p_2 and p_3 from its external eBGP neighbors. However, the selection algorithm outputs only one single best path that R_b can propagate in the network. The other internal peers can discover that a second path exists only in case of failure [5], after the re-convergence of BGP.

Furthermore, even if a border router receives a route from an eBGP neighbor, it is not

4. Next-hop-self is used when the operator needs to specify the next-hop as being the ASBR in its own network. It is usual for when the external eBGP routers are not directly reachable through the IGP.

5. Even in case of failure, if the next-hop-self option is used, the inner routers are not aware which way the traffic goes out from R_b. The visibility of both external routes remains local to the ASBR. Indeed, if the session advertising p_2 goes down, R_b switches from p_2 to p_3 and the change is transparent to the internal peers.

guaranteed to pick the external route as best over an iBGP received route. Such is the case for R_c that receives the eBGP path p_4 from the provider AS and the iBGP path p_1 from its route reflectors. Since the provider path p_4 has a lower *local_pref* value than the customer path p_1, the router R_c uses p_1 and does not propagate the path p_4 to any of the other routers inside the network.

Fig. 3.11 reproduces a graph from [Oprescu *et al.*, 2011b] showing a measurement somewhat more recent of the BGP next-hop and AS diversity on five randomly picked routers.

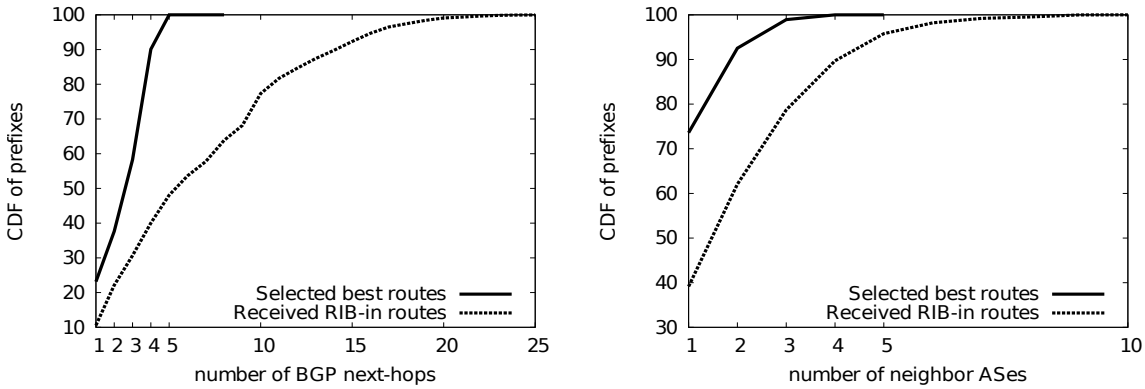


Figure 3.11 – Path diversity on five random routers

The data confirm previous findings: when it comes to next-hop-router diversity, the graph on the left side shows that more than 90% of the prefixes exit the AS through the same 4 next-hop-routers out of the total available of approximately 170 routers. The situation is not very different with respect to the neighbor ASes, with almost 95% of the prefixes being directed to a restricted set of only 3 next-hop ASes.

Increasing diversity in BGP

[Bonaventure *et al.*, 2004] states that it is the case to have more versatile route reflection in order to palliate the reduced diversity. In the proposed solution, route reflection adapts as a response to an evolutionary algorithm allowing it to optimize different objectives in the iBGP and reroute the traffic accordingly.

The improvements from [Pelsser *et al.*, 2008] aim to offer each router in the network at least two different ways for reaching distant destinations. This goal can be achieved by adding a small number of iBGP sessions between the border routers, according to the method described by the authors. They show that for complete diversity in all topologies, new external peering sessions should be established.

To minimize the impact in case of failure, the BGP Prefix Independent Convergence (PIC) solution covers a destination by providing data plane rerouting inside a service provider's network via an alternate path [Filsfils *et al.*, 2011]. The mechanism limits traffic loss to un-

der one second by re-engineering the FIB architecture in the routers towards a generalized hierarchical organization. With PIC, routers store a backup path in the RIB and in the FIB so that when failure is detected, the alternate path can quickly take over, enabling fast failover. Some of the usage restrictions stipulate that in the case of route reflectors only part of the control plane, there is no need for BGP PIC since only data plane convergence is addressed. Also, PIC and Best External [6] are two mutually exclusive features.

On a different note, but keeping in mind the same goal of ensuring robustness and even providing the basis for multi-path routing, *add-paths* [Walton *et al.*, 2011] proposes to include advertisements of multiple paths in BGP. This implies that each BGP speaker can advertise a set of paths (first 2 best paths, AS-wide best paths, all known paths...) for a destination prefix. An analysis of the different selection modes is available in [Van den Schrieck and François, 2009], whereas [Bornhauser *et al.*, 2011] quantifies the effect of such deployment on operational networks.

Scalability issues are often brought forward when evaluating the impact of the add-paths option; indeed routers will need to handle bigger routing tables because of the multiple paths advertised. The inflationary effect can be less dramatic if divided accross the network, as pointed out by Bornhauser.

3.1.4 Convergence time and path exploration

Studies show that in the global Internet, BGP convergence time can sometimes take up to tens of minutes [Labovitz *et al.*, 2000]. Although in previous work, the causes had been attributed to protocol interactions and timers such as Minimum Route Advertisement Interval (MRAI), in [Feldmann *et al.*, 2004] the authors pay attention to the impact of individual routers in the overall delay. A methodology is presented for quantifying the correlation between BGP pass-through times and the operational parameters like the rate of BGP update messages and the number of BGP peers. Their conclusion is that under certain conditions, the large number of routes and messages can be a major factor in slow convergence.

Further details about route propagation delays are covered in [Ben Houidi *et al.*, 2009], where the authors reveal that on several equipments, the transfers of BGP/MPLS IP VPN tables suffer from *gaps* that can take up as much as 90% of transfer time. The long periods during which there is no sending or receiving activity seem to be a consequence of the design choices adopted by router vendors. The analysis points out a timer-driven implementation, allowing routers to be idle as a means for controlling load and multiplexing between different sessions.

[Teixeira, 2005] gives a classification of the BGP routing changes and provides assumptions about the interactions between the IGP and BGP and their subsequent impact on traffic. Sensitivity to routing failures is studied and results show that small changes inside an AS

6. Best External is a feature allowing a router to have a backup path which is the most preferred route from external neighbors and which can be different from the best route currently installed in the RIB.

can have large effects on inter-domain routing, causing a cascade of BGP updates despite the fact that the AS path is not affected.

Such "noise" impacts the performance of the control plane in the BGP routers, since it requires a large memory and a lot of processing. Protocol messages arriving at an increased rate put pressure on the routers that need to accommodate the high activity peaks. The message-passing overhead takes up the CPU at the expense of real traffic. Although presumed to be a quiet protocol in stable networks, BGP activity can become quite important, since it depends on the behavior of other ASes.

Route Flap Dampening [Villamizar *et al.*, 1998] is a mechanism that tampers the churn provoked by a flapping link. If a link is unstable, BGP announces the path, then withdraws it, then advertises it again and the process keeps going. When the advertisements become too frequent, the router no longer accepts the path. This method it not recommended anymore, since it can be difficult to configure the right values for thresholds and timers. In addition, if there is an occasional failure and BGP explores several paths, the Route Flap Dampening mechanism can be triggered and thus delay convergence for a legitimate prefix. Recent work has been done to bring modifications to the algorithm so that it will no longer penalize well-behaving prefixes during the normal convergence process [Pelsser *et al.*, 2011].

Other work [Huston *et al.*, 2010] proposes the Path Exploration Dampening mechanism implemented and tested in the Quagga [Ishiguro, 1991] software routing suite. The selection algorithm takes into account the behavior of the AS path attribute across successive updates, leaving the other updates pass without any delay. The quantity of updates incurred with the proposed mechanism is compared to the number of update messages generated in existing damping mechanisms: Path Exploration Dampening behaves better than Route Flap Dampening, Withdrawal Rate Limiting and the MRAI.

The MRAI is a timer that rate-limits the message bursts on routers. Currently, the MRAI is subject to debate because it is difficult to configure with the appropriate values [Fabrikant *et al.*, 2011]. In [Wenhua *et al.*, 2009], the authors put forward new MRAI setup methods to improve route convergence as a function of different topology classes. They divide all possible topologies into distinct network types have specific characteristics related to the route update exchanging process. Based on these clusters, MRAI values can be adjusted to perform in a satisfactory manner.

BGP churn, policy interaction between different ASes and inter-domain routing behavior are interesting fields for research studies, but in this dissertation we limit ourselves to observing BGP behavior inside an AS. The solution presented in this dissertation regards the intra-AS routing and iBGP architectures, so work related to the previously mentioned subjects is hereafter out of scope.

3.1.5 Management and troubleshooting

The initial full mesh of iBGP sessions was simple to handle because of the flat architecture. With the arrival of confederations and route reflection, network management raised the stakes towards a more complicated manner of configuring, monitoring and controlling the behavior of the routers in the network.

Management and troubleshooting are often complex and challenging: inconsistency of the routing policies, path exploration meeting flap dampening and difficulties in achieving network-wide traffic engineering are some of the issues encountered by network operators. Although simple in definition, BGP is often difficult to configure and can be tricky to master. Engineers express the ISP's economic interest in an indirect and distorted manner through policies. Faults in BGP configuration can lead to the anomalies presented in the section 3.1.2 and can cause unintended routing between hosts in the Internet.

To solve some of the problems related to router misconfiguration, Feamster and Balakrishnan have come up with *rcc*, the router configuration checker [Feamster and Balakrishnan, 2005]. This tool relies on static analysis for determining the errors in the BGP configurations. It can detect two large types of faults: *route validity* faults that are related to paths not usable from a correctness standpoint and *path visibility* faults, obviously related to the hiding of routes that exist in the network.

The events that trigger cascades of updates in the AS graph of the Internet are generated by equipment failure or even router misconfiguration sometimes hard to discover. Network operators have difficulties in inferring the root cause of a routing change or even the AS responsible of the instability. The authors of [Teixeira *et al.*, 2007] introduce a method for diagnosing and determining the characteristics of BGP route dynamics between two neighboring ASes. Their findings show that there is a disparity in the reaction of each AS when confronted to similar BGP routing changes. The internal parameters such as network design, engineering decisions, even the number of BGP prefixes per session, the number of sessions per router and the BGP timers play a crucial role.

In [Wu *et al.*, 2005], the authors take up the challenging task of finding a needle in a haystack by identifying the essential BGP information in a large volume of measurement data. The proposed system can discover significant BGP routing changes in the traffic data collected by a Tier-1 ISP backbone.

The network operators strive to gather AS-wide data and subtract relevant knowledge that will help make an enlightened choice. Management and troubleshooting rely on humongous crops of data that need to be rendered human-readable. The complexity of steering such large networks could be avoided if the network status can be concentrated on a reduced number of routers. Such ideas have stemmed routing platforms that aim to give more control over the network variables while at the same time offering robustness in routing.

3.2 Routing Platforms to fix iBGP

Nowadays, routing is done in a highly distributed manner, as seen in Fig 3.12. The need to have a consistent view of the network state requires new design constraints for routing. Similar to the concept of Path Computation Element in an MPLS network, routing platforms separate the selection of paths (routing plane) from the actual forwarding (data plane) onto distinct equipments. This aggregation of the routing intelligence allows a network operator to reduce the number of contact points useful for controlling the network. If policies and configuration can be deployed from only a fraction of the routers inside the AS, then the management effort is less substantial.

From an organizational standpoint, routing platforms can be situated between the decentralized and the distributed network topology (see Fig. 3.12). When reducing the redundancy specific to highly distributed networks, the design of a routing platform must concentrate on guaranteeing robustness, while at the same time achieving the desired control.

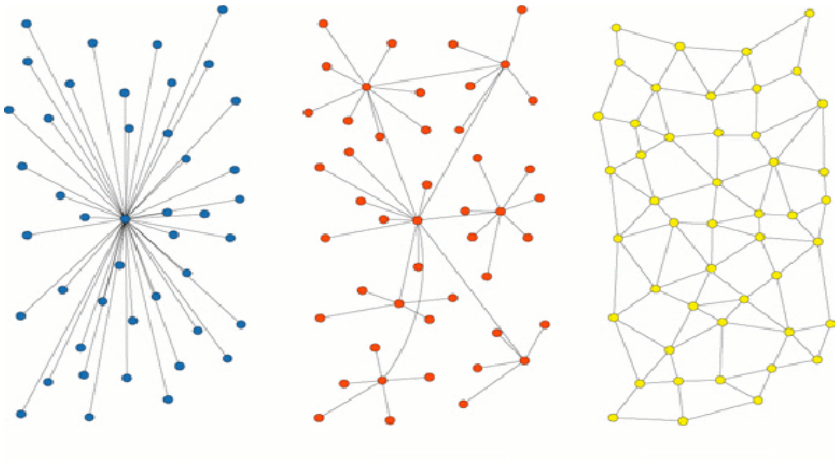


Figure 3.12 – Centralized, decentralized and distributed networks (P. Baran)

Other routing platforms may position themselves as concurrent solutions to the architecture we here present. The next paragraphs describe two routing platforms presented in the literature: Routing Control Platform (RCP) and SpliTable.

N. Feamster et al. [Feamster *et al.*, 2004a] advocate the interest of separating routing from the routers. Their proposal, called Routing Control Platform (RCP), is based on three architectural principles: path computation in accordance with a consistent view of the network, mastered interactions between the stacked routing protocol layers and finally, an expressive specification of the routing policies. RCP aims to offer separate selection of routes on behalf of the routers while maintaining backward compatibility and transparency to neighbor ASes.

M. Caesar et al. later offer an implementation to the RCP concept. The prototype described

in [Caesar *et al.*, 2005] has three modules: the IGP Viewer to collect topology information, the BGP engine that learns the BGP routes, performs the decision algorithm and then communicates the best paths to the routers and finally the Route Control Server that processes messages received from the other two modules and makes it possible to store one single copy of each BGP route, keep track of the routers to which each route has been assigned and maintain an order of preference of the egress point for each router. We extend this work by going a step further in reaching scalability: in our approach, the prefix table is split, making possible parallel computation of routes while in the RCP solution, all the BGP information is concentrated in one point, even if there are multiple replicas of it.

C. Pelsser et al. propose a method for scalable support of inter-domain routes inside a single AS. The proposal is called SpliTable [Pelsser *et al.*, 2009] and it relies on distributed servers that perform the selection of routes on behalf of the routers. The routes are stored in an adapted Distributed Hash Table (DHT) and as a beneficial consequence, each router keeps only a share of the Internet routes plus a cache of the routes currently in use for forwarding. The authors show a comparison in terms of control messages between SpliTable, ViAggre and traditional route selection in full mesh and sparse topologies. The implementation of a SpliTable prototype is later presented in [Masuda *et al.*, 2011]. A new evaluation is provided, for a sample AS topology reproduced in a virtualized environment.

While RPC is a mere concentration of the network view in one AS-wide entity, the SpliTable concept comes closer to the idea of using a division of the routes in the network, through the use of distributed route servers. The described routing platforms take into account the classical BGP model, whereas the work presented in this dissertation focuses on a new iBGP architecture that responds to the evolution of the network in terms of growth. Although the concepts are similar in the presented solutions, none of them integrates the distribution of the control plane in iBGP routing which is key in achieving scalability in the near future.

[Koponen *et al.*, 2010] advocates a common control platform applicable in large scale networks. It presents a comprehensive API that allows the operator to tradeoff between generality, scalability, reliability, simplicity and control plane performance. According to the authors, Onix is "a platform on top of which a network control plane can be implemented as a distributed system". The Onix concept stems from the need to better tailor the desirable functionalities in a network by separating the control plane from the forwarding elements. Onix provides modularity in network design and queries, while relying on a common Network Information Base. As stated by the authors, the Onix platform is an enabler for managing the state of the network, but does not magically solve scale and consistency problems by turning the networking issues into distributed system issues.

3.3 Summary and Remaining Issues

This chapter has explored the various drawbacks that a BGP network administrator is likely to encounter. A literature survey of previous studies allows the reader to get a

clear picture of the reported problems and the associtated efforts for improving scalability, correctness, visibility and management of routing.

The first recurrent aspect is that the majority of the issues discussed in this chapter come from the use of route reflection. A compromise has been made between scalability and uncontrolled or unknown behavior of the protocol in various topologies. In order to scale, route reflection hides information, ultimately resulting into a lack of visibility that can turn out to be harmful. Sub-optimal routing and deflections are not likely to happen in practice because of the popular deployment of MPLS. Tunneling the traffic to designated egress points avoids these two inconveniences.

The reader can find a summary of the approaches covered by this chapter in Table 9.2. A rough classification determines the goals of each proposed solution.

There is a need, however, to separate theory from practice. Some of the presented issues are commonly avoided with engineering tricks and configuration tweaking. Network operators adapt to inadvertences by enforcing specific route reflector placement and building convenient topologies that behave correctly. Ideally, these aspects can be handled in an automated manner and this dissertation proposes an approach for better control over the network.

Assumptions

The utility of the oBGP solution relies on some preliminary hypothesis regarding the evolution of the Internet. The most important assumption we make is that the size of the routing table will continue to expand and that this gradual RIB growth cannot be backed up by the equipment capacity. Frequent upgrades to more powerful routers have been the solution in the past. We argue that vertical scalability will become more expensive and operators will opt for the introduction of multiple equipments that work in a distributed manner towards a common goal. This premise is valid to a greater extent if the wide spreading of the *add-paths* option is considered. The effects of the adoption of *add-paths* are yet to be quantified at the global scale, but it is presumed that its deployment can lead to a drastic increase in the number of BGP entries.

The second major assumption is that the popular prefixes that carry most of the user traffic in the Internet will continue to remain stable. Indeed, the bulk of the Internet traffic is made of a few prefixes that are responsible for the majority of the data volume. Other than that, these destinations are remarkably stable and account for a very tiny portion of the total number of BGP events. A plausible explanation would be that the frequently accessed services are hosted on managed platforms that offer high availability. Quality being a constraint, the equipments are well managed and when problems arise they tend to be solved rather quickly [Rexford *et al.*, 2002].

Table 3.1 – Taxonomy of previous solutions

proposed solution	requirement
route reflection & confederations	scalability
filtering of bogon & martian prefixes, prefix aggregation	
Ballani *et al.* — FIB reduction with ViAggre	
Zhang *et al.* — Core Router-Integrated Overlay (CRIO)	
Dobrescu *et al.* — add capacity with RouteBricks	
Vissicchio *et al.* — migrate to more route reflection levels (MIRTO)	
Griffin and Wilfong — definition of iBGP correct topology	correctness
Griffin and Wilfong — analysis of the MED oscillation problem	
Flavel and Roughan — stable and flexible iBGP	
Van den Schrieck *et al.* — lightweight BGP sessions	
always-compare-med & set-deterministic-med	
Griffin and Sobrinho — Metarouting Project	
use of MPLS to avoid deflections	
Raszuk *et al.* — optimal route reflection draft	
Rawat and Shayman — iBGP graph construction	
Feamster *et al.* — (up)*(over)?(down)* propagation model	
Buob *et al.* — fm-optimality check and iBGPv2	
Sarakbi and Maag — session reduction with BGP Skeleton	
Vutukuru *et al.* — BGPSep for a correct RR topology	
Uhlig and Tandel — impact of route reflection on diversity	path diversity
Bonavelnture *et al.* — intelligent route reflection draft	
Filsfils *et al.* — BGP Prefix Independent Convergence (PIC)	
Pelsser *et al.* — iBGP next-hop diversity	
Walton *et al.* — add-paths draft	
Bornhauser *et al.* — scalability of add-paths	
Feldmann *et al.* — BGP pass-through times	convergence
Ben Houidi *et al.* — table transfer gaps	
Teixeira — sensitivity to routing changes	
full mesh	management & troubleshooting
Feamster and Balakrishnan — router configuration checker	
Feamster *et al.* & Caesar *et al.* — Routing Control Platform (RCP)	routing platforms
Pelsser *et al.* & Masuda *et al.* — SpliTable	
Koponen *et al.* — Onix	

Chapter 4

The oBGP Solution

After an extensive state of the art, this chapter introduces the main concepts and ideas that build the basis of the oBGP model. The opening Section 4.1 provides the context and the objectives of the oBGP routing platform: in a complex routing environment, the oBGP model comes to replace the classic iBGP architectures that suffer from multiple drawbacks and proposes a new routing paradigm based on a distributed routing platform. Coming from its design that separates the control plane from the forwarding plane, the oBGP framework brings a set of new features such as improved visibility on the choice of available routes, correctness guarantees and controllable scalability in terms of routing table size and session meshing.

Further, the second section consists of a short presentation of the graph models used throughout the manuscript. The following Section 4.3 unveils some of the essential concepts that allow oBGP to function and be more flexible than classic routing: the splitting of the reachable IP space into several virtual prefixes that are managed according to previously defined boundaries in the form of control sub-planes. Since a network endures changes from the other ASes in the Internet, the interaction could imbalance the distribution of virtual prefixes to the pre-defined sub-planes and thus an algorithm is necessary for reaching a state of equilibrium. This issue is solved by the method proposed in the sections regarding allocation and re-allocation of virtual prefixes.

Once the design principles have been presented, the final part of the chapter takes up the challenge of illustrating the general architecture applicable on top of the oBGP model. Presented in the form of articulated pieces, the three main views conclude the chapter: the overall view of the network, then the more detailed view of a specific sub-plane and finally a zoom in on the client that receives its routes from the oBGP platform.

4.1 Overview

In today's IP networks, routing is highly distributed: each router in the AS makes its own decisions. We propose to separate the selection of paths (routing plane) from the actual

forwarding of traffic (data plane) on distinct equipments. Offloading the control plane from the routers can be seen as a remedy to the explosion of the routing table size and provides more visibility of the routes received by the AS, guaranteeing thus more accurate and correct routing choices.

When rethinking the current design, we place all the knowledge of routing data into a separate iBGP routing plane handled by an overlay of routing processes that do not forward user traffic. We propose to implement BGP routing engines called *oBGP*. The oBGP nodes act as a distributed entity that collects all messages from the domain border routers that are connected to the external peers through eBGP sessions. This approach allows the overlay to receive all the routes from the neighboring ASes and gather the announced routes to achieve a unified complete view.

The purpose of the oBGP framework is to provide a viable alternative to iBGP routing in the light of future evolutions of the Internet in terms of growth (e.g., number of ASes, BGP RIB size, number of paths advertised for a single destination) and to fix some of the existing anomalies in iBGP. The main goal of oBGP is to collect the eBGP received routes and redistribute them within an AS, allowing thus the routers and hosts to reach external prefixes. At this stage, it is important to understand that oBGP replaces completely the iBGP session mesh and that these two architectures are distinct and should not operate in a hybrid manner due to complications that may appear in routing.

In the long term, oBGP routing software is intended to be integrated by vendor equipment, but can also be executed by additional servers running on commodity hardware. The logical overlay is composed of routing processes (or nodes) that are jointly responsible of:

- collecting, splitting and storing the complete set of routes received from eBGP and the internally originated routes,
- storing the routing policies and configurations of all the routers in the AS,
- redistributing the computed paths to the client routers.

One of the main concerns of an iBGP architecture is its ability to scale: support the growing routing table and handle protocol messages over time. To achieve scalability, we design an oBGP solution where the routing information is divided in several sub-planes. In this approach, distinct subsets of overlay nodes each handle only a fraction of the entire set of prefixes in the routing table.

The next paragraph explains the passage of a route advertisement in the oBGP overlay from the arrival in the AS to the installation of the best path in the RIB. Fig. 4.1 shows the chronological steps of a route announced to the oBGP overlay.

The first contact between the neighboring ASes and the oBGP platform happens at the border router. Even if it is possible to use multi-hop eBGP sessions to reach nodes deeper in the topology, some network operators are reluctant because of security concerns. When a route towards a destination (e.g., the prefix 203.0.113.0/24 from Fig. 4.1) is advertised in the Internet, it reaches the first router, the ASBR — called a distributor node in oBGP — that determines the corresponding sub-plane in charge of the prefix. The distributor then forwards the information to the oBGP nodes handling the correct sub-plane. After running the BGP decision process and applying the according configuration and IGP

topology constraints, the node outputs a best path per prefix for each of its client routers. The overlay distributes the best paths to the client routers connected through sessions and they can immediately install it in their RIBs. Upon reception of the best route, the native routing mechanism takes course and installs the path to the prefix in the FIB for actual packet forwarding.

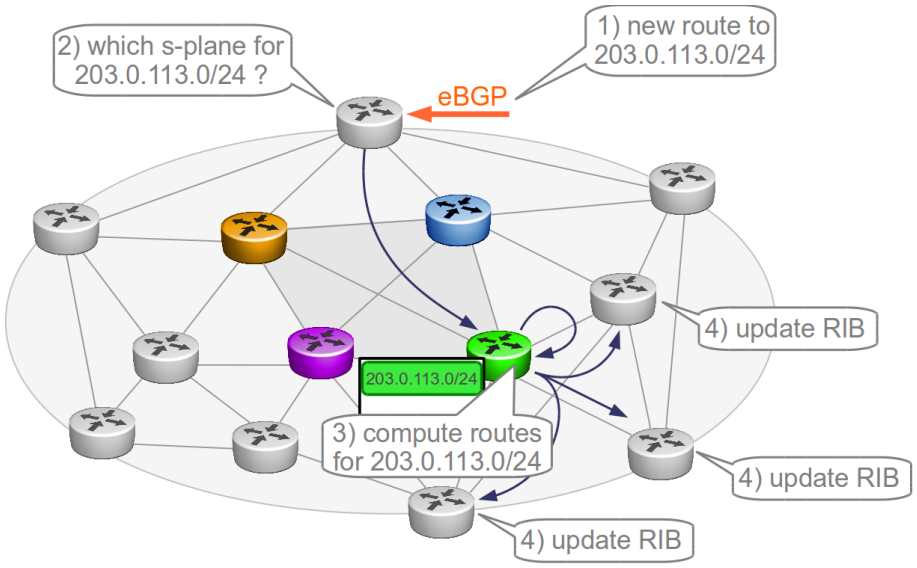


Figure 4.1 – The steps followed by an advertisement in the overlay network

The oBGP nodes need to be aware of the actual mapping of the reachable IP space within the overlay. To insure resiliency and avoid a single point of failure, a sub-plane is replicated on several oBGP nodes (not pictured in Fig. 4.1, but developed later in Chapter 5). Coordination between the copies of sub-planes is accomplished through an exchange of meta-data across the overlay. The following paragraphs depict the sub-plane concept.

4.2 Graph models

To formally introduce the oBGP concept, several graph models are necessary. The following paragraphs present the underlying IGP and iBGP graph models and the notations associated to network elements.

The network represented by an AS can be modeled with the help of two graphs describing the IGP and the BGP topology. The two graphs need not be identical, but they are usually superposed and all the routers in the IGP are also running a BGP instance. The terms node and router are used interchangeably. We further consider an equipment running both an IGP and a BGP instance to be one same router designated by one node in each of the two graphs.

4.2.1 The IGP graph

IGP graph

Let us consider V the set of vertices (routers) and E the set of edges (links) in the network. If a link exists between two nodes u and v, we denote it as the directed edge (u, v) and its associated IGP metric. Let $d(u, v)$ be the distance between u and v. Note that there is a distinction between the symmetric links (u, v) and (v, u) since the two IGP metrics can be configured with different values. We define the directed graph $G_{IGP} = (V_{IGP}, E_{IGP})$ as the model of the IGP topology.

IGP path

We define $path(u, v)$, a path between two vertices u and v in the graph G_{IGP}, as a sequence of nodes $(v, v_k, v_{k-1}, ..., v_0, u)$, $k \geq 0$, such that $\forall i, k \geq i > 0, (v_i, v_{i-1}) \in E_{IGP}$.

IGP shortest path

If $u, v \in V_{IGP}$, then $spf(u, v)$ denotes the shortest path from u to v. In case multiple shortest paths exist between the same pair of vertices, then $spf(u, v)$ refers to any of these shortest paths.

IGP metric

We use the notation $|u, v|$ to designate the cost of the shortest path between the vertices u and v. This distance is usually computed as the result of Dijkstra's algorithm applied on the G_{IGP} graph.

Connectivity

In a graph G_{IGP}, two vertices u and v are part of a connected component if G_{IGP} contains a path from u to v denoted by $path(u, v) = (u, v_k, ..., v_l, v)$ and a path from v to u denoted by $path(v, u) = (v, v_m, ..., v_n, u)$.

4.2.2 The iBGP graph

BGP router

Each router in the network is represented by a vertex in the iBGP graph. We identify two distinctive sets of BGP routers:

- $\mathcal{S}$ = the set of ASBR routers in the AS with $card(\mathcal{S}) = s$.

- $\mathcal{T}$ = the set of all the BGP routers in the AS. In particular, $\mathcal{S} \subseteq \mathcal{T}$ with $card(\mathcal{T}) = t$ and $s \leq t$.

iBGP session

An iBGP session established between two routers u and v is denoted by two directed edges (u, v) and (v, u). We consider $G_{iBGP} = (V_{iBGP}, E_{iBGP})$ the graph of iBGP sessions, where $V_{iBGP} = \mathcal{T}$. According to the type of session between the vertices, a label is attributed to each edge (u, v):

- if u acts as a route reflector for v, the label is *down*.
- if u is a client of v, the label is *up*.
- otherwise u and v are iBGP peers of the same level, the label is *over*.

$L_{iBGP} = \{up, over, down\}$ is set of iBGP labels corresponding to the iBGP sessions. The function $label : E_{iBGP} \rightarrow L_{iBGP}$ returns the label associated to a given edge of the graph.

Valid path

We introduce the notion of a valid signaling path. If an edge $(u, v) \in E_{iBGP}$ such that u and v belong to the same IGP connected component, then the session (u, v) is called *mountable*. A BGP message can be propagated along a valid path that contains zero or more *up* edges, followed by zero or one *over* edge, followed by zero or more *down* edges. For more details, we refer the reader to [Feamster *et al.*, 2004b] and [Buob, 2008]. We here abuse notation conventions and consider that all the iBGP paths following the pattern of a regular expression are valid paths:

$$\mathcal{P}_{valid} = \{(up) * (over)?(down)*\}$$

Update messages are exchanged between an internal router and a next hop that receives and forwards them to or from the external peers. An iBGP signaling graph is said to be valid if at least one valid path exists for each couple ASBR-router:

$$\forall u \in \mathcal{S}, \forall v \in \mathcal{T} \quad \exists\ path(u, v) \in \mathcal{P}_{valid}$$

4.2.3 The oBGP graph

All the routers already present in the BGP topology are included in the oBGP topology, with some additional features that might be necessary for handling tasks such as load balancing or more advanced operations such as optimal route reflection. The routers can be divided into three main categories corresponding to their specific role.

oBGP nodes

We denote the set of intelligent oBGP nodes with $N_{oBGP} = \{N_1, ..., N_i, ..., N_m\}$, where $0 < i \leq m$, with m being the total number of nodes that act as enhanced route reflectors.

oBGP clients

The oBGP nodes perform computations and distribute routes to a set of routers that act as clients of the oBGP platform. $C_{oBGP} = \{C_1, ..., C_i, ..., C_c\}$, where $0 < i \leq c$, with c being the total number of client nodes that receive routing information from the intelligent oBGP nodes.

Distributors

The routers that handle the distribution of the routing information across the multiple oBGP nodes and that feed the incoming external routes to the oBGP platform are called distributors. In a real network, they can be implemented as load balancers. We denote the set with $D = \{D_1, ..., D_i, ..., D_a\}$, where $0 < i \leq a$, with a being the total number of distributors in the network.

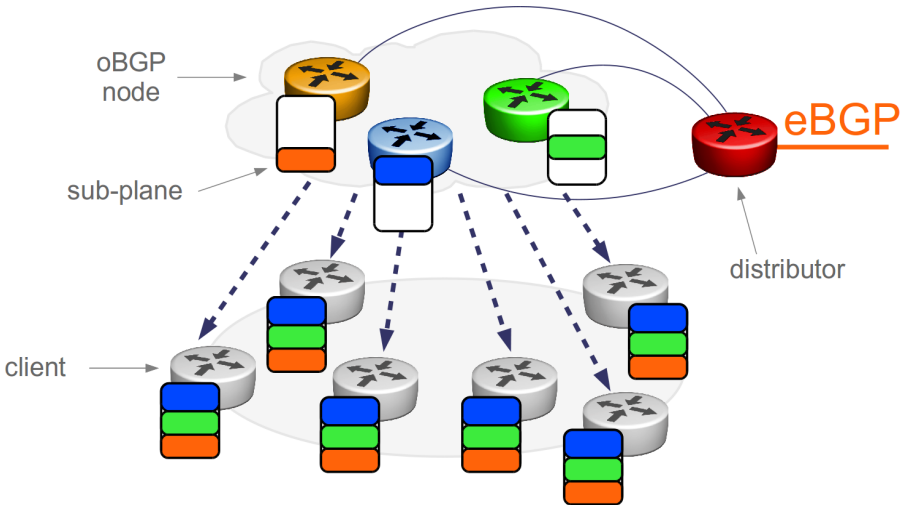


Figure 4.2 – The elements of the oBGP framework

4.3 Design principles

oBGP is an evolved routing platform that sets about dealing with multiple issues that occur simultaneously in today's iBGP networks. The way oBGP is designed responds to several challenges that production networks face on a common basis:

■ **scalability** in terms of RIB entries, number of sessions and protocol message load. The need for control plane scalability is a response to current limitations of TCP mechanisms that cannot be solved by upgrading the CPU or adding more memory to routers. Vertical scaling is a short-term strategy that brings relief for only a few years and there is no guarantee that it can work forever. For this reason, there is a growing movement driven both by academia and industrial research to separate the control plane from the forwarding elements. This decoupling of the routing intelligence from the data plane allows the building of the control plane as a distributed system. The distributed control plane thus handles the original workload as a task split into multiple fractions. How does oBGP handle scalability? First of all, oBGP makes a clear distinction between routing and forwarding: the oBGP smart nodes in the platform are in charge of computing the best routes for the client routers, much in the same way a Route Server would do. The client routers become in fact simplified equipments, keeping only minimal routing information used for the actual forwarding. Second, the control plane information is split into several containers called **sub-planes** that are disjointly assigned across the multiple oBGP nodes.
We denote as $\mathbb{P}$ the whole set of IPv4 addresses, the equivalent of 0.0.0.0/0. If the total IP space is divided in n sub-planes $S_1, S_i, ..., S_n$, then we have:

$$\cup S_i = \mathbb{P},\ 0 < i \leq n$$

Also, the routing information needs to be disjoint between the various sub-planes, so the subsets of $\mathbb{P}$ attributed to the sub-planes do not overlap, which translates into:

$$S_i \cap S_j = \varnothing,\ \forall i \neq j,\ 0 < i, j \leq n$$

■ **visibility** of the external routes received on eBGP sessions and all internally generated routes. The oBGP nodes collect all the BGP advertisements received or generated by the routers inside the AS, acting as a single entity having a global unified view of the network, similar to a common library. The increased visibility allows for a more reliable decision process that can take into account the complete set of routes seen by the AS as a whole. The extensive perception of available routes in oBGP helps avoid phenomena such as information masking due to route reflection or diversity loss due to the cascaded concealing of non-preferred BGP routes. Although acting as a distributed system, the oBGP platform aims to provide an exact and consistent network view, federating all the information available to all the routers in the domain.
If $\mathcal{P}_v^{ext}$ denotes the set of all external BGP paths received by the ASBRs $v \in \mathcal{S}$ and $\mathcal{P}_u^{int}$ denotes the set of all internal paths generated by the routers of the AS $u \in \mathcal{T}$, then the oBGP nodes $N_i,\ 0 < i \leq n$ aggregate all these available routes:

$$\cup \mathcal{P}_{N_i} = \mathcal{P}_v^{ext} \cup \mathcal{P}_u^{int},\ \forall v \in \mathcal{S}, \forall u \in \mathcal{T},\ 0 < i \leq n$$

■ **correctness** of routing and forwarding. Guaranteeing correctness can sometimes be tricky, but the oBGP design makes it possible to avoid anomalies related to the masking of the underlying IGP graph. Indeed, the oBGP nodes performing the BGP decision

process are aware of their own position in the topology graph like regular route reflectors, but most importantly they can perform the selection algorithm from the client's standpoint. When determining the next hop, it is useful to know the exact situation in the graph of the instance that will actually be using the route to forward traffic. The oBGP nodes compute the best route based on all the available candidate routes on behalf of the client router, avoiding sub-optimal routing and possible deflections due to the IGP metric.
In practice, we need to verify that no matter the metric on the links, for a given destination prefix, the oBGP nodes will always prefer the same next hop that the client router would have preferred. In the conventional routing, this means that the client is situated between the oBGP node and the AS exit point:

$$c_i \in spf(N_j, v_k),\ \forall c_i \in C_{oBGP}, \forall N_j \in N_{oBGP}, \forall v \in \mathcal{S}$$

From the metric point of view, the oBGP node is farther from the AS exit point than the client that will actually forward traffic through the next hop:

$$d(N_j, v_k) \geq d(N_j, c_i) + d(c_i, v_k),\ \forall N_j \in N_{oBGP}, \forall v \in \mathcal{S}, \forall c_i \in C_{oBGP}$$

- **reliability** and robustness in case of a network event (router, link, session or oBGP node failure). Redundancy is an important aspect since simple and double failure scenarios can reveal robustness to be a critical point of such a framework. As seen in the previous chapter, routing platforms are spread on multiple equipments, but remain slightly less distributed than the "classic" form of routing.
To offer competitive routing performance compared to the current paradigm, oBGP keeps multiple replicas of the routing information contained in the different sub-planes. It is equally important to provide a general solution for meshing the nodes that store identical copies of the same sub-plane. Section 5 provides details about two possible redundancy schemes, with a study on the resilience of the network when faced with simple failure cases and other multiple failure scenarios.

4.3.1 Distributed sub-planes

A router learns routes toward a given prefix from its neighbors, and in the general case routers of the same AS do not learn the same exact set of routes or the same quantity. The full visibility of BGP routes received from external ASes can be assimilated to a sum of queries on all border routers of an AS. The total of routes received on the border routers is equivalent to the global view of the advertised Internet as seen by the domain.

oBGP manages to keep this external view intact by indexing it directly in the overlay according to a mapping mechanism. The oBGP nodes act as an aggregator for the collection of external messages received by the border routers of the AS who establish eBGP sessions with neighbor ASes.

Storage of prefixes is distributed across the overlay and nodes divide between each other the computational load of the control plane. We define several chunks of the reachable

address space that are allocated on distinct nodes. These large IP spaces are called routing sub-planes. The overlay is in charge of keeping a coherent state: no pair of sub-planes has overlapping prefixes and they are stored on different nodes. In a possible implementation, a structure similar to a distributed hash-table can be used for managing the sub-planes or simple route filtering on the oBGP nodes can determine which routing information to accept (as configured on the node) and which to discard. The oBGP nodes guarantee the frontiers of the sub-plane, but another aspect to take into account is the replication of the information on the nodes covering the same sub-plane.

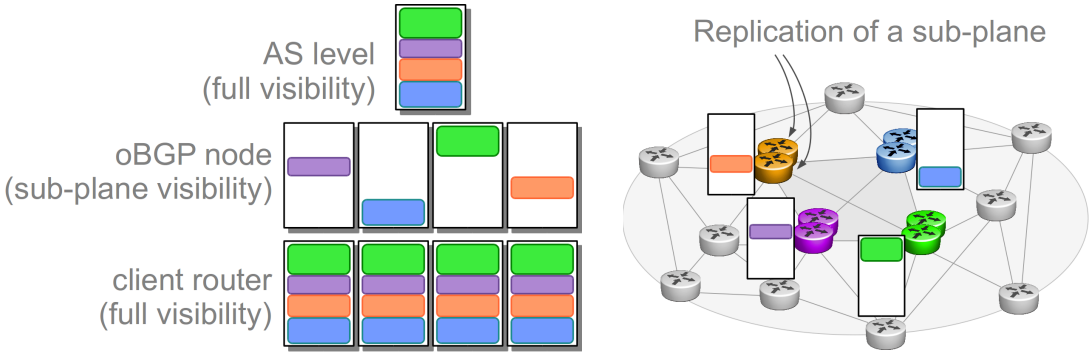


Figure 4.3 – The routing table is split between the $n = 4$ sub-planes of the overlay

4.3.2 Index of Virtual Prefixes

The mapping of the sub-planes on the oBGP nodes takes into account a split factor n (e.g. $n = 4$ as seen in Fig. 4.3) and allocates each chunk of $total/n$ prefixes to a corresponding sub-plane. This strategy turns out to be very coarse grained and thus we introduce smaller containers for the IP space (denoted by $\mathbb{P}$) called Virtual Prefixes as in [Zhang *et al.*, 2006].

Table 4.1 shows an example of a possible configuration of the sub-planes: the reachable IP space is divided in $n = 4$ sub-planes and each sub-plane covers the equivalent of a /2 prefix (consisting of roughly 2^{30} possible hosts). To better control the load incurred by the oBGP nodes handling the sub-planes, the network operator may choose to define several virtual prefixes as is the case for sub-plane 1 that contains 2 virtual prefixes or the sub-plane 4 that contains 3 virtual prefixes. The virtual prefixes may be swapped between the oBGP nodes in order to achieve a balanced load on the sub-planes. Data [1] in columns 3 and 4 shows that the density of prefixes advertised in the Internet can be almost uniformly distributed across the previously defined sub-plane space.

Another solution for dividing the IP space reachable in the Internet is to take into account the number of routes advertised by the external peers for each of the received BGP announcements. In this manner, the network operator may decide to make a distinction between destinations that can be reached through many diverse paths and other prefixes

1. Private Tier-1 AS, dataset of November 2010, based on a total of 354682 prefixes.

Table 4.1 – Sub-planes containing virtual prefixes

sub-plane ID	virtual prefixes	# of prefixes	% of total
sub-plane 1	64.0.0.0/4	53250	17.85 %
	32.0.0.0/3	21408	7.17 %
sub-plane 2	160.0.0.0/3	38425	12.88 %
	192.0.0.0/4	37667	12.62 %
sub-plane 3	80.0.0.0/4	34552	11.58 %
	96.0.0.0/3	35679	11.96 %
sub-plane 4	208.0.0.0/4	40207	16.82 %
	128.0.0.0/3	17719	5.93 %
	0.0.0.0/3	9411	3.15 %

that have poor diversity at the edge of the AS. When looking to balance the allocation of Internet destinations to a specific virtual prefix, the operational teams can mix equitably prefixes with high diversity and prefixes with a lower number of associated paths. Pairing up prefixes based on this criterion offers higher precision in evaluating the potential processing load on the oBGP nodes. As an example, for a prefix that can be reached through 23 paths, the BGP decision process needs to lookup and compare all the 23 candidate routes for every oBGP client, whereas for a prefix with a diversity of 2, the processing is much faster. Note, however, that this type of allocation of prefixes to the sub-planes generates an increased number of virtual prefixes, since the distribution of the actual prefixes is no longer done based on the contiguous IP blocks.

4.3.3 Allocation of prefixes to sub-planes

For a more tight management of the load on the oBGP nodes, it is possible to develop an off-line procedure that allocates virtual prefixes to the oBGP nodes to obtain a fine grain arrangement. Such a procedure can be based on a greedy algorithm that orders the virtual prefixes from largest to smallest and distributes them according to the existing load of the sub-planes.

Algorithm 4.1: The algorithm for balancing virtual prefixes across oBGP sub-planes

```
Input: Set of virtual prefixes
Output: Mapping of virtual prefixes to sub-planes
begin
    order all virtual prefixes from largest to smallest;
    foreach virtual prefix do
        mark as unallocated;
    end
    foreach virtual prefix do
        if virtual prefix is unallocated then
            allocate largest virtual prefix to smallest sub-plane;
            mark virtual prefix as allocated;
        end
    end
end
```

The following example shows a distribution of the same set of data as in the table 4.1. The single column on the left side represents the first phase of the algorithm, when the virtual prefixes are ordered decreasingly according to the parameter that the network operator wants to optimize. In this case, the ordering of the virtual prefixes is done as a function of the total number of actual Internet prefixes contained. The ratio expresses the relative size of destinations contained in a virtual prefix with regard to the total number of destinations in the RIB.

The first step allocates the biggest virtual prefix that holds 17.85 % of destinations to the sub-plane 1. The following steps are similar until the algorithm reaches sub-plane 4 and needs to evaluate where to allocate the following prefix since all the sub-planes have been allocated one virtual prefix by now. As mentioned in the algorithm, in an attempt to obtain a proportionate allocation, the following prefix goes to the smallest sub-plane, i.e. sub-plane 4. The attribution continues until the last virtual prefix available. At the end, the arrangement of virtual prefixes results in the following distribution:

Note that there is a compromise between the simplicity of the algorithm and the balance achieved across the various sub-planes. Indeed, the equilibrium of the distribution depends on the size of the virtual prefixes: it is easier to achieve a harmonious arrangement if the virtual prefixes are somewhat homogeneous. The decision about the degree of granularity of the virtual prefixes belongs to the operational entities running the network that must take into account future evolutions.

The natural growth in the number of advertised destinations and the instability of certain prefixes can influence the actual state of the virtual prefixes, making them evolve in time. To keep a fair balance, a periodic check is necessary in order to verify that the initial splitting of the space into the current virtual prefixes and their corresponding sub-planes is still satisfactory. If the division of the reachable space into several virtual prefixes is not very demanding in terms of complexity and network availability, the situation is different

Table 4.2 – The algorithm allocates the defined virtual prefixes to the four sub-planes

virtual prefixes	step	sub-plane 1	sub-plane 2	sub-plane 3	sub-plane 4
17.85 %	1	17.85 %	.	.	.
16.82 %	2	.	16.82 %	.	.
12.88 %	3	.	.	12.88 %	.
12.62 %	4	.	.	.	12.62 %
11.96 %	5	.	.	.	11.96 %
11.58 %	6	.	.	11.58 %	.
7.17 %	7	.	7.17 %	.	.
5.93 %	8	5.93 %	.	.	.
3.15 %	9	3.15 %	.	.	.
	total=	26.93 %	23.99 %	24.46 %	24.58 %

for the reallocation of virtual prefixes to the sub-planes. Rerunning the greedy algorithm might turn out to be a bad idea since many virtual prefixes could change sub-planes. A more sensitive approach is to limit the impact and perform minimal changes by identifying the biggest disparities between two sub-planes and swapping virtual prefixes. The idea consists of finding two virtual prefixes, one in each sub-plane, who have a difference between the prefix counts that is equal to half the global difference between the sub-planes and then swap them.

In the example depicted by Table 4.3, two sub-planes have become very imbalanced due to a contrasting density in the repartition of the actual BGP prefixes within the virtual prefixes. The smallest virtual prefix in sub-plane 2 ends up receiving a lot less announced destinations (going from 7.17% to only 4.18%), whereas another virtual prefix in sub-plane 1 has a much higher density of the BGP prefixes at the AS level, with a staggering increase from 5.93% to 8.92%. This shift builds a gap of approximately 10% of the total prefixes between sub-planes 1 and 2. The solution is to swap the two virtual prefixes between these two sub-planes and reach a more balanced global repartition.

Table 4.3 – Disproportionate evolution of virtual prefixes and reallocation

s-pln 1	s-pln 2	s-pln 3	s-pln 4
17.85 %	16.82 %	12.88 %	12.62 %
8.92 %	4.18 %	11.58 %	11.96 %
3.15 %	.	.	.
29.92 %	20.00 %	24.46 %	24.58 %

s-pln 1	s-pln 2	s-pln 3	s-pln 4
17.85 %	16.82 %	12.88 %	12.62 %
4.18 %	8.92 %	11.58 %	11.96 %
3.15 %	.	.	.
25.18 %	25.74 %	24.46 %	24.58 %

An aspect to take into account before reallocating virtual prefixes to sub-planes is the threshold that sets off the swapping. When should the network engineers trigger the operation of reallocation? The difference of load becomes important when the most stressed sub-plane reaches certain imposed restrictions related to sizing limitations. For example, the operational impact is not the same if in a two-plane division, a single sub-plane handles

80% and the other one handles 20% compared to a situation where two sub-planes handle 40% and 60% of the total prefixes. The oBGP node running 80% of the total protocol load may be approaching a situation where it can no longer handle the computational overhead. For a more detailed discussion, please refer to Section 7.1.

It is also possible to enforce a more complicated rule allowing for paths to popular prefixes to be cached in the oBGP nodes based on a statistical computation of the frequency of occurrence. One option is to cache the popular prefixes that are more stable as opposed to swapping more often the less popular prefixes.

4.4 General architecture

The concepts behind oBGP rely on a distribution of the control plane across several nodes that each handle computations and route redistribution for a fraction of the $\mathbb{P}$ set. This division is intimately associated with a network architecture that can handle the constraints and goals of such a routing model. oBGP is a general framework for a family of possible architectures that can implement different protection schemes. This section presents a generic view of the oBGP framework and its high-level functionalities that are guaranteed by each network compliant with the routing model.

To ease the task, the following paragraphs follow a "zoom in" logic where a global network view opens the path, leading to more detailed information about the meshing of sessions at the sub-plane level. The highest degree of depth is reached in the final part where things are considered from the point of view of an oBGP client.

4.4.1 Network view

For visual simplicity, the general architecture example considers a division into three sub-planes and makes abstraction of the virtual prefixes contained in these three sub-planes.

Figure 4.4 depicts an AS that addresses internal BGP routing by dividing the control plane information into three oBGP sub-planes. From the external point of view, the interaction between the oBGP network and the neighbor domains stays the same. The eBGP sessions ensure the redistribution of external routes towards the AS considered here.

There are though a few changes that the oBGP framework brings inside the network:

1. all border routers act as distributor nodes. The complete information received on eBGP sessions goes through the distributor nodes that are in charge of mapping the inbound advertisements to the corresponding sub-plane:

$$\forall v \in \mathcal{S}, v \in D \rightarrow \mathcal{S} = D$$

   Retrieving all external messages on the distributor nodes is essential to the reliability of the routing platform. Indeed, if the data is not mapped at the AS border, the oBGP nodes cannot coherently index the routes in the RIB by themselves, leading

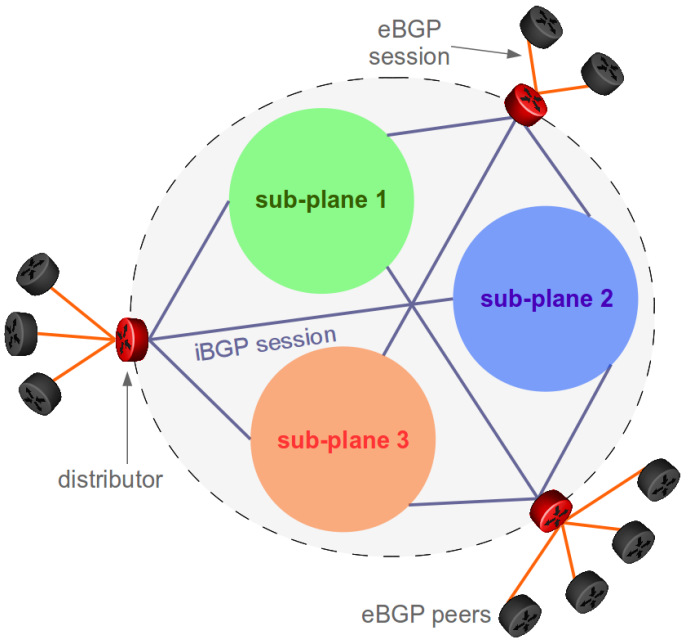


Figure 4.4 – Network view

to possible incoherent or incomplete routing tables. The oBGP nodes in charge of a certain sub-plane are unaware of how other sub-planes are distributed across the rest of the oBGP nodes. The only function that an oBGP node can perform on prefixes that it does not index is to filter them out, without any further concern about acting as a proxy and distributing them to another corresponding oBGP node.

2. distributors make sure that sub-plane attribution is coherent and correct, with no overlapping space (unless explicitly stated that a node carries two or more sub-planes). Obviously, all the distributor nodes share a common policy regarding the allocation of sub-planes to the oBGP nodes. Also, it is useful to enforce routing policies on the distributor nodes when it comes to filtering *bogon* or *martian* prefixes or other specific filtering of external prefixes based on economic or political reasons. For redundancy, in a practical setting, each border router that needs to distribute BGP messages according to the division into sub-planes will connect through classic iBGP sessions to at least two routers of each sub-plane.

3. no direct communication between the sub-planes. One of the major concepts in the oBGP framework consists of dividing the control plane into multiple subsets or sub-planes that are distributed across several equipments. The idea of splitting the routing information resides on the need to diminish the size of the routing table and to be able to compute routing decisions in an independent manner on each oBGP node, regardless of the other chunks of IP space. In oBGP, there is no need for an exchange between the different sub-planes. After splitting the RIB into sub-planes, the various BGP messages propagate only in the same sub-plane and there is full

isolation with regard to the rest of the sub-planes.

4.4.2 Sub-plane view

The communication between nodes of the same sub-plane must be very reliable and ensure a fast dissemination of the routing information. This is why the general design of a sub-plane resembles a lot the classic distributed architecture with route reflectors in an iBGP network. All messages are exchanged through iBGP sessions, similar to how it is usually done between the route reflectors connected as iBGP peers.

The highly distributed character of the protocol data gets diminished in oBGP because of how the information is concentrated at the level of the oBGP nodes. For this reason, redundancy must be taken into account when flooding the BGP advertisements to the multiple nodes of a sub-plane. If each distributor keeps only one session to each sub-plane, it becomes a single point of failure. This can turn out to be damaging in the context of a session error, resulting in the loss of a part of the external messages that can no longer reach the isolated sub-plane.

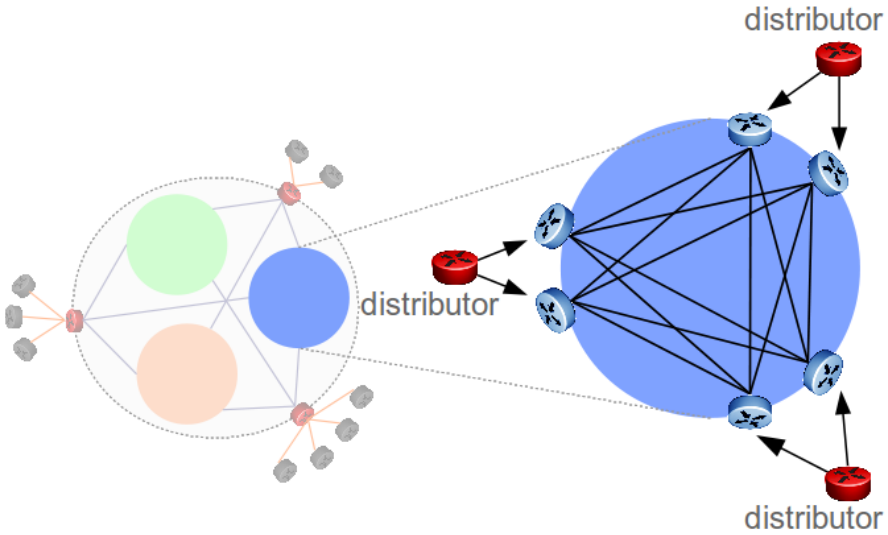


Figure 4.5 – Sub-plane view

A so-called “odd-even” meshing within a sub-plane would not allow the architecture to survive a specific dual failure case later explained in Section 5. So there is a need for “diagonal sessions” between oBGP nodes of each sub-plane. However, a session between two oBGP nodes connected to the same border router is of no use since they both receive the same external routes from the same edge router that acts as a distributor for both. The meshing will therefore be close to a full mesh, except for the sessions within each pair of oBGP nodes that share a distributor.

Figure 4.5 depicts an example of a sub-plane meshing for the network architecture previ-

ously presented. There are three distributor nodes that collect eBGP messages from the external peers and that spread them across three sub-planes. The detailed picture shows how each distributor node connects with two oBGP nodes from the sub-plane 2.

4.4.3 Client view

From the client point of view, the oBGP platform acts as a scattered route reflector. Each client router connects to at least two oBGP nodes of each sub-plane in order to retrieve the complete routing information that is available at the AS level. The propagation of routing information is handled through classic iBGP sessions, with an extra constraint: client routers do not send their internally generated routes directly to the oBGP nodes, but to the distributors. This way, the distributor nodes collect all the messages and manage the splitting and assignment of the entire dataset into sub-planes.

Another aspect to take into account is that the clients no longer keep any iBGP session between themselves, like in the sparsely meshed classic iBGP topology. There is no use for such inter-client sessions since each router receives the whole routing dataset from the oBGP nodes and is able to reconstruct the entire IP space without any additional information. The iBGP sessions between clients are removed, otherwise the correctness of the whole solution is undermined and scalability becomes negatively affected.

Figure 4.6 goes further into the details related to the general architecture. As previously, the network features three sub-planes that send their protocol messages to the client routers. In this particular setting, the client is redundantly connected to a pair of oBGP nodes in each sub-plane. It is equally communicating with two distributor nodes, to ensure resiliency. As seen in the figure, the sessions have different purposes: the iBGP sessions between the client and the sub-planes are used only for retrieving BGP messages, whereas the connections to the distributor nodes are necessary for sending the routes that are locally generated by the client router.

4.5 Gain through Design

The main goal of this chapter was to present a new framework for scalable iBGP routing. Some preliminary graph models offer some indications about the whole solution, leading the way to a more in-depth display of the core ideas behind oBGP. Subsequently, the oBGP concept is illustrated: an overlay responsible for performing the BGP decision process on behalf of the client routers within the AS. After exposing some of the major drawbacks in current iBGP in the previous chapter, we show how the oBGP routing platform solves some of these issues. We provide the design principles and advantages of oBGP then reveal the split algorithm that allocates the virtual prefixes to the corresponding sub-planes and a re-allocation method that can gracefully handle the dynamic reorganization of the virtual prefixes on the oBGP nodes. The final section presents the three views that make the global picture: the global view of the network, the more detailed view as seen from the

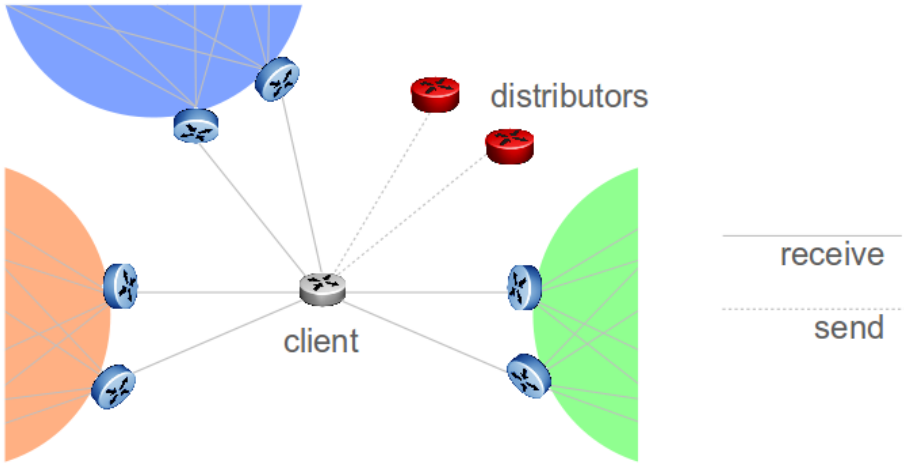


Figure 4.6 – The client sessions for receiving routing data from the oBGP nodes and sending internally generated routes to the distributors

sub-plane perspective and the most granular view, the client side.

The network architecture presented here is just the departure point for more refined and better adapted setups that can be built on top of the oBGP framework and supported by the general design. Since this chapter does not provide sufficient details for a real operational deployment, it is the objective of the following chapter to unravel real network setups and take into account aspects such as redundancy and to analyze the consequences of failures that could have an impact on the platform or its clients.

Chapter 5

Resilient architectures

The general architecture presented in the previous section is a model that serves for instantiating a certain practical architecture. The following paragraphs aim to offer real life examples of applicable architectures that take into account additional constraints related to resilience and survivability to failures. First, we take a look at how the networks of today are built in order to resist to failure scenarios. Further, we investigate two possible redundancy schemes that can apply on top of the oBGP framework. Finally, we analyze the response of the oBGP platform to some well-identified errors and failures.

5.1 Redundancy and replication

In the current route reflection architecture, a large network is usually divided into several clusters, for an easier management. The clusters are in their turn mapped to a different geographic distribution according to Points of Presence (PoP). A PoP is a region where a network operator has decided to place network equipment. For example, in Fig. 5.1, PoP 1 can identify a site such as Paris and PoP 2 could be Lyon. Each of the five clusters (A, B, C, D, E) are distributed over both PoPs.

Common engineering rules dictate that each client has to receive the routes from at least two route reflectors working together as primary and backup, for redundancy purposes. Here, a client has one route reflector in each of the two PoPs which allows it to be resilient in case of site failure caused by flood, fire, power outage, etc. In this conventional setting, each RR in the network carries the complete set of routes received from its clients and from its BGP peers, the other RRs in the iBGP mesh.

The same robustness logic is applied to the oBGP architecture where the different sub-planes need to be replicated in the network. The information from a sub-plane is copied on several nodes; note however that a client is not connected to all the possible oBGP nodes that carry a given sub-plane. As seen in the Section 4.4, redundancy happens not only at the sub-plane level, but is also enforced for each individual client. A client can always retrieve routing data corresponding to a certain sub-plane from at least two oBGP nodes.

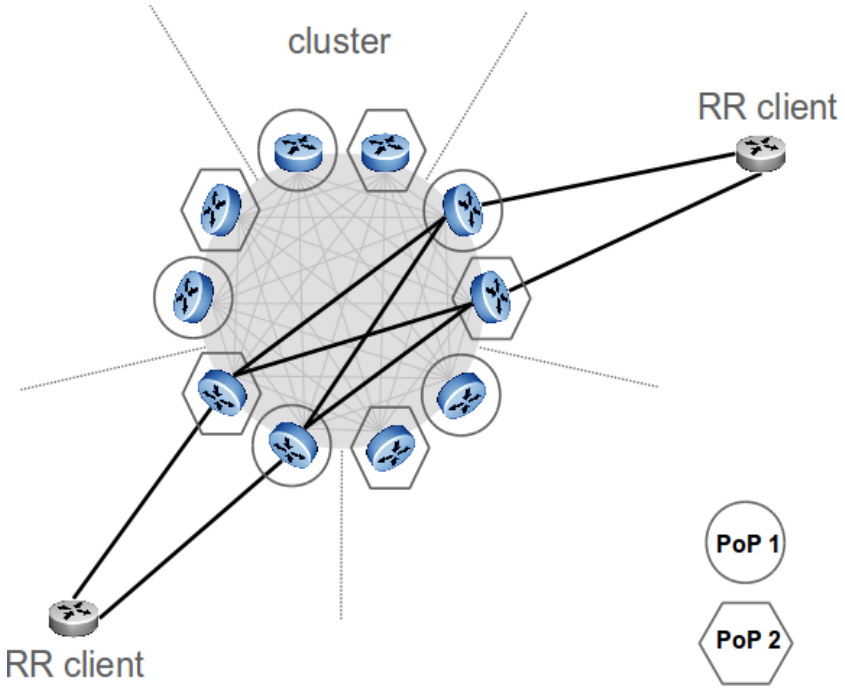


Figure 5.1 – The current iBGP design with PoPs, clusters and redundancy RRs

Another parameter to take into account is that the client router sends its locally generated routes to at least two distributors that later dispatch the data to the corresponding sub-planes.

Protection is usually good enough in today's networks with a simple duplication of the architecture: install two route reflectors instead of just one, in order to survive simple failures. In critical infrastructures the components may be tripled, but the service provided by the network considered here does not justify such an investment. We therefore define k as the replication parameter and for facility, we consider the minimal value $k = 2$ in the following examples. With $k = 2$, it means that a client receives BGP route advertisements for each sub-plane from two oBGP nodes of that particular sub-plane and that, in its turn, the client sends the internally generated routes to two distributors.

Although other architectures are compliant with the oBGP framework and can be successfully applied on top of the general architecture, we later examine two particular schemes for allocating sub-plane datasets to the oBGP nodes.

5.2 The 1:1 redundancy scheme

In theory, an oBGP architecture with n sub-planes will distribute $1/n$ of the total number of routes to each node. When taking into account redundancy, the same content is copied k times, meaning that a node holds a fraction of k/n of the global RIB. This distribution

is needed to cover single failure cases when a node carrying a sub-plane is down. With two replicas ($k = 2$) the IP space allocated to the failing sub-plane can still be reached on the second copy located on another physical equipment.

The 1:1 redundancy scheme relies on the idea of sharing a single oBGP node between two sub-planes, acting as a primary for one sub-plane and as a secondary for the other sub-plane. In this setting, the most straightforward implementation of an architecture with three sub-planes would be to group each two adjacent sub-planes on a given oBGP node. The elementary approach for $n = 3$ is to consider a distribution according to three equally sized sub-planes s_1, s_2 and s_3 across the oBGP nodes N_i: $N_1 = (s_1 s_2)$, $N_2 = (s_2 s_3)$ and $N_3 = (s_3 s_1)$. This type of redundancy scheme works in a circular manner and thus can be successfully applied for an odd number of sub-planes.

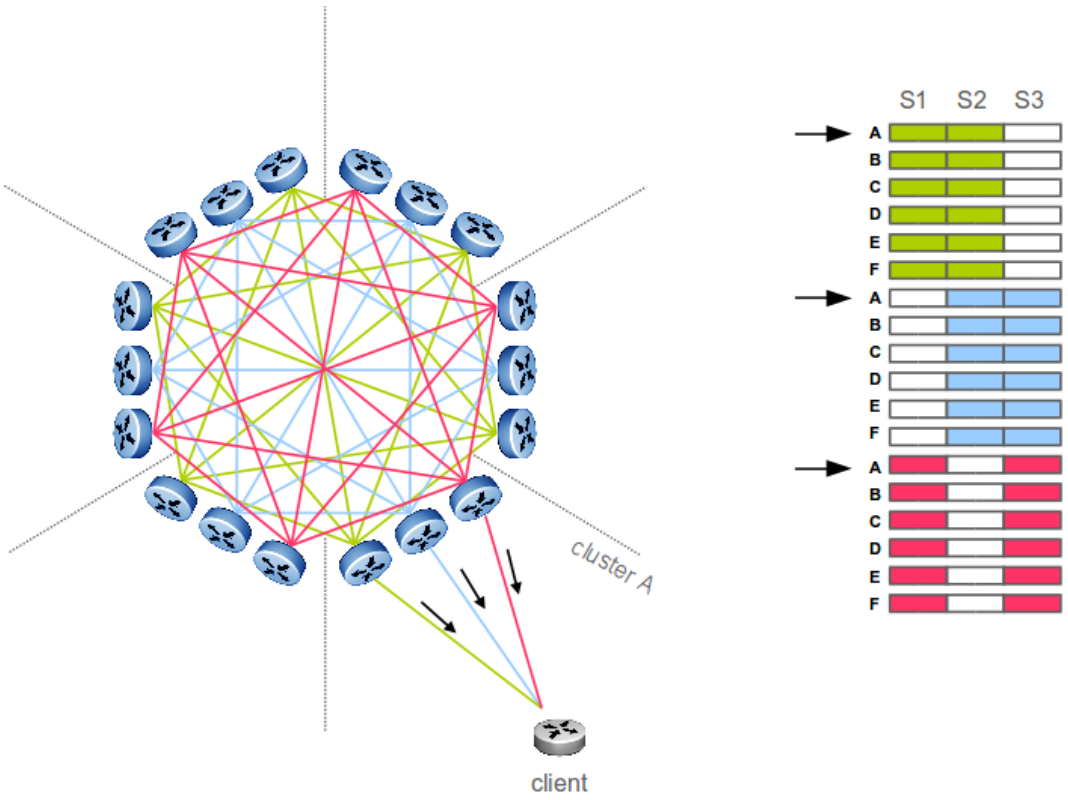


Figure 5.2 – The architecture for three sub-planes in the 1:1 redundancy scheme

Let us consider the network in Fig. 5.2 as a simplified example that implements such a redundancy scheme. The network is divided into six clusters: A, B, C, D, E, F. Each cluster contains three routers that are in fact the oBGP nodes in charge of flooding the BGP information according to the sub-planes. In each cluster, the nodes N_i have a specific "color" that identifies the particular combination of sub-planes [1] they are storing. The

1. For example, the green mesh identifies nodes carrying $(s_1 s_2)$, blue identifies the oBGP nodes in charge of $(s_2 s_3)$ and pink is for the remaining subset $(s_3 s_1)$.

oBGP nodes carrying the same combination of sub-planes (e.g., $(s_1 s_2)$) are fully meshed and redistribute among each other only the information corresponding to these two sub-planes. This arrangement leads to a routing control plane divided according to the three isolated "colors".

The client in cluster A shows how the reception of the entire dataset of routing messages happens: a client is connected to all the "colors". The right side of the figure offers an abstraction of the reachable routing space. For the client in cluster A, each oBGP node delivers the BGP updates for two sub-planes. If one of the oBGP nodes in cluster A happens to go down, this simple failure can be overcome since the client can reconstruct the entire routing table from the other two oBGP nodes. For example, let us assume that the node N_3 in charge of $(s_3 s_1)$ is unreachable. The client still receives information from $N_1 = (s_1 s_2)$ and $N_2 = (s_2 s_3)$ and manages to rebuild the initial complete data.

Although the 1:1 redundancy scheme has the advantage of being compatible with an odd number of sub-planes, this architecture is not able to handle a specific case of double failure that we hereby detail.

The scenario described in Fig. 5.3 is the following: the distributor on the left side is situated in cluster C and connects to the three oBGP nodes representing the different "colors". One hop away, the oBGP nodes situated in another cluster forward the information received from their oBGP peers to the client. In case of a double failure involving one oBGP node in cluster C and a node of a different color in the cluster of the client, the complete dataset is no longer available: s_2 from cluster C is lost.

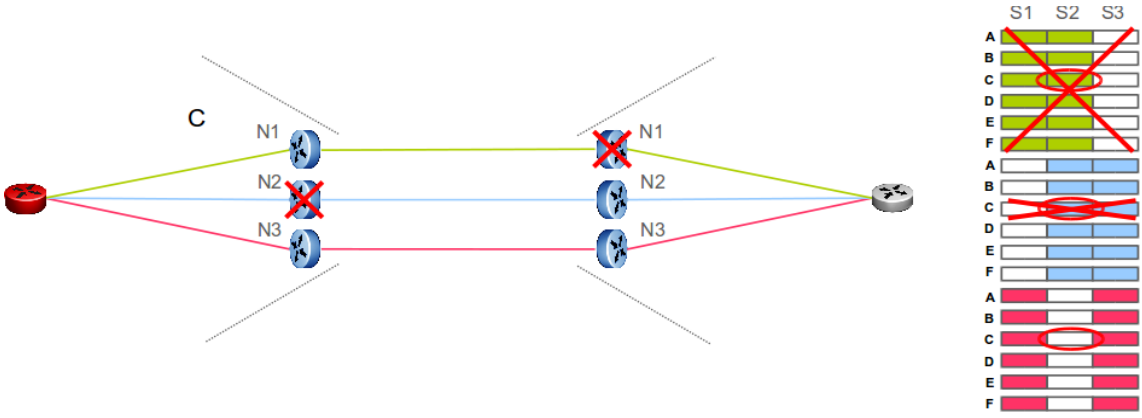


Figure 5.3 – A double failure in the 1:1 redundancy scheme

The oBGP node N_1 feeding routes to the client fails and thus hides all the information carried by this "color" — $(s_1 s_2)$. However, we have seen that the architecture can withstand a single failure case and the client can reconstruct the entire dataset from the remaining oBGP nodes. What happens if there is another failure affecting a node on cluster C? The client has already lost N_1, so it relies on N_2 and N_3. Suppose N_2 in cluster C fails also, then the client no longer has access to the routes in sub-plane 2 coming from cluster C.

This shortcoming can be fixed with a different distribution of sub-planes on the oBGP nodes and a more redundant meshing that introduces diagonal sessions between nodes of the same color. The 1+1 redundancy scheme explains in more detail the concepts of the workaround.

5.3 The 1+1 redundancy scheme

The 1+1 solution is funded on the idea that the specific dual failure case can be handled with a different distribution of the sub-planes and a more robust meshing of sessions. 1+1 comes from the fact that one equipment (or oBGP node) is necessary for the nominal task and an identical copy, the +1, is responsible for taking over in the case of a network event.

As seen in the previous 1:1 architecture, for $n = 3$ sub-planes and redundancy parameter $k = 2$, each oBGP node needs to keep a fraction equal to 2/3 from the entire set of routes. Since the gain is not that significant (the routers store only 1/3 less routes than before) and the 1+1 scheme cannot handle an odd number of sub-planes, we look at the next even value $n = 4$ that should bring more relief.

The new division into subplanes consists of a duplication of the existing route reflection schemes. For a 4 sub-plane network, the organization would be the following: $N_1 = (s_1 s_2)$, $N_2 = (s_1 s_2)$, $N_3 = (s_3 s_4)$ and $N_4 = (s_3 s_4)$, thus halving the number of routes needed on each of the oBGP nodes.

The diagonal meshing required for survivability in case of a double failure of oBGP nodes representing two different "colors" is shown in Fig. 5.4. As seen, backup sessions need to be added, making it thus possible to put up with the double failure. The diagonal sessions insure that whenever an oBGP node fails, it does not take down the corresponding sub-plane. In the explicit configuration depicted here, the client can retrieve the entire routing space and reconstruct the initial RIB.

If node N_1 from the same cluster as the client fails, then the information can still be recovered from the remaining nodes. Indeed, because of the diagonal meshing, N_2 receives now all the information from cluster C and is able to send it to the client. If an additional oBGP node goes down, here N_3 from cluster C, then the client will perceive no impact because the node N_4 from cluster C is still connected to the nodes N_3 and N_4 from the client cluster. The abstraction of the client RIB-In on the right side of the figure shows that all sub-planes, s_1, s_2, s_3 and s_4 are received from at least one source.

How does this meshing translate into practice? Fig. 5.5 renders a possible architecture for an oBGP platform distributed across an entire network. This target architecture is based on an division of the routers across six PoPs and three clusters. The global RIB is split into 4 sub-planes that are mapped on two "colors" which can be basically reduced to two distinct sub-planes each accounting for half of the RIB and each backed up by an exact replica. We thus further refer to the different fractions of the RIB as being sub-planes: the previous "colors" become sub-planes by federating $(s_1 s_2)$ into a single sub-plane s_1 and merging $(s_3 s_4)$ into s_2.

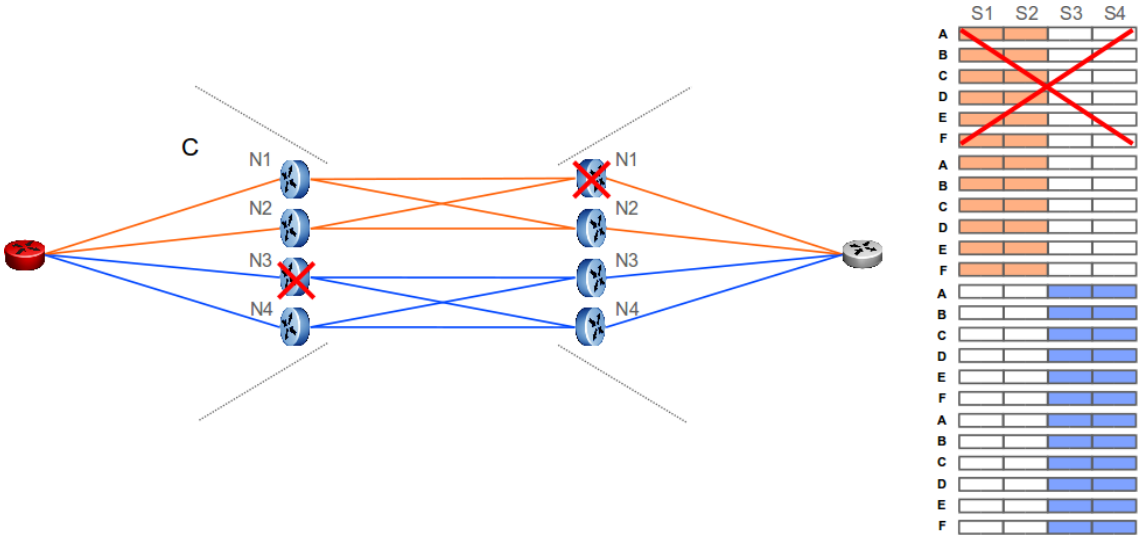


Figure 5.4 – A double failure in the 1+1 redundancy scheme

In the current 1+1 architecture, the client receives $k = 2$ copies of the same routing information. For example, the nodes N_1 and N_2 both deliver the same sub-plane, meaning that this redundancy scheme is a mere duplication of the current route reflection architecture. We introduce thus the concept of primary and secondary oBGP nodes, similar to the previous nominal and backup route reflectors.

As shown by the figure 5.5, a client receives data from each of the sub-planes through a primary and a secondary session. Since the oBGP nodes in a given sub-plane will store the same routing information, when a client receives two (almost) equivalent routes, the BGP decision process will reach the last step where the tie breaking will be solved based on the sender's IP address. For a given cluster of a sub-plane, the client will receive the exact same routes from the two oBGP nodes and will end up choosing as best the route advertised by the node with the lowest address. Based on this fact, it is interesting to setup a convention by labeling as "primary" the oBGP node that will always be preferred and as "secondary" the node with the highest IP address.

It is useful to have the two redundant oBGP nodes serving the same sub-plane located in different PoPs, offering thus an extra advantage in case of site failure. The next section offers more details about specific failures and how the oBGP architecture handles them through corresponding design constraints.

5.4 Failure cases

Concerning error scenarios, the construction of oBGP is redundant as to avoid single points of failure. Depending on the deployment topology (e.g. an oBGP platform per geographical region, per Point of Presence or per AS), the failure impact varies correspondingly. We

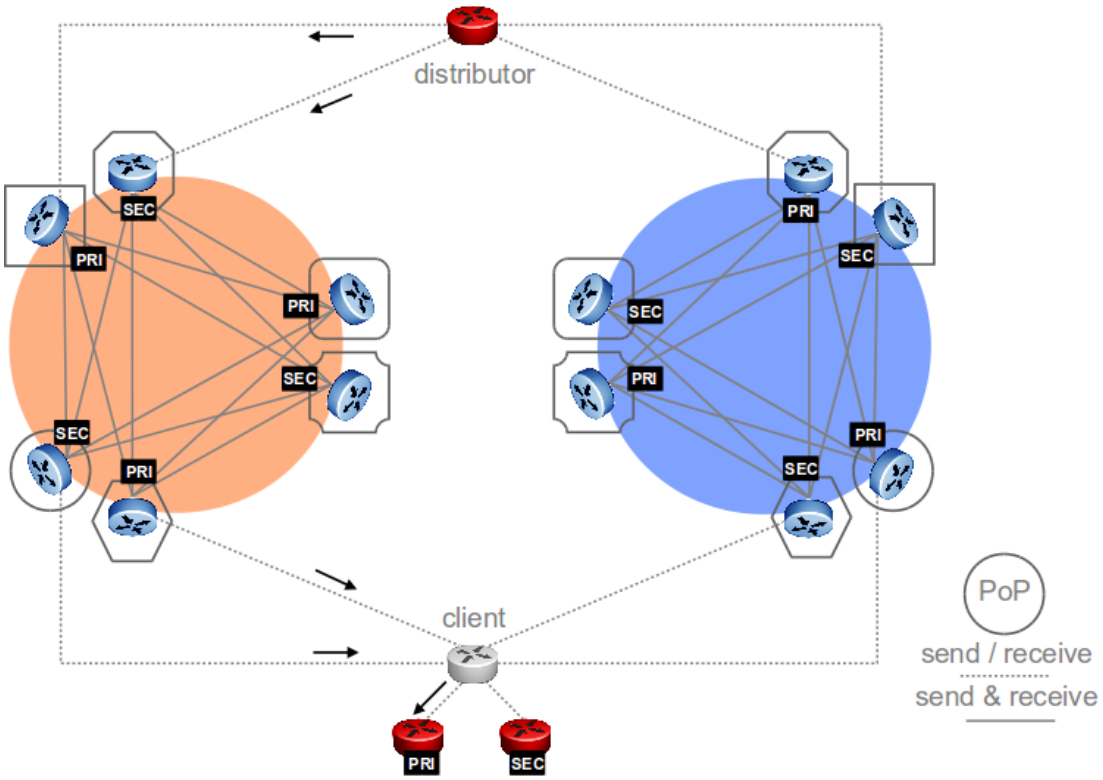


Figure 5.5 – A global view of the 1+1 architecture implemented with two sub-planes.

hereby concentrate on the architecture presented so far, where the oBGP platform is deployed in the entire network and the emphasis on redundant meshing is within each of the sub-planes. In the following sections, let us consider some specific examples of simple and multiple failures.

5.4.1 Node failure

As justified in 5.3, a network that deploys the oBGP model should be shaped according to the concepts presented in the 1+1 design, with redundancy in mind. Due to the routing data being stored in a redundant manner with the sub-planes replicated on k oBGP nodes, both the 1:1 and 1+1 architectures can take a simple node failure. On the other hand, the simple meshing of three "colors" in the 1:1 scheme turns out to be insufficiently robust in cases of double node failure, while the 1+1 architectures fixes this shortcoming.

Since the oBGP platform is in charge of disseminating control plane data, it is critical for the architecture to be able to survive extensive error scenarios. Indeed, the propagation of routing data should not be impacted by unexpected network events. The idea behind the 1+1 type of session meshing is to guarantee that the information can flow continuously.

Let us further take a look at how a node failure impacts the clients of the oBGP platform.

We next investigate the direct consequences of such simple and dual node failures on the behavior of BGP at the client level.

Failure of a primary oBGP node

When a primary oBGP node fails, convergence will occur for all the clients of its cluster and for the other four oBGP nodes in the two other clusters of the sub-plane. The clients are impacted because they are directly fed by the primary node and the other oBGP nodes in the mesh need to withdraw the routes received from the failing node and that they had been forwarding to their own clients. All these client routers switch from the routes that were previously advertised through the faulty node to the routes that are advertised by the corresponding secondary node, part of the same cluster of the same sub-plane. Since the new routes are actually the same, the only difference being the identifier of the advertising router, convergence should theoretically guarantee that there is not any traffic loss. However, this hypothesis should be assessed against the vendors' BGP implementation during testing.

One of the possible consequences of switching from the primary to the secondary oBGP node is the duplication on the remote oBGP nodes of the routes sent from the distributor to the former primary–secondary pair. Indeed, duplicate routes are a phenomenon observed in operational networks, specific to certain implementations of equipment vendors that do not filter such updates (for more details, see Appendix).

Failure of a secondary oBGP node

When a secondary oBGP node fails, the routes it was advertising are deleted on all clients of its cluster and on the four oBGP nodes of the two other clusters, in the same sub-plane. Taking into account that those routes were not the preferred routes since only the primary routes are selected by the clients, this simple node failure should have no further consequences.

Dual node failure

What happens though in the case of a dual node failure? We have seen previously that the 1:1 architecture does not survive a specific case where the dual failure impacts nodes belonging to different clusters and to different "colors". This is why we introduced the diagonal meshing required in the 1+1 architecture. The improved design can withstand this specific dual failure thanks to the sessions between the oBGP nodes handling the same sub-plane.

We continue a more detailed analysis of the different types of dual node failures in the next paragraphs. Three possible scenarios are depicted, emphasizing the impact of the failures on the client nodes.

An oBGP node in a given sub-plane & any node in the other sub-plane:

Since both sub-planes are totally independent and they each provide internal redundancy, a failure of an oBGP node in a given sub-plane and a simultaneous failure of a node in the opposite sub-plane will not result in service disruption.

When looking at Fig. 5.6, we can observe that a connection is maintained between any possible distributor–client pair. All distributors are thus able to communicate to all oBGP clients the BGP updates received on the external BGP sessions from other peer ASes.

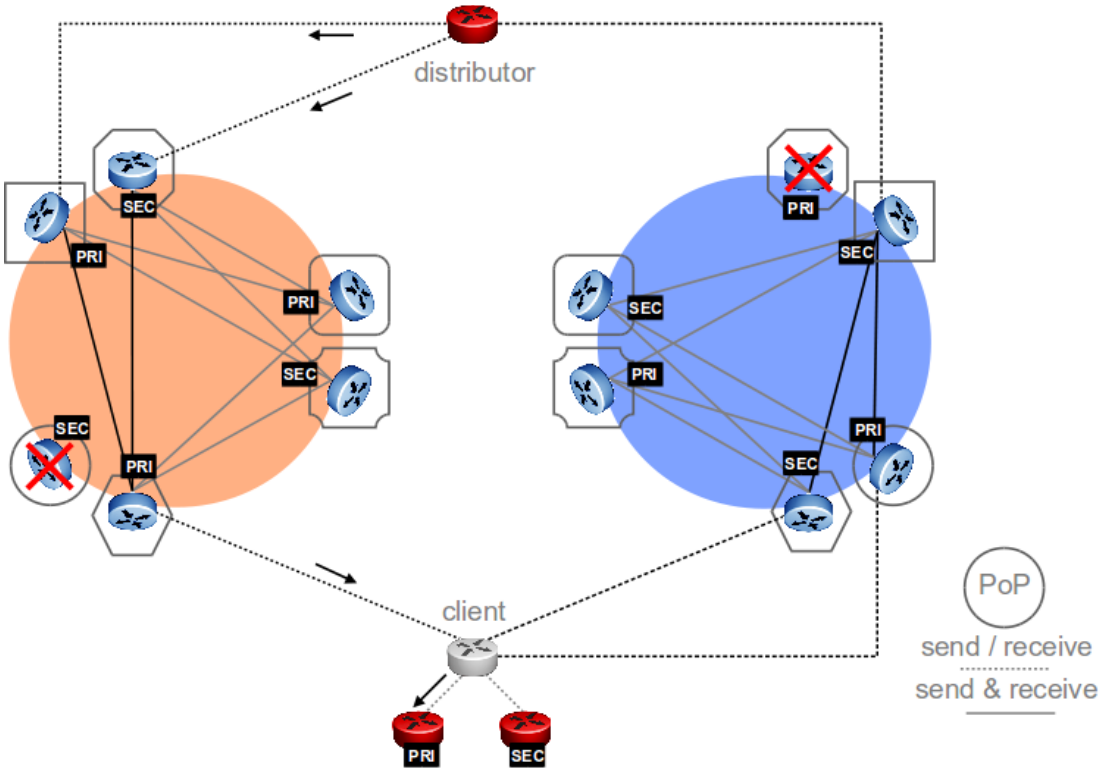


Figure 5.6 – Double node failure in the 1+1 architecture, failing nodes located on different clusters of different sub-planes. The number of sessions is diminished in both sub-planes.

The actual impact of this kind of session loss depends on whether the affected oBGP nodes are primary or secondary (refer to previous paragraphs for corresponding single node failure cases).

Both nodes in the same sub-plane, but in distinct clusters:

The chosen 1+1 architecture is redundant to another case of dual node failure where the two oBGP nodes are situated in the same sub-plane, but in different clusters. Although intuitively we could anticipate that the network is affected by this type failure because all errors occur in the same sub-plane, the example in Fig. 5.7 shows that communication continues between any possible pair of distributor and client. Again, due to the diagonal meshing between the oBGP nodes in clusters of the same sub-plane, this design proves to

be a viable solution when it comes to network survivability.

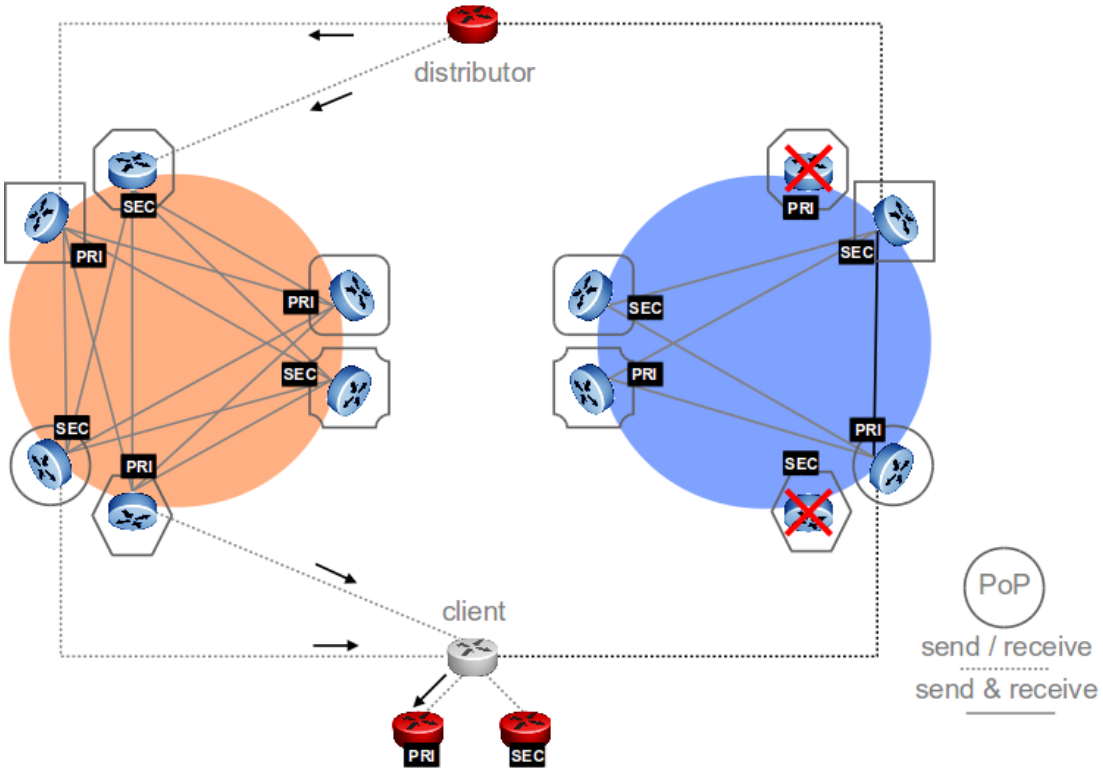


Figure 5.7 – Double node failure in the 1+1 architecture, both failing nodes located in the same sub-plane in distinct clusters. Only one sub-plane is impacted and the number of sessions between the two clusters is severely reduced, a single session still remaining.

From the client point of view, the impact is more significant depending on the type of the failing oBGP nodes: primary or secondary (again, refer to the discussion in previous paragraphs).

Both nodes in the same sub-plane and the same cluster:

The only non-survivable dual oBGP node failure case is when a client loses both nodes that are feeding routes of a specific sub-plane. This kind of failure is the equivalent of losing both the primary and the secondary node in a given cluster. This is why these two oBGP nodes in charge of one sub-plane are required to be located on two distinct sites to provide site redundancy (see next section).

For a complete session loss between a client and one sub-plane, there is a short interval (BGP re-convergence time) during which the disconnected router forwards packets according to a stale FIB, causing possible sub-optimal routing.

5.4.2 Site failure

A minimum of two physical sites is required for the architecture to resist to a full site failure caused by power issues, cooling problems, fire, flood, etc. The example in Fig. 5.8 actually depicts six distinct sites distributed into three clusters. An important aspect is that redundancy is provided within each colour, not across: both colours need to survive in order to keep all routes flowing through the control plane.

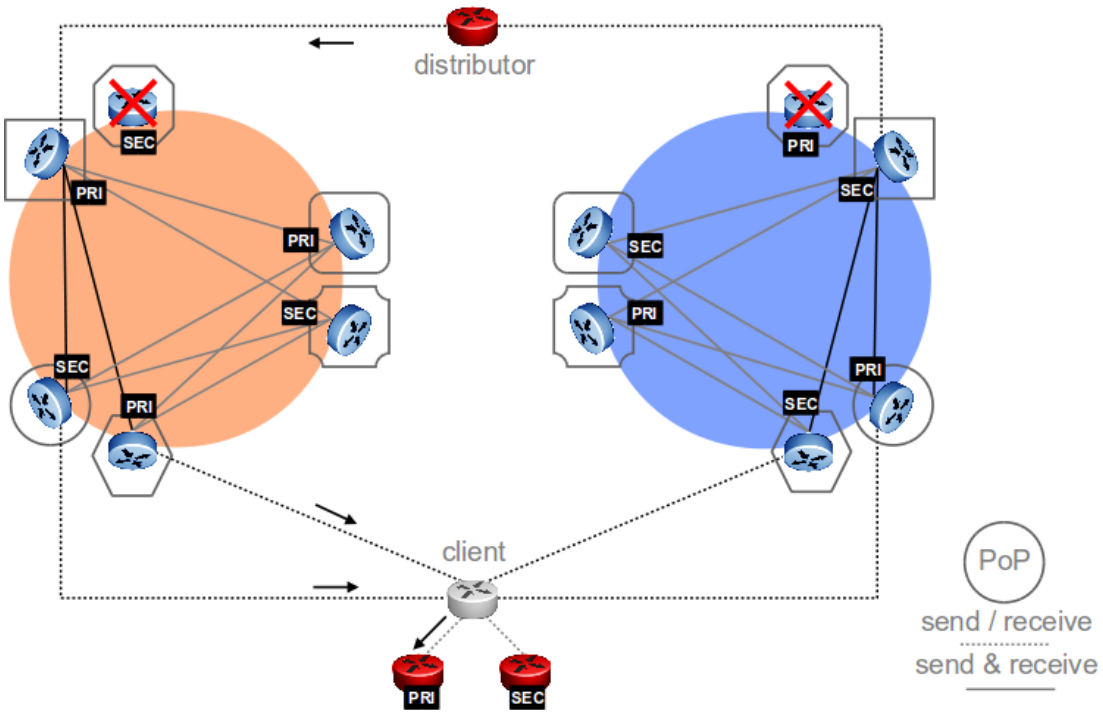


Figure 5.8 – Site failure in the 1+1 architecture

Because all clients of a given cluster will rerun the BGP selection only when their primary oBGP node fails, but not when their secondary node fails, any given cluster should collocate the primary node of a given sub-plane with the secondary node of the other sub-plane. If that particular PoP should become unavailable, clients will rerun the BGP selection only for half of their BGP routes, more precisely for the routes belonging to the sub-plane hosted by the primary node located on that failing site.

Another option to consider would be to have the oBGP nodes configured with two loopback addresses and serve half of the cluster's clients as their primary node and the other half as their secondary node. However, this engineering trick does not bring anything but free complexity: it only changes the number of clients impacted by a node failure, not the number of BGP routes that require rerunning the BGP selection algorithm; since this is a distributed process, it does not matter if twice as many clients run the BGP selection half less often or if half as many clients run it twice more often.

The particular network layout with six sites and three clusters depicted in Fig. 5.9 can survive several dual site failures. In fact, it can resist all dual site failures that do not include a primary and secondary node of the same cluster in the same sub-plane: $N_1 + N_2$, $N_3 + N_4$ and $N_5 + N_6$ cannot be survived.

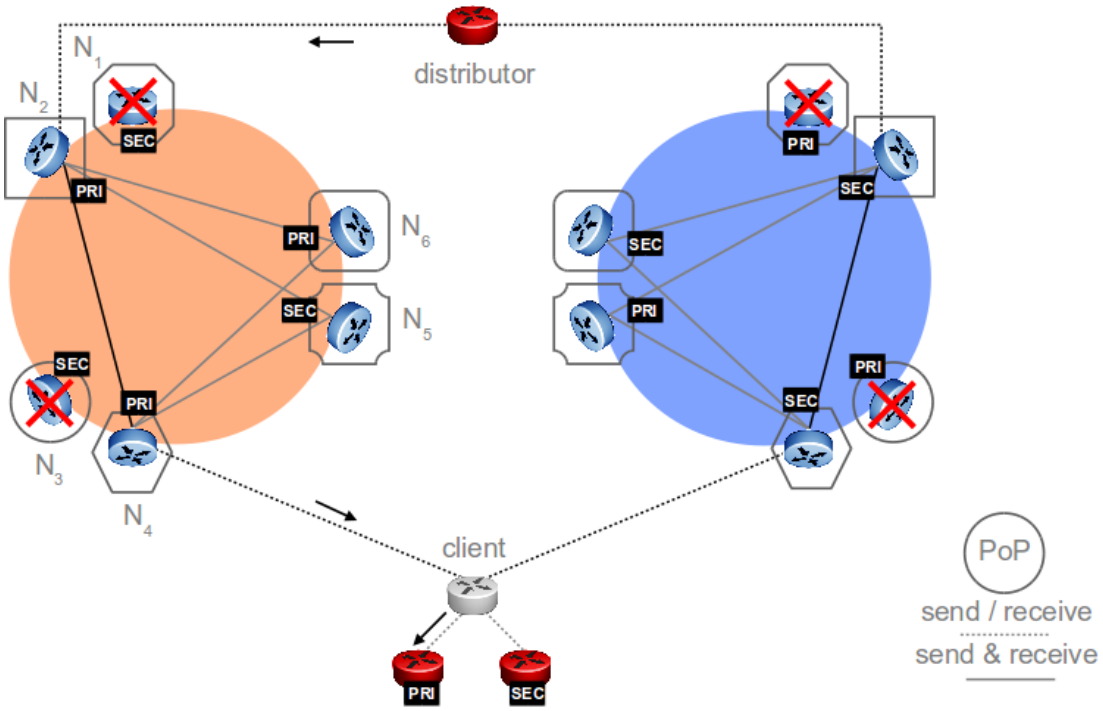


Figure 5.9 – Dual site failure in the 1+1 architecture: $N_1 + N_3$

5.4.3 Other failures

The 1+1 architecture can actually survive some triple failure cases, but we will not go into details since they are highly unlikely. On the other hand, another failure can occur in a key element of the oBGP platform: the distributor. How does the network behave when undergoing a **distributor failure**?

Keep in mind that the distributor has a double role in the oBGP framework: it spreads external routes received from neighboring ASes on eBGP sessions inside the considered network implementing oBGP and it also collects all internally generated routes from the client nodes that no longer redistribute these routes by themselves. The distributor becomes thus a central point in the flow of control plane information.

From the first perspective, when a distributor is unavailable for feeding external routes to the inner sub-planes, the network reacts the same way as in the case of a simple ASBR failure in a classic BGP network. If the node is unable to reply with *keep alive* messages, the sessions are automatically removed and communication is no longer maintained between

the border router and its BGP peers. This action has several consequences: the external peering routers no longer send new BGP messages and the ASBR is considered as no longer a part of the BGP topology; the same behavior is mirrored within the internal network where the iBGP peers stop receiving any messages and thus delete the ASBR from the list of connected peers. When translating this state into an oBGP platform, it means that the sub-planes get an incomplete view of the global RIB dataset, unless the other distributors send equivalent routes that are fully redundant with the missing set.

The second role of the distributor is to gather the routes generated internally by the oBGP clients. Not all routes come from the external peers and in some networks there is an important amount of routes that are originated by the routers in the network. If a distributor fails, it can no longer receive announcements related to routes coming from the oBGP platform clients. However, this is not a serious issue since every client node has two distributors to which it can send information about its own routes. Ideally, redundancy is ensured at the physical level too, with clients connected to primary and secondary distributors located in different PoPs.

5.5 Final considerations

This chapter presents the current redundancy techniques and best common practices enforced by network operator in BGP networks. Starting from doubling the route reflection architecture, some of the existing concepts are leveraged in the 1:1 and 1+1 architectures proposed for the oBGP framework. After more thorough investigation, it turns out that the 1:1 redundancy scheme is not fit to handle some cases of double failure, so the proposal evolves towards the diagonal meshing within a given sub-plane, as specified in the 1+1 setup. Also, this chapter looks at different failure scenarios, either on specific elements of the oBGP framework such as primary and secondary oBGP nodes or even the distributor. Moreover, the architecture is built in such a way as to withstand certain site failures without any major impact on the clients, as presented in section 5.4.

The possible meshing and disposal of oBGP elements presented here are only a few specific solutions among many others that can be implemented on top of the oBGP model. Some of the additional constraints are imposed in order to obtain better practical results, such as the non-overlapping of sub-planes and the fact that one node carries only one specific sub-plane that later allow the 1+1 architecture to survive specific dual failure cases. The oBGP framework in itself allows for more flexibility, but the chosen architectures are here presented as a proof that the concept is applicable.

Chapter 6

Practical oBGP

The oBGP platform requires modification to the BGP routing software, namely in the control plane part. Since such changes are very hard to introduce into marketed products like vendor routers, we rely on a software solution. In order to present an achievable implementation, this section provides a workaround for the intended architecture: the use of control plane virtualization within a remote oBGP platform that runs on commodity hardware. First, the dVirt test platform is introduced, along with its architecture, implementation details and the typical usage parameters. Leveraging the existence of the dVirt testbed, the final part describes the precise connectivity within the PoPs and shows how the oBGP nodes an the client routers work together. The presented setup can be used for simulating the oBGP architecture.

6.1 The dVirt test platform

Transitions to new routing conditions can be complex and unexpected issues may arise. Despite the plethora of software tools available to model and experiment BGP configurations, there is no dedicated tool offering an automated testbed that allows for accurate simulations and interactions with the underlying protocol layers. dVirt federates multiple functionalities into a flexible tool enabling the user to automatically deploy and evaluate a network.

The ultimate goal for dVirt is to be able to "clone" a full ISP network on top of a virtualized infrastructure running on a smaller number of servers. dVirt aims to reproduce the actual events in a network, experiment with the real configurations and addressing schemes. The proposed framework can be used to check correctness (e.g., avoid oscillations), compare convergence time for different setups or even implement and test additional features on top of the existing protocol stack.

dVirt relies on virtualization techniques and routing tools: we use the Xen hypervisor and virtual machines to represent a large number of network equipments and we put to work the Quagga software routing suite for simulating the multiple routing instances. We

take advantage of sbgp, a simple BGP4 speaker and listener, to mimic the behavior of neighboring ASes by injecting external route advertisements into the edge routers of the simulated network.

dVirt is a software library written in Python for automatically deploying a given BGP network. It can be controlled from a single machine through simple and flexible inputs that avoid the individual provisioning of resources and configuration of routing protocols. The typical use of dVirt is to simulate the entire topology of one or more large ISP networks and incorporate realistic configurations.

6.1.1 dVirt Overview

Compared to previous tools, dVirt is a heavier simulator but removes many barriers thanks to its full customization. dVirt relies on open-source software and runs real operating systems, it supports many real network conditions such as addressing, multitasking of the routing processes and inter-protocol interactions.

dVirt creates an Ethernet topology of virtual machines (VMs) running on top of hypervisors that are mutually reachable at the IP layer. dVirt emulates virtual point-to-point Ethernet connectivity between pairs of router interfaces using virtual switches provided by the Open vSwitch software. Open vSwitch enables virtual Ethernet connectivity between two routers located either on a single or two distinct hypervisors by encapsulating Ethernet traffic inside GRE tunnels.

The OSPF and BGP topologies are automatically configured to enable full reachability inside each AS and setup (mono-hop) eBGP sessions. Routers are running the Quagga routing software with OSPF and BGP daemons to simulate the demanded network. External neighbors of the deployed topology are emulated using sbgp software instances running in one or more additional VMs.

dVirt emulates the full protocol in each router and allows the study of the BGP protocol dynamics by directly running Quagga with all the implemented features. The tool can also be used to deploy modified versions of the Quagga software and therefore handles many routing protocol testing scenarios.

dVirt simplifies the instrumentation of experiments conducted according to a simulation scenario. The user can directly use python bindings to execute existing or user-defined functions. Network monitoring functions run on routers and provide information about the state of the BGP routers, allowing thus to obtain exact measurement data.

6.1.2 dVirt Management Network

To allow permanent communication between the user and the routers, dVirt separates the infrastructure in two distinct networks: a management network for remote access and a test network for the actual simulation. Each VM has a local IP address configured on the interface attached to the management bridge defined on each hypervisor as seen in Fig.

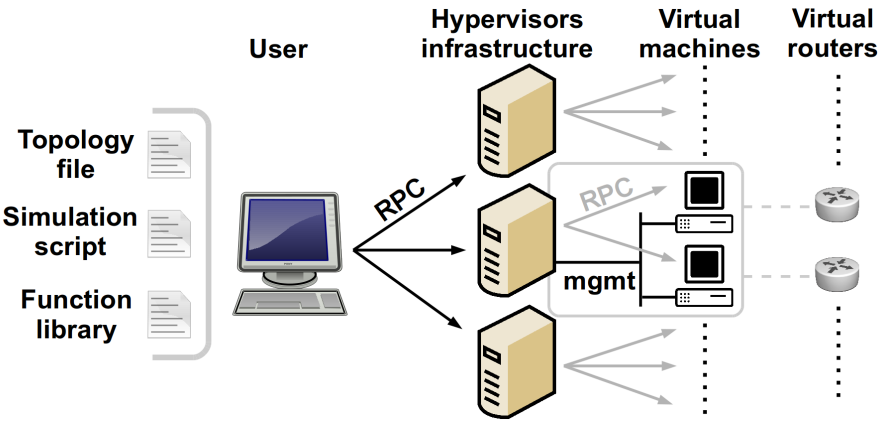


Figure 6.1 – An overview of the dVirt components: the user interacts through the RPC with the remote hypervisors and the corresponding virtual machines.

6.1.

To interact with the remote routers, dVirt provides two libraries to exchange files and do remote calls: SSH and RPC. The SSH library allows to exchange files and send commands to the VMs over an ssh connection with text output.

The RPC (remote procedure call) library enables the creation of a TCP tunnel between the user and any hypervisor in order to execute requests directly on the hypervisor with a simple function call. The output is a python object that is serialized and sent back to the user over the TCP session. The dVirt RPC library has the particular ability to allow transparent execution of RPC requests from the user to a VM or router. The RPC resorts to the hypervisor as an intermediate point that forwards the request in an embedded call directed to the virtual router, as shown in Fig. 6.2.

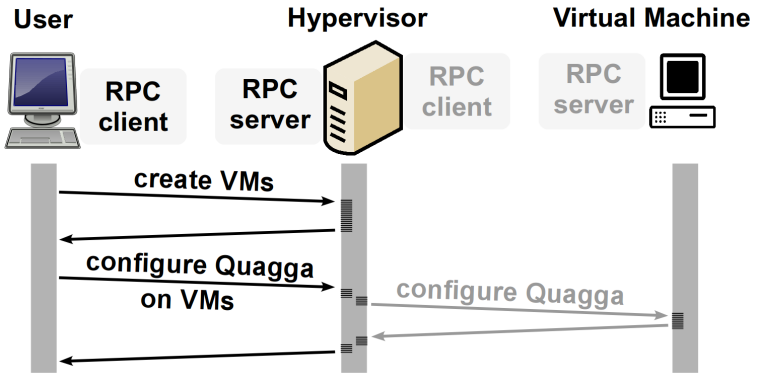


Figure 6.2 – The user launches a request that can be executed either by the hypervisor or the virtual machines. Note that for the calls on VMs, an embedded request is forwarded by the hypervisor to the corresponding VM.

dVirt comes with a set of pre-defined functions for the RPC server-side for interaction with hypervisors, VMs, routers and their installed software. The user can easily improve the existing library by adding new functions to the python files in the library. During the next deployment, dVirt will automatically update the RPC library of each hypervisor and each VM making available the new user-defined functions.

6.1.3 Virtual Routers and Virtual Ethernets

Virtualization provides a way to run multiple operating systems called virtual machines (VMs) on top of a single hardware platform. This means that each of the separate virtual machines can run a distinct version of software and different applications, while having concurrent access to the hardware resources, as seen in Fig. 6.3. The multiple virtual machines are logically isolated and can each act as a simulation of a classic standalone machine running one operating system on top of dedicated hardware.

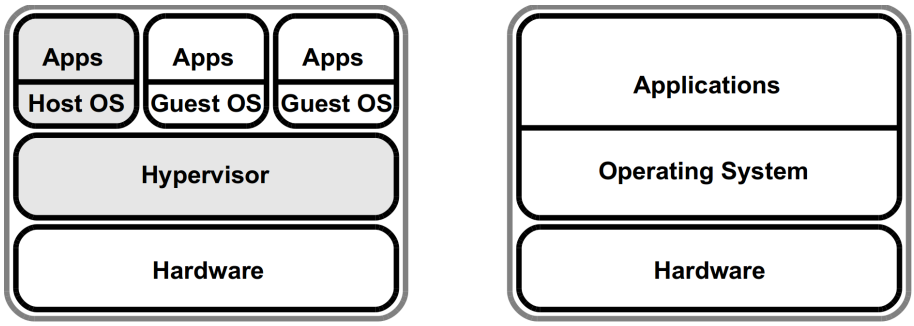


Figure 6.3 – A comparison between two systems: the left one runs three virtual machines on top of a hypervisor compared to the right one running a classic operating system

More precisely, dVirt relies on the Xen open-source software to achieve the virtualization of x86 CPU architectures. The Xen hypervisor allows one physical machine to run multiple router instances by acting as an abstraction layer to the bare hardware and isolating the virtual machines from the external networks.

Xen handles the concurrent access of the VMs to the resources and manages the execution of the guest OSes. In Xen terminology, *Dom*0 is the first operating system that boots automatically and receives privileged rights regarding hardware access and management. From the *Dom*0, the administrator can launch new virtual machines, called *DomU*s and manage all the existing guest machines. dVirt uses *virsh* management interfaces of the libvirt [Libvirt - The virtualization API] API to create machines from a customizable xml file where memory and CPU allocation can be changed for any router.

By default, each VM in dVirt is a router or it hosts sbgp software instances to simulate external BGP neighbors. Each VM has a dedicated SWAP filesystem, a CPU, a dedicated memory of 512 MB, and a generic pre-installation of the Linux Debian Lenny operating system (distribution 2.6.26-2-xen-686) customized with the required software such as

Quagga or Python. In the *Dom*0 of each hypervisor, dVirt configures the management network through *virsh* and uses Open vSwitch to emulate point-to-point links in the experimental network. Quagga runs as a regular application on each virtual machine and takes over the kernel routing of the virtual machine. As seen in Fig. 6.4, for a pair of source-destination routers, two scenarios are possible: if the routers run on the same hypervisor, they interconnect through a dedicated virtual switch (e.g., R2 and R3 linked with grebr3 on hypervisor B); otherwise the two ends of the link are on distinct hypervisors and dVirt needs to define two Open vSwitches, one for the test interface of each router. The traffic between the routers is then transparently forwarded inside a GRE tunnel (e.g., R1 and R2 connect respectively to grebr1 on hypervisor A and B).

A GRE tunnel is setup between two hypervisors only if two distant routers share a point-to-point link. Multiple links between the same two hypervisors can take the same tunnel since isolation is guaranteed by Open vSwitch.

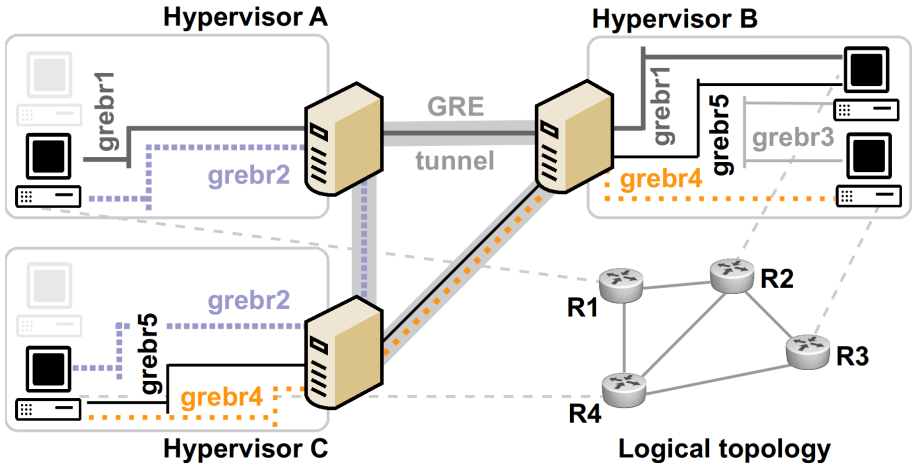


Figure 6.4 – An overview of the communication in point-to-point mode: the bridges that have the source-destination pair on distinct hypervisors will be encapsulated in the GRE tunnel between the hypervisors.

6.1.4 Simulated BGP Network

OSPF handles routing within the AS, whereas BGP interconnects different ASes through external BGP (eBGP) sessions. OSPF achieves full intra-AS reachability and external neighbors are directly connected on the specified interfaces, later redistributed inside the OSPF network.

For inter-domain routing, Quagga implements many BGP features, going from different types of BGP sessions (iBGP or eBGP, route reflector or route-server) to ACLs, filtering, prefix aggregation, etc. During the test phase, the routers are fully capable of forwarding traffic, performing the BGP best path decision process as well as receiving or sending

OSPF and BGP protocol messages.

dVirt Typical Usage

dVirt requires privileged access to a set of hypervisors running Xen with pre-installed software (an SSH server, Python interpreter, Open vSwitch, Quagga and libvirt).

To automatically deploy the testbed, dVirt needs as an input a topology file describing the actual network. Table 6.1 illustrates the elements for defining the simulation: routers, links, BGP sessions. dVirt uses the specified attributes to instantiate a VM for the router RR3, with all the required configuration parameters (distinct management and test addresses, BGP loopback and AS number, etc.). A link between the routers RR3 and Rc is another entry in the topology file, just as the different types of BGP sessions with the desired options.

Table 6.1 – Configuration elements in the topology file

[RR3]	[RR3-RC_link]
type=vm	type=vm_link
hypervisor=A	src=rr3
bridge_ipaddress=10.0.0.104	dst=rc
router_id=203.0.113.198	src_ip=203.0.113.98
name=rr3	dst_ip=203.0.113.97
as=64497	netmask=255.255.255.252
bgp_scantime=5	cost=5
	cost2=5

[RR1-RR3]	[RR3-RC]
type=bgp_session	type=bgp_session
src=rr1	src=rr3
dst=rr3	dst=rc
session_type=ibgp	session_type=rr
mrai=10	mrai=0

Once a network has been deployed with dVirt, the user can perform specific tasks by running customized code: simulate network events such as incoming routes, link failures, etc. It is possible to run any software application or traffic generator on any of the existing virtual routers or in additional virtual machines. Opposed to most of the existing BGP simulators or emulators, dVirt does not restrict the set of potential experiments on top of the deployed BGP network.

By default, dVirt simulates external BGP neighbors of routers with the sbgp software. One or more dedicated virtual machines can host sbgp software instances, where each instance emulates one external BGP neighbor. Sbgp can inject BGP routes from a customizable mrt file but dVirt also includes functions to randomly generate routes.

Another feature of dVirt is that it enables different monitoring strategies (tap the traffic over network interfaces, query the Quagga routing daemon periodically or call functions through the command line) to collect protocol and router behavior data.

Although initially conceived as a simulation platform for testing pre-deployment BGP architectures, dVirt can be seen as an enabler for a transitional implementation of oBGP. Some of the testbed components can be reused for the automated deployment and testing of the oBGP framework, as further detailed in section 6.2.

6.2 The oBGP hub

Integrating into the existing network a software oBGP platform that will be in charge of the routing decisions is much easier than redeploying a new protocol on legacy equipment. The distributed platform handles only the BGP decision process and the dissemination of routes to the client routers that are the actual legacy routers present in the network. We leverage existing virtualization techniques in order to recreate the routing engines performing the BGP decision process. Indeed, let us take a look at how virtualization can work in the context of the oBGP platform.

To achieve the distribution of BGP routes as proposed by the oBGP model, the mix of two main ingredients is required: the virtualization of the routers' control plane for per-client router BGP decisions and the add-paths BGP option in order to take advantage of the route diversity at the edge of the AS.

The add-path option is crucial on all the sessions where route diversity needs to be maintained: from the distributor to the oBGP nodes and the oBGP nodes need to take diversity into account when computing the routes for each client router. To keep the richness of the control plane information, we propose a design that takes route diversity all the way down to the last BGP decision process, just before feeding the client route. To achieve this, we rely on two elements called the oBGP hub and the computational nodes.

We introduce the concept of oBGP logical node which is in fact a detailed view of the oBGP node first presented in the general model. It consists of a route reflector with add-path capabilities called an oBGP hub that is responsible of receiving all the sub-plane paths from the distributors. The hub is in charge of sending the information to the client routers. In the previous section, control plane virtualization was introduced and now it is put to use in the form of computing nodes. The client routers' control plane is replaced by a virtual machine running the BGP routing software that is able to take advantage of the diversity of routes and compute the best x routes on behalf of the real client router. Each computing node is in charge of the control plane processing on behalf of the associated client router. Even if the clients still maintain a legacy control plane, the computing node can feed it

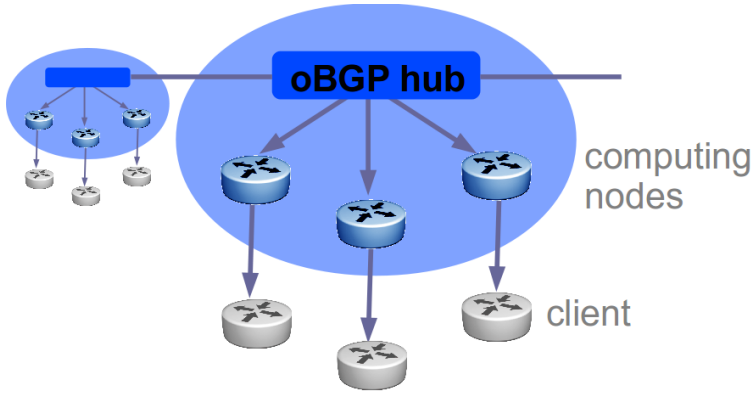


Figure 6.5 – The oBGP hub concentrates all the sub-plane routes and redistributes them to the computing nodes that perform the BGP decision process on behalf of each client

only one route to each prefix that gets automatically picked as best and then installed in the FIB. Another option is to have the virtual control plane perform all decisions until reaching the tie break and send to the client all the routes remaining in the selection process. When using the latter option, the client router still has to perform the last step of the route selection, implying that its local control plane should finish the work. Since there is a one to one mapping between the computing nodes and the client routers, these schemes can be seen as a physical splitting of the control and forwarding plane.

The sessions in Fig. 6.5 are the same type as in the classical route reflection setting. What changes is the ability of the nodes to handle route diversity. The figure shows one oBGP hub connected to another oBGP hub inevitably of the same sub-plane. The oBGP hubs within the same sub-plane are situated at the same level of the route-reflection hierarchy. This means that the oBGP hubs communicate through iBGP sessions and that the computational nodes they serve are their route reflection clients. In their turn, the computational nodes act as the control planes of the final clients and the results they compute need to be sent to the actual client routers that forward the traffic. Since the computational nodes are a virtualized infrastructure that is geographically remote from the clients, another route reflection layer becomes a quick solution. It is worth to notice that in the final step, the diversity of routes does not go down all the way to the client router since we consider it to be legacy equipment that does not have add-path capabilities. More details about reconvergence and route redundancy follow in the next chapter.

The client point of view is depicted in Fig. 6.6, using an example of a network with two sub-planes. The client router needs to retrieve the routes for the entire reachable IP space, so it is fed by both sub-planes. When zooming into the oBGP node in charge of each sub-plane, it becomes clear that the client receives its routes from a computational node that is fed by each of the oBGP hub of the respective sub-plane. The figure equally shows how the oBGP hubs are connected between each other, within a given sub-plane.

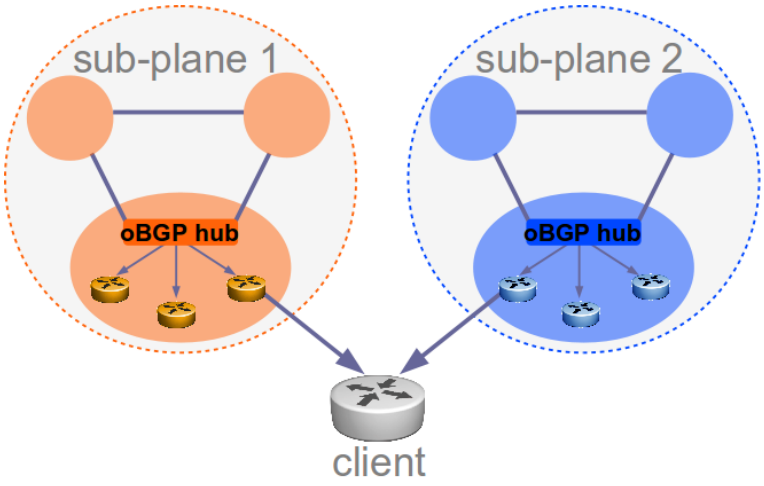


Figure 6.6 – A client receives routes from the computational nodes in both sub-planes

6.2.1 The example of a virtualized PoP

When looking at the actual setup within a PoP, the example provided by Fig. 6.7 shows how the oBGP hubs and the computational nodes interact with the client router.

In this representation, there are two physical machines, each running a hypervisor and several virtual machines. For a clear visual illustration, a single client router is depicted here, served by the computational nodes located on two different hypervisors: the routes corresponding to sub-plane S_1 are sent from the hypervisor Hyp_1 and the routes from the second sub-plane are delivered by the other hypervisor. The hardware performance (in terms of memory allocation and CPU) of the physical systems dictates the maximum number of oBGP client routers that can be served by one hypervisor. Note that each client router requires a virtual machine running its control plane and a common oBGP hub shared by all the computational nodes within the same sub-plane.

Virtualization is used within the PoP in order to reduce the modifications required by the implementation of a solution such as oBGP. Indeed, with the actual scheme it only requires deploying few actual hardware, but the virtual machines offer the same flexibility of real equipment running the BGP control plane. With virtualization comes an overhead, since the additional layers introduced by the concurrent access of the virtual machines to the physical resources. However, we consider the additional latency insignificant compared to the benefits brought by the oBGP model.

6.3 Closing remarks

The content of this chapter is about the dVirt testbed and the benefits that can be leveraged in order to propose a viable implementation of the oBGP framework with the help of

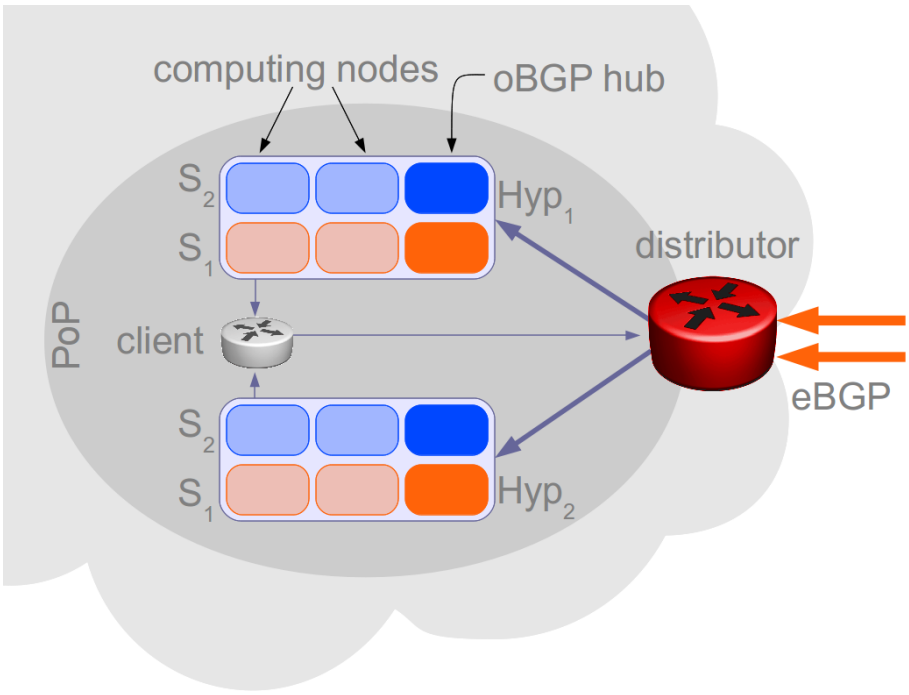


Figure 6.7 – PoP view of a possible implementation, redundancy links not shown. Each sub-plane is made of at least one oBGP hub and a computational node per client router.

virtualization techniques. Initially proposed as a tool for testing and checking actual BGP configurations before actually deploying the given topologies, dVirt is flexible enough to be at the basis of a possible test implementation of oBGP. Keep in mind that the presented architecture is more of a workaround to be used for immediate implementation rather than the long term objective. Since the final goal is to have the oBGP logic integrated in the router vendor equipment, the oBGP hub is only an intermediate solution considered here solely for testing purposes or for networks that are not impacted by a BGP control plane that could be slowed down by the virtualization overhead.

Chapter 7

Evaluation

With a clear view of the oBGP context in mind, the evaluation chapter aims to provide a deeper comprehension of the performances that could be delivered on top of the oBGP model. In order to set the ground for what the chapter encompasses, the reader should keep in mind that the following graphs and charts are based on analytical results. The calculations rely on variations of the different parameters that were previously specified in the Chapter 4. Section 7.1 explains some of the current practices when it comes to network architectures and provides further details about the compromise to be made between the routing table size and the number of sessions required in the meshing; all the investigated features are presented for three well established network sizes with two different meshing types each: a sparse meshing and a more dense session mesh. The final sections in the chapter offer a proof of correctness of the oBGP model and a more practical method for migrating from operational iBGP sessions to the oBGP platform.

7.1 Sizing rules

When designing a network architecture, the total capacity of the equipment is a limiting factor. Router performance is not a simple indicator, but a cumulative measure that takes into account several constraints. It varies as a function of the total number of routers in the topology, the degree of each node in the graph (the number of sessions determines if a node is highly meshed or in a sparse topology), the capacity of the peer router, the size of the RIB on each router, the usage of CPU and memory that can fluctuate according to spurious peaks or periods of idleness.

As an example of explicit sizing rules in networks, we consider here a network with a single level of route reflection and 200 routers distributed across 5 clusters covering a total of 20 PoPs. Each cluster contains two route reflectors that work together as primary plus backup for redundancy reasons. Therefore, each of the 10 RRs in the network is therefore in charge of 19 clients. Since the iBGP mesh requires that the same RR level be highly (possibly fully) meshed, we assume each RR holds the 19 sessions towards its own clients

and at most another 9 sessions to its peers.

Note that the example above is only one possible setting among others. In the same parameter space, a network with different properties responding to another service can be built if the network operator decides to explore the space of only one of the variables: the same network of 200 nodes could be distributed in 10 clusters instead of only 5 clusters over 20 PoPs, with the distinct implication that the distribution and nature of the sessions will change and that a gain in granularity can be canceled by the extra overhead in management. The same logic applies in the case of the oBGP platform, where the model of the split control plane needs to take into account the different compromises acceptable for a given network layout. A smaller set of data to be handled by the client nodes implies dividing the global RIB into more sub-planes which in turn requires introducing more oBGP nodes in the network, supporting a higher number of sessions, configuring more filtering rules on distributors and nodes.

The networks depicted in Fig. 7.1 each have a total of $t = 42$ nodes, but they are distributed in two different manners. The network on the left is sparse, has less interior nodes than ASBRs whereas the one on the right side is more dense, with lesser ASBRs and a higher concentration of iBGP nodes. Note also that the oBGP nodes can be distributed in two different ways: on the left, they are pure oBGP nodes while on the contrary, the oBGP nodes in the right network can be hybrids, acting as distributors at the same time.

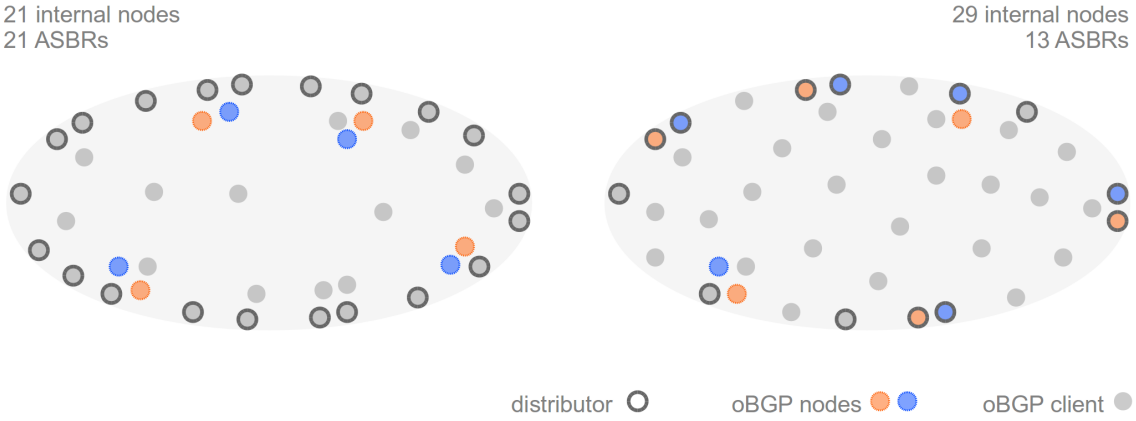


Figure 7.1 – Comparison of two possible scenarios: both networks have an equal number of nodes, but the distribution of the nodes is different.

Table 7.1 provides an overview of the variables used to compute the number of routes, the number of sessions or the number of extra nodes required in the oBGP architecture compared with the classic settings that already exist in current BGP networks. The analytical evaluation of the oBGP platform is presented as a comparison against different versions of a network implementing either the route reflection architecture, the ViAggre settings or the SpliTable solution. The computed parameters that are compared in the rest of this chapter rely on a subset of the following variables that describe the main features of a network.

Table 7.1 – Variables and example values for each notation

small		medium		big		network parameters	
sparse	dense	sparse	dense	sparse	dense		
20	20	200	200	500	500	t	number of routers in the AS topology
380000	380000	380000	380000	380000	380000	p	number of prefixes in the DFZ RIB
10	100	100	2000	100	1000	i	number of internally generated prefixes
5	10	5	10	5	10	q	average number of iBGP peers per router
3	3	3	3	3	3	e	average number of eBGP peers per ASBR
15	3	160	20	300	100	a	number of ASBRs
3	3	20	20	50	50	l	number of PoPs
2	2	2	2	2	2	k	replication factor (SpliTable and oBGP)
3	3	20	20	25	25	s	number of SpliTable Route Servers
2	3	2	4	5	6	n	number of oBGP sub-planes
4	8	12	30	20	60	m	number of oBGP nodes
16	12	188	170	480	440	c	number of oBGP clients
1	1	3	3	5	5	r	number of clusters in one oBGP sub-plane

The following sections present an analytical assessment of the size of the routing table by taking into account variations such as the number of internally generated routes that add up to the Internet RIB in the case of a VPN network and how this increase impacts the routers' RIB; another aspect to take into account is the total number of sessions in small, medium and large topologies, both in the case of a sparse setup or a highly meshed network.

7.1.1 Table size

When it comes to running a network, different constraints are possible depending on the level of quality expected for the delivered services. For example, in most of today's networks, the decision process selects only one best route per Internet prefix and it is the only route passed on to the neighbors in the network. With the arrival of the *add-paths* option, more routes can be selected as being best, increasing the diversity of routes present in the network and hence the number of entries to be processed during the BGP decision algorithm. Note that several best routes per Internet prefix means that each router has to keep a bigger RIB-In for each neighbor in order to accommodate the increasing control plane information conveyed by its BGP peers.

Table 7.2 provides a glimpse of the logic behind such a statement. Indeed, the oBGP nodes

have to process more route advertisement when using *add-paths*. The table also shows that some other solutions such as ViAggre do not attempt to improve the RIB size, but tackle more the number of routes installed in the FIB memory. Since oBGP does not have among its objectives to alter the FIB size, we only take into account RIB calculations, hence from here on the route reflection solution and ViAggre are considered equivalent in terms of RIB size. The SpliTable setup of route servers is a different approach and we remind that this solution is highly dependent on the routes used by the FIB at a precise moment and both the RIB and FIB size fluctuate according to the actual traffic destinations.

Moreover, to give a illustration of these quantities, the graph in Fig. 7.2 shows how the different solutions compare when applied to the six different topologies presented earlier. In the case of oBGP, the graph presents the number of candidate routes entering the BGP decision process on the oBGP nodes. It is visible that each oBGP node running the decision process has a wider palette of choice when it comes to selecting the best routes for its clients. As stated before, the advantage of route diversity is accompanied by a drawback when it comes to CPU load: the task of computing the best routes becomes more tedious since the number of routes increases. However, this compromise is applicable only to the oBGP nodes; the clients receive directly the computed paths and need not rerun a decision process unless specifically mentioned and desired by the network operator.

Table 7.2 – Number of paths processed in various BGP configurations

architecture	Adj-RIB (In+Out)	FIB
full mesh	$p * (2e + 2t - 3)$	p
route reflector	$p * (2e + 2q - 1)$	p
RR client	$p * (2k - 1)$	p
ViAggre	$p * (2e + 2q - 1)$	$2l * p/t + 127 - 2 * 127 * l/t$
SpliTable RSS	$p/s * (a * e + 2l)$	$-$
oBGP node	$p/n * [k * a/m + i/n + c/r + k * (l - 2)]$	p
oBGP client	$p * k$	p

The graph from Fig. 7.2 shows that it is possible to reduce the amount of routes to be processed on the oBGP nodes while increasing the number of sub-planes. This is another compromise since the number of sub-planes drives up the number of sessions involved in the topology. For a study of the number of sessions in the small, medium and big networks, please refer to the following section.

Another important observation is that the diversity of candidate routes depends on the network topology. This is due to the average number of neighbors connected to a node: the amount of routes in the RIB-In is proportionate to the degree of the node in the graph.

Moving forward to another important figure, let us take a look at the Internet DFZ RIB. As seen in the BGP presentation chapter 2, in order to be able to route all Internet traffic,

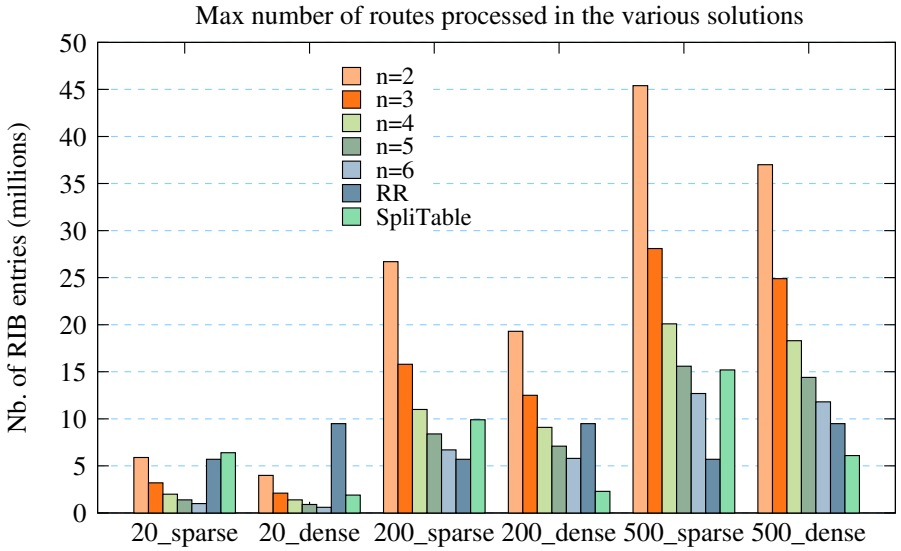


Figure 7.2 – Variation of the RIB size of the different solutions.

a node must maintain a table of all reachable prefixes and the associated exit point. This is in fact the Loc-RIB that feeds the forwarding base and it contains all the routes selected as best by the BGP decision process to be eventually sent to the other BGP peers. In the case of a purely Internet network containing no VPN routes, the Fig. 7.3 shows the actual size of the local RIB on the routers in a classic route reflection topology and in the oBGP scheme.

The data presented here takes into account two possible settings for the oBGP solution: with a duplication of the available best routes for redundancy purposes (meaning that the *add-paths* option is used and the first two best paths are selected by the decision process) and a classic setup with only one best route per Internet prefix. In the first case, a distributor node can use the *best external* option and decide to advertise in the iBGP mesh the route it selects as best among the eBGP received routes. This is a way to achieve hot potato routing, since the distributor does not advertise iBGP routes as being best, but the external routes, forcing thus the traffic to exit the AS. The first case represents a somewhat degraded mode of functioning because it selects a single best route that can be impacted by network events. If a threat appears and the path becomes unavailable, then the network has to wait for reconvergence when another available path is selected.

The second scenario is more optimistic and takes advantage of the *add-paths* option and the path diversity supported by the oBGP platform. With *add-paths*, a distributor can advertise all the eBGP received routes, but this could lead to very high number of routes to be processed on each of the oBGP nodes of the platform, such as the RIB size seen in Fig. 7.2. On the other hand, a more restrictive choice can be made and support two best paths. In this case, the number of routes in the network will likely double, but the benefits come from a secondary path that is already installed and to which the forwarding

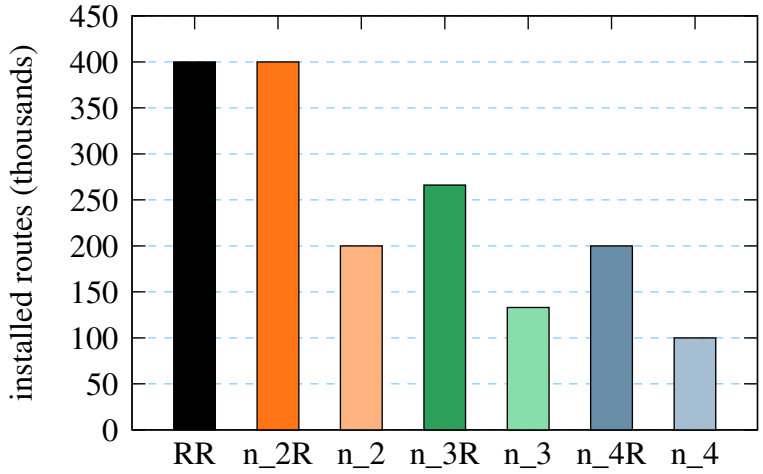


Figure 7.3 – A comparison between the RIB size with no VPN routes.

can resort in case of failure. This failover mechanism is faster because it takes place at the scale of the router and does not require full network reconvergence. However, this mechanism cannot be applied to very sparse networks. Assume an eBGP failure or even a failure on the distributor acting as next hop for a given Internet prefix. If no other exit point is available for the traffic going to this destination, then redundancy becomes useless. This leads to the conclusion that redundancy with two paths per prefix works best in dense networks, where at least two next hops exist for any given prefix. This way, in case of a failure, there is always another next hop to switch to, should the primary one become unavailable.

Similar to the Internet DFZ RIB, the size of the oBGP routing table does not vary as a function of the network topology, but depends on the numbers of sub-planes. The graph in Fig. 7.3 confirms that when using two sub-planes, the RIB gets halved.

All the previous networks deal with a pure Internet routing table, with no VPN routes. What is the impact of the internally generated routes and how do they compare to the DFZ RIB size? In a VPN service network, each router can advertise up to a few thousand routes to its VPN clients. When adding it up to the Internet RIB, one can realize that the number of routes can double or even triple, with some VPN topologies supporting up to a few million routes. The graph in Fig. 9.8 shows how the RIB size increases when each router advertises more and more VPN client routes. The topology used for determining the RIB size is a medium network of 200 nodes. Note that for VPN routes, redundancy cannot be enforced by the engineering rules: two routes need to be announced in the network in order to protect a single VPN site or client.

For comparison purposes, the graph depicts a classic route reflection solution next to the oBGP scheme, using two, three and four sub-planes. If redundancy is used for the Internet routes, the estimations presented hare are not applicable since it is necessary to take into

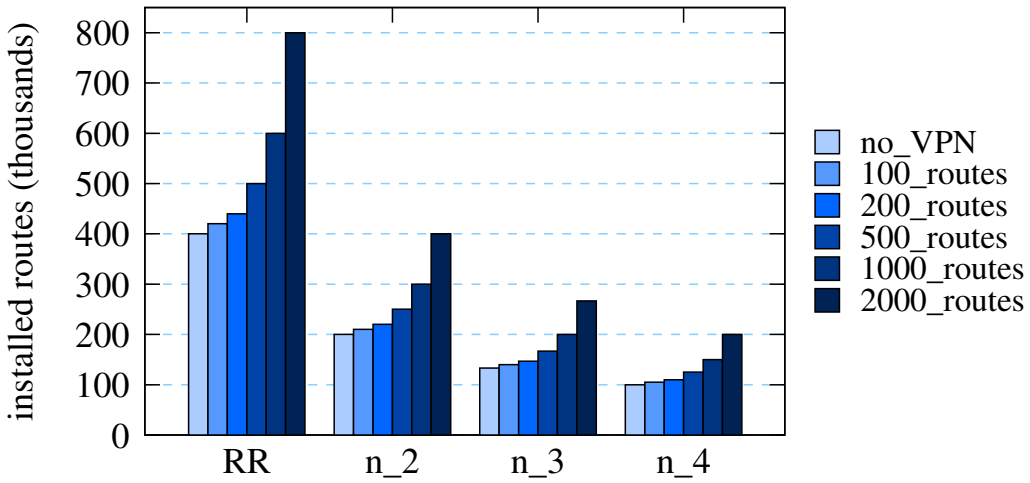


Figure 7.4 – A comparison of the RIB size with a varying number of VPN routes in medium-sized network topology with 200 nodes.

account the correct size of the Internet RIB, including the redundancy. To give an idea, the *no_VPN* number needs to be doubled and the VPN routes added, thus uniformly pushing all the values upwards.

When a VPN route is sent to the distributor, it forwards it to the oBGP nodes that in their turn must make it available to the other oBGP clients present in the network. To avoid loops of VPN routes within the oBGP platform, classic BGP mechanisms are used. A simple approach for preventing a client route to return to the same client that has originally advertised it is to resort to the *originator id* field that gets propagated along the route reflection level. This means that an oBGP node, that behaves in fact as a route reflector for its clients, can propagate the VPN route to all its clients, except for the client that has announced it. A second way to avoid such a looping phenomenon is to use the already established rules in the route reflection schemes who state that no route received on an iBGP session should be forwarded in the iBGP mesh. This feature implemented by route reflection allows the oBGP client to receive VPN routes generated by all the other clients in the network and collected by the oBGP nodes, without being able to resend the received routes again in the iBGP mesh. In the oBGP specific case, the client cannot readvertise the received route to the distributor. In the worst case scenario, if the route is not already filtered based on the *originator id* field, the client receives the route that itself has previously advertised, but without any further consequence.

Management of the VPN routes in the network can be done in another manner that leverages the modular design of oBGP: a network operator can decide to use a specific sub-plane dedicated to VPN traffic. Based on the *address family information* (AFI) or subsequent AFI (SAFI), a filter on the distributor can redirect all the VPN routes to

a previously configured sub-plane. Note that this method can be successfully applied to other types of routes that the engineering teams migh want to process differently, e.g., IPv6 routes.

7.1.2 Number of Sessions

As seen previously, the overall performance of a network is not a uni-dimensional measure, it takes into account multiple aspects. In this section we continue to provide an analysis of the total number of sessions required in the investigated architectures, while keeping this view correlated with the scenarios in the previous section dedicated to the RIB size. Within oBGP, a compromise is required between dividing the routing table in more sub-planes, thus obtaining a smaller RIB to be handled by each node, and the total number of sessions in the network that increases with the number of sub-planes.

Table 7.3 summarizes the computations of the number of sessions, both on a per node basis and a total for the entire network.

Table 7.3 – Number of sessions in different BGP architectures

architecture	Sessions per node	Total sessions
full mesh	$t-1$	$t*(t-1)/2$
route reflector	$k*l-1+c/r$	$k*(t-2*l)+2*l*(2*l-1)/2$
RR client	k	
SpliTable RSS	$a+q$	$[s*(a+q)+a*(s+q)]/2$
SpliTable ASBR	$s+q$	
oBGP node	$c/r+k*(r-2)+a*k/m$	$k*(r-2)+c*k*n+(c-a)*k$ [1]
oBGP client	$k+k*n$	
oBGP distributor	$c*k/a+n*k$	

The three graphs is Fig. 7.5 show how the number of sessions varies in the three different topologies. For each of the small, medium and big network, two settings are possible: in the first case, the routers are connected according to a sparse graph and in the second case, the session mesh is more dense, each node having a higher average number of BGP peers. The graphs explore the results of the oBGP platform using an increasing number of sub-planes that varies from 2 to 6 (in the Fig. 7.5, due to lack of space, the variation of the number of sub-planes in oBGP is defined by n_2, n_3 ... n_6) and two other solutions, the classic route reflection setup (RR) and the SpliTable Route Servers (SpliTable).

In the leftmost graph, corresponding to a small network, we can notice that the differences in terms of sessions between the sparse and dense topologies exist only in the case when

1. The last term is valid only when there are more clients than distributors, otherwise the sessions between client and distributor are already incorporated by the second term.

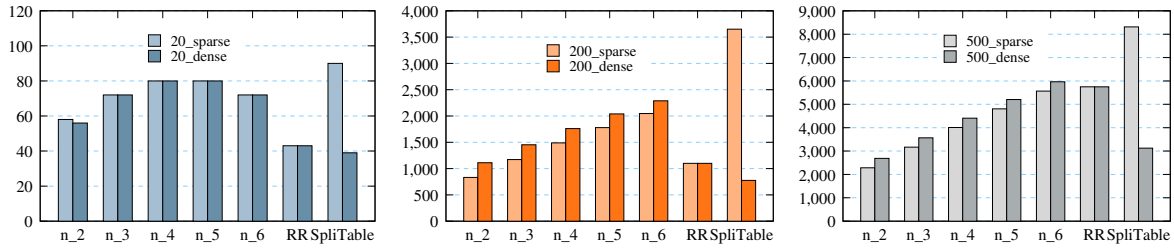


Figure 7.5 – A comparison between the number of sessions in the different solutions. On the y axis the number of sessions of each network topology and on the x axis the different solutions that were investigated: oBGP with a varying number of sub-planes going from 2 to 6, route reflection and the SpliTable Route Servers in small, medium and big networks.

two sub-planes are used in the oBGP platform. This is due to the fact that for higher numbers of sub-planes, the set of oBGP nodes becomes comparable to the set of clients and in certain cases it even surpasses it. Such scenarios are not realistic because there is no logic behind running 12 oBGP nodes to feed only 8 clients, as would be the case for a setting with 6 sub-planes and a redundancy factor of 2.

The gap between the sparse and dense topologies is more visible in the medium and big topologies, where the difference is of the order of 200 and 500 sessions, respectively. This means that the oBGP platform can have an economy of sessions in specifically sparse topologies, bringing thus even more advantages in scalability. On the other hand, this observation contradicts the previous findings about the routing table size who state that oBGP is best fit to more dense topologies in order to guarantee redundancy at the next hop level for Internet destinations. Again, there is a compromise to be made between the RIB size and the total number of sessions, but the figures indicate that the amount of sessions is comparable to today's route reflection session mesh and should not bring any significant penalties at the router level.

7.1.3 Additional equipment

When it comes to introducing a new architecture in the network, the operational teams also take into account the financial aspect. How many new machines are necessary for supporting this new solution? For oBGP, it is clear that additional equipment is required because extra nodes are in charge of the multiple oBGP sub-planes. Taking into account today's DFZ RIB size, a simple solution would be to move to $n = 2$ sub-planes in order to divide and ease the current task of route reflectors. Alleviating the computational requirements can prove to be even more necessary in the case of large VPN or complex networks that support both VPN traffic and the Internet traffic, especially if the hypothesis of continuous RIB growth is maintained. As previously stated, dividing the control plane into two oBGP sub-planes consists of nothing more complicated than actually doubling the number of route reflectors, adding the *add-paths* option and tweaking with routers'

configurations.

The oBGP clients present in the network do not need any special enhancements and on the contrary, the oBGP platform can even integrate legacy equipment that can run a BGP process, but that might not be up to date when it comes to hardware performance. In other words, it means that network operators can exploit equipment for a longer time, thus reducing the necessary investment and delaying the ferocious race to get a more powerful machine to keep up with the traffic frenzy.

Unlike the SpliTable solution, oBGP has no interference whatsoever with the decision process (if we exclude the *add-paths* option advertising the two best paths who will require, depending on the implementation choices, either a selection of the first best path and a second iteration of the decision process over the remaining candidates, either a selection of the last two paths to reach the end of the selection algorithm).

Historically, the full mesh of BGP sessions guaranteed a stable behavior of the protocols, but had poor scalability attributes. With the arrival of route reflection, new equipments and standards have entered the scene. For example, in the case of a network with l PoPs and a redundancy factor of $k = 2$, a total of extra $k*l$ route reflectors have been introduced. With oBGP, this number increases by a factor: $k*l*n$ nodes are now required for running the oBGP platform. Remember, however, that the additional nodes are there to take over the task that was previously done by fewer machines. In the SpliTable solution, things are different since the decision process is altered and the clients need to be aware of a new scheme to store the routes as a distributed hash table. Additionally, SpliTable Route Servers need to be installed in each PoP and each server maintains iBGP sessions with all the ASBRs in the network.

In the proposed oBGP implementation, the oBGP hub is entirely supported by virtual machines, meaning that the overhead for the installation of the new equipment is lower than for actual physical routers. The virtualized PoPs run on a single physical system all the computational processes required by oBGP, so one physical machine can host several oBGP nodes handling multiple sub-planes and the corresponding computational nodes for the clients. This, of course, can limit the PoP redundancy in case of failure, but the proposed intermediary solution can be an easier step towards a full migration to oBGP. Setting up such an architecture has less strict requirements in terms of power, cooling and rack space than a complete oBGP architecture.

From the statements above we can conclude that a moderate effort is required in terms of management of newly introduced equipment and that the complexity of setting up an oBGP platform is similar to a doubling of the classic route reflection architecture.

7.2 Convergence time and correctness

Unlike in other conventional architectures, in the oBGP platform the eBGP received advertisements traverse a finite number of hops: from the external peers to the distributors on eBGP sessions, from the distributors to the oBGP nodes and finally from the nodes

to the client routers. The circuit followed by the control plane messages is guaranteed to avoid looping inside the network.

As seen in Chapter 3, previous studies have confirmed that under given conditions, the transfer time of a full RIB from a RR to a client can currently take as long as five minutes. Although the oBGP architecture does not define a required target time to converge from cold boot, a smaller delay would be an improvement. With the full RIB divided into n sub-planes, the diffusion of BGP routes can be done in parallel since the oBGP nodes in different sub-planes are running the decision algorithm simultaneously and on a smaller number of candidate routes, thus speeding up the entire process.

The convergence time of a protocol is also related to phenomena such as loops in the network. Indeed, in certain settings, BGP is not guaranteed to converge at all and the amount of control messages does not have an upper boundary. Although in the case of the classic route reflection scheme in regular topologies it is not easy to foresee the number of advertisements generated after the reception of a new eBGP route, the oBGP platform is built in such a way that the routes cannot infinitely loop [2] inside the network. Such issues related to the correctness of the signaling path are further discussed.

Let us now examine the steps followed by a route advertisement in the signaling plane, going from the external BGP peers to the clients. Just like in an unmodified BGP architecture, the routes of the Internet flow between the different autonomous systems on external BGP sessions, which means that the oBGP architecture does not interfere in any way in the relationship with the neighboring ASes.

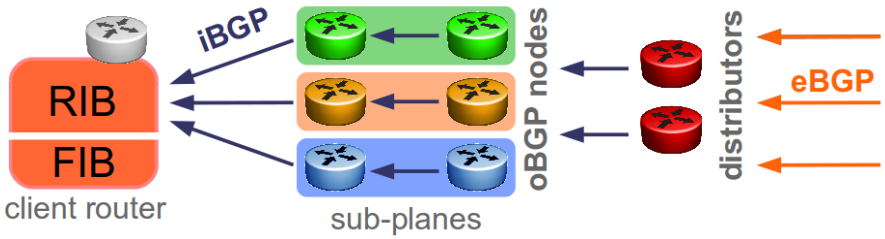


Figure 7.6 – Control plane messages flow from the eBGP peers to the client routers

Although the oBGP model states that the classic ASBRs are renamed distributors and some extra functionalities are added, their purpose remains the same: collect all the external routes and manage to distribute them to the other routers inside the network. The only difference is that the distributor sends the eBGP routes to several oBGP nodes, based on filters describing the different sub-planes. Thus, the behavior of the distributors with respect to the oBGP nodes can be assimilated in practice to that of an ASBR acting as a client of a route reflector.

In the following examples, we consider d is one of the distributor nodes in the platform,

2. In reality, in operational networks, the messages cannot loop infinitely because of mechanisms such as Time To Live, cluster lists, etc.

N is an oBGP node in any of the defined sub-planes and c is a client, as seen in Fig. 4.6. When looking at the session labels between the consecutive elements traversed by a received advertisement, we obtain the following first label on the distributor — oBGP node segment:

$$\forall d \in D, \forall N \in N_{oBGP}, \text{ if } \exists (d, N) \in G_{iBGP}, \text{ then } label(d, N) = up$$

Furthermore, the distributor decides to forward the received route advertisement to the corresponding sub-plane and it is up to the oBGP nodes to disseminate the information to the remaining oBGP nodes of the sub-plane. Remember that redundancy is enforced inside a sub-plane and that the multiple sub-planes of the oBGP model are completely separate and there are no sessions between them. Once the advertisement has reached the correct sub-plane, peer iBGP sessions handle the task of spreading the information to all the other nodes in the same sub-plane. The mesh of sessions between the oBGP nodes resembles the iBGP sessions between the route reflectors of the same level in a classic BGP architecture.

$$\forall N_i, N_j \in N_{oBGP}, \text{ if } N_i = N_j = (s_k), \text{ then } label(N_i, N_j) = over$$

Finally, the last step consists of the session between the oBGP nodes and the client routers that need the route advertisement information for sending the actual traffic to the next hop. Since the client routers act as classic route reflection clients, the changes are minimal. Indeed, the session that feeds the new routes of each sub-plane to the oBGP clients is a conventional route reflector — client session:

$$\forall N \in N_{oBGP}, \forall c \in C, \text{ if } \exists (N, c) \in G_{iBGP}, \text{ then } label(N, c) = down$$

From the concatenation of the steps described above it results that the circuit of a new route advertisement in the control plane follows the valley-free pattern:

$$\mathcal{P}_{oBGP} = \{(up)(over)?(down)\}$$

Indeed, the distributor must send the update message to at least one oBGP node (up) that will flood it to the other nodes in the same sub-plane ($over$), in order for it to reach the final client ($down$). The ($over$) session is not compulsory in the case of a client that is directly connected to the first node that has received the route from the distributor. The regular expression that characterizes the path of an advertisement is in fact a sub-set of the valid paths in the ordinary route reflection architecture:

$$\mathcal{P}_{oBGP} = \{(up)(over)?(down)\} \in \mathcal{P}_{valid} = \{(up) * (over)?(down)*\}$$

This observation does not come as a surprise since the oBGP model relies on the route reflection concept, only split into several sub-planes. In the scenarios depicted in Chapter 4,

the oBGP platform is in fact a mere duplication ($n = 2$ sub-planes) of the route reflection architecture.

In practice, with the implementation described in Chapter 4, due to the extra session between the computing node and the final client, we obtain the following signaling path:

$$\mathcal{P}_{oBGP_implementation} = \{(up)(over)?(down)2\} \in \mathcal{P}_{valid} = \{(up) * (over)?(down)*\}$$

Even so, similar to the general oBGP framework path, this signaling path used in the implementation is also a subset of the correct signaling paths.

From an operational point of view, correctness can be enforced through filtering measures, by making sure the sub-planes are well isolated and that no loops are possible for VPN routes. This requires inbound filters on the client nodes that should check for the correct association between a prefix that has been advertised and the oBGP node that has sent it. Of course, the clients need not be aware of the sub-plane allocation, but configuring such filters can be an extra measure of correctness. The same reasoning applies for the oBGP nodes that should refuse any routes that do not match the sub-plane configured on the oBGP node: the received foreign routes need to be discarded since it means sub-plane integrity has been compromised and the distributor behavior is faulty. These two actions are in fact an additional guarantee to the filtering required on the distributor nodes. Although not compulsory in the oBGP architecture, they are a welcome feature.

7.3 Migration scenario

An oBGP overlay network can be progressively deployed on top of an existing iBGP architecture, using a step by step approach. The overlay is a logical topology and can be optimized according to the underlying physical graph and the location of the oBGP nodes in the network.

As a first step, the network operator has to decide of an initial number of sub-planes and a mapping of each available oBGP node to one sub-plane while taking into account the expected redundancy of the overlay. An initial partitioning of the IP space specifies the granularity of the virtual prefixes and their correspondence to the defined sub-planes. The overlay nodes can be configured with specific policies to apply and finely tune the selection of routes for the individual client routers in the AS.

The second step is twofold: setting up the topology between the nodes of the same sub-plane and interconnecting the different sub-planes. Note that the oBGP approach does not specify a topology and several arrangements of the nodes can be studied in order to optimize the performances. The oBGP nodes participate to the IGP topology to retrieve knowledge of routing costs between any given pair of routers within the AS. The overlay implements redistribution of routes between the sub-planes and replication of the routes in a given sub-plane using reliable flooding mechanisms.

The final step is more delicate and consists of safely migrating the eBGP sessions to the overlay and removing iBGP sessions between routers. Integrity and coherence of routes

must be guaranteed during the transition, until the oBGP manages to take over the route redistribution.

For each border router of the AS, establish one iBGP session with at least one oBGP node of each sub-plane, preferably the closest possible node. At this point, routers can receive the routes from the oBGP overlay. To enable the router to redistribute the eBGP-received routes to its iBGP neighbors, we turn all iBGP sessions of the router into a route reflector to client iBGP session. Once all the internal sessions are replaced by the overlay, the eBGP sessions migrate directly onto an oBGP node. As soon as a pair of border routers is migrated, the session between them can be removed.

When all border routers have been migrated, internal routers having only iBGP sessions are migrated the same way we migrate eBGP connected routers.

7.4 Global assessment

This chapter wraps up the presentation of the work in this manuscript by giving an analytical evaluation of the oBGP solution. The evaluation takes into account two different network topologies with a sparse and dense meshing of the sessions applied to various sizes: a small network of 20 nodes, a middle size network made of 200 nodes and a big topology that includes a total of 500 nodes. As discussed in the section 7.1, there is a tradeoff to be made between the complexity of the session meshing and the burden of the table size; this chapter offers some figures as indicators of the results to be expected in the settings described above. Moreover, section 7.2 presents the arguments that make oBGP a routing platform that can guarantee correctness and certain convergence features thanks to its design and the implemented propagation rules. Finally, a light migration scenario can guide operational entities when it comes to deciding the right order for the passage from classic iBGP sessions to the oBGP architecture.

From the comparisons and analysis provided in this chapter, oBGP does not position itself as the universal solution when it comes to scalability, diversity and correctness in routing, but does come as a viable solution for bigger networks with a more dense meshing. Up to the present day, it turns out that the oBGP routing platform is one of the few proposals that is able to handle several aspects concurrently, given that previous solutions were pointed at one well defined routing issue.

Chapter 8

Conclusion

In this thesis we present a new framework for scalable iBGP routing. The literature survey and a short study presented in 3.1.3 show indeed that BGP currently suffers from a set of flaws, among which reduced scalability, poor route diversity and protocol correctness issues that could impact the network uptime. The major drawbacks identified in the current iBGP are exposed in Chapter 3 and lead to the conviction that there is still room for improving iBGP routing. In order to tackle these shortcomings, the oBGP concept is introduced: an overlay responsible for performing the BGP decision process on behalf of the client routers within the AS.

The main design principles and advantages of the oBGP routing platform are provided in Chapter 4. To answer the need for more scalability in iBGP routing, the general construction of oBGP takes into account a division of the information carried by the control plane. The routing data is split into several subsets called sub-planes that contain in their own turn more granular compartments named virtual prefixes. A sum of benefits accompany this design choice since the oBGP nodes that are in charge of keeping the different sub-plane advertisements incur a smaller load thanks to the splitting of the Internet RIB. The imbalance that could naturally occur between the oBGP nodes due to the heterogeneous size of the virtual prefixes can be fixed with the help of a simple offline greedy algorithm. The method presented in 4.3.3 can automatically allocate a more fair quantity of prefixes to each of the nodes handling a part of the protocol data.

A second direct consequence is that the load on each sub-plane node is less significant than on today's route reflectors. Controlling only a subset of the IP space means less memory needed to store a sub-plane instead of the entire RIB and less BGP reachability information to process on each oBGP node. This means that more complicated computations can be done by the nodes without any penalties compared to the performance obtained in a classic route reflector environment. Furthermore, overall convergence can be improved through a speedup in the selection of the best routes; this positive feature comes also from the distribution of prefixes on several nodes, allowing for parallel computation.

The de-correlation of route selection from propagation allows routers to gain a broader

visibility: the decision process takes into account both BGP and IGP information leading to an optimal selection at the AS-level. The oBGP overlay is aware of all external routes received through eBGP, all the internally generated routes and of the IGP topology, which implies that the decision process is based on complete knowledge of the routes. Being aware of what the client choice would be based on global information, the logic behind the decision process performed on the oBGP nodes can avoid certain routing anomalies. These local events could happen when the BGP selection run by the client takes as input a reduced number of candidate routes.

Another advantage is the increased diversity of routes that can be received by the client routers. Indeed, we first federate knowledge of the routes at the overlay level and then distribute the computational load according to sub-planes so that in the end, client routers connect to each of the sub-planes and receive the optimal routes. With the deployment of the *add-paths* option in the operator networks, a new type of redundancy can be put to use. Receiving at least two best routes for any given Internet destination can be the guarantee of increased resiliency and reduced downtime in highly sensitive networks. Keeping two available routes for each prefix comes, though, at a cost: bigger Adj-RIB-In tables are required per BGP neighbor, meaning that the oBGP nodes have twice as much information to process than in the route reflection setting. This increased load can be compensated by a higher number of sub-planes in the network that will attract, in its own turn, a more dense meshing of the topology. Chapter 7 illustrates the tradeoff necessary between the routing table size and the number of BGP sessions required.

In a broader sense, oBGP can be seen as a reinterpretation of fundamental concepts of distributed computing: the *divide et impera* paradigm for problem decomposition or the more recent *map and reduce* in Google's processing of large amounts of raw data. The novelty proposed by oBGP consists, though, of applying these concepts to routing which is already highly distributed by nature. Previously, routing would have an overall resiliency due to the fact that failures in one region could not generate a failure at the global level and network events would remain local, with the possibility of hiding and isolating them from the rest of the world. With oBGP, visibility goes global and so the routing platform becomes more sensitive to failures and errors and it needs to take into account specific redundancy schemes. From an operational point of view, the architecture and engineering rules need to be adapted to the field reality and the PoP topology. Such considerations are more broadly presented in Chapter 5 where two redundant architectures previously studied for a VPN network are adapted to suit the iBGP model. To make the transition easier towards such architectures, an intermediary setup is proposed at the end of the chapter where a virtualized PoP is depicted. A supplemental migration scenario in Chapter 7 evaluates the steps necessary to a deployment of oBGP, having as departure a one-level route-reflection architecture.

When using the oBGP framework, older equipment such as legacy routers can coexist with more recent router models having richer features. This is possible thanks to the offloading of the control plane on a dedicated platform and the exploitation of the forwarding plane of the client routers. From the ISP point of view, this benefit could mean a potential financial

gain since major investments for replacing the old routers could be delayed, prolonging thus the equipment lifetime.

Among all the solutions cited in the state of the art section, oBGP distinguishes itself from the other iBGP routing proposals due to its threefold objective: improve scalability, increase route diversity and guarantee protocol correctness. There are many aspects in iBGP routing that still demand interest from the research community and comprehension efforts from the network operators, but the oBGP framework takes further the need to cumulate several solutions in order to respond to a larger set of requirements.

8.1 Future Work

The objectives of oBGP are evaluated in an analytical manner in Chapter 7, but a large scale implementation could offer more insights on dynamic features of the protocol that cannot be easily assessed without a real deployment. Within the network, variations are possible for the topology graph, so quirks or corner cases could show up. The oBGP platform should be able to gracefully handle such situations and an implementation could confirm such suppositions. Also, the oBGP model presented and the proposed redundant architectures are built having in mind that the oBGP nodes are not included in the forwarding plane. Even if more simple, the option of having the oBGP nodes completely separated from traffic could be further investigated.

As previously mentioned, the advantage of splitting the routing table can be overshadowed by the computational overhead induced in the overlay. An important point of the solution evaluation consists of determining the optimal threshold for which it is appealing to compute paths with oBGP. After having identified the architecture tradeoffs, it could be useful to observe the behavior of the software on the routers and the CPU load when faced with more complicated computations such as optimal route reflection on the oBGP nodes. Still at the router level, it could be useful to look at methods for optimizing the storage of data structures when it comes to compacting the Adj-RIB-In for oBGP. Currently, compression and pointers are used within classic routers for taking advantage of repetitive information in the data structures kept in the different tables that routers handle. Convergence time is yet another aspect related to the software implementation; depending on certain choices of the router vendors, router behavior during protocol convergence may vary and have considerable impact on the overall network convergence time.

From a purely practical consideration, it could be interesting to observe the behavior of the oBGP platform when facing configuration errors, as they frequently occur in operational networks. The consequences of such misconfigurations can sometimes have very large scale effects, such as in well-known incidents when a Pakistani operator has absorbed all the Internet traffic meant for the YouTube website.

When deploying the oBGP platform, it could possibly impact the eBGP peers. Although conceived not to have any effect on the neighbor ASes, some patterns could show up in the control plane information which might lead the peering ASes to inquire themselves about

the interaction with the oBGP network. In most cases, it is difficult to infer the topology of a given AS only from the exchange of BGP messages; this property should remain unchanged with oBGP. Moreover, the routing platform should not lead to any instability concerning the reachable prefixes and the number of control messages exchanged within the AS in case of failure is guaranteed to be bounded.

Other research perspectives include refining the split algorithm and improving it to gracefully handle the dynamic re-organization of the virtual prefixes on the oBGP nodes. Smarter optimization methods exist that can perform better than the proposed greedy algorithm in terms of a finer balance for the allocation.

If need be, the oBGP nodes can be improved with new BGP options, with limited or no impact on the clients. Through the construction of this approach, oBGP provides ground for implementations of extra features and proposes a new direction in the study of the iBGP control plane scalability.

Chapter 9

Version française abrégée

Introduction

Ce premier chapitre a pour but d'introduire le contexte général des travaux menés pendant la thèse, ainsi que les objectifs et enjeux. Il traite dans un premier temps l'environnement de l'Internet et le routage. Ensuite sont présentés les problèmes engendrés par les solutions choisies pour répondre aux besoins liés à la scalabilité dans le routage. Finalement, les objectifs et les enjeux de ces travaux sont discutés.

L'Internet a commencé comme une expérience dans un laboratoire de test menée par l'armée américaine pour arriver de nos jours un succès incontestable dans le domaine du grand public. En effet, l'Internet a beaucoup changé la manière dont les gens communiquent de nos jours, passant par le banal e-mail jusqu'aux services plus complexes de commerce en ligne, les réseaux sociaux comme Facebook et Twitter ou bien des plateformes de partage vidéo tels que Dailymotion et YouTube.

Puisque l'infrastructure de l'Internet est à la base d'une plage très large de services variés et elle regroupe de multiples réseaux hétérogènes, elle a besoin d'un langage commun pour permettre à tous ces réseaux de s'interconnecter. C'est ici que le protocole Border Gateway Protocol (BGP) fait son apparition en tant qu'unique moyen pour faciliter l'échange entre tous les acteurs de l'Internet. BGP est le seul langage à travers lequel les divers réseaux peuvent intéragir pour s'échanger des messages de routage. Le routage global permet aux utilisateurs de communiquer au-delà de leur réseau local et de joindre des destinations lointaines auxquelles ils ne sont pas directement connectés.

Bien que souvent perçu comme un simple problème de chemins dans un graphe composé de nœuds et de liens, BGP est un protocole plus riche car il permet aux ingénieurs réseaux d'exprimer des politiques de routage. Cela veut dire que le choix du chemin à prendre entre deux points n'est pas toujours basé sur une simple distance, comme dans le cas des protocoles internes utilisant le chemin avec la plus petite métrique. Pour cette raison, BGP ne garantit pas de converence ou des propagations correctes des messages selon les règles établies par les algèbres de routage. Afin de contrôler plus finement le comportement du

protocole et de mettre en place une coordination globale sur les règles de bonne conduite dans l'Internet, plusieurs normes et standards ont modelé la version finale de BGP utilisée aujourd'hui dans les réseaux.

Une des évolutions subies par le protocole consiste dans l'introduction des schémas d'architectures telles que la réflection de routes et les confédérations. Ces deux solutions sont apparues pour réduire le nombre de sessions BGP entre les routeurs d'un réseau. Si avant leur mise en place, chaque routeur devait se connecter directement avec tous les autres routeurs du réseau, cette contrainte est évitée avec ces deux alternatives. En terme de scalabilité, le gain est visible puisque dans les grands réseaux le maillage complet était difficile à gérer et grâce à ces deux solutions, le nombre de connexions est réduit.

En revanche, si ces solutions ont répondu à des contraintes liées à la scalabilité concernant le nombre de sessions, dans certains cas elles ont eu des effets secondaires adverses sur le bon fonctionnement du protocole. Les oscillations de routage et les déflections dans le plan de transmission qui peuvent générer des boucles sont quelques uns des défauts découverts lors de la mise en œuvre de la réflection de routes. La communauté des chercheurs et les opérateurs de réseau continuent à faire des efforts pour mieux comprendre ces anomalies et proposer des nouvelles solutions. C'est pour toutes ces raisons que le travail présenté dans cette thèse se concentre sur une nouvelle approche pour le routage BGP à l'intérieur des réseaux qui peut substituer les schémas actuels.

Objectifs et enjeux

Si la majorité des solutions existentes résolvent un aspect précis de la problématique du routage BGP, la solution proposée dans ce manuscrit vise à prendre en compte des contraintes liées à la scalabilité et une optimisation de la redistribution des routes vers les routeurs clients tout en garantissant un propagation correcte des routes. Cette thèse présente une nouvelle plateforme de routage pour remplacer les schémas en place: oBGP est un réseau de type *overlay* qui est responsable de collecter, traiter à l'aide du processus de décision BGP et redistribuer les meilleures routes pré-calculées pour des routeurs clients. Les composants faisant partie de la plateforme oBGP sont les nœuds, les distributeurs et les clients. Ces éléments sont responsables du plan de contrôle du réseau et leur objectif est de faire en sorte que les routeurs clients aient toujours un meilleur chemin pour joindre une destination. À travers son design qui divise l'information sur plusieurs sous-plans, oBGP gagne en visibilité et peut choisir les meilleures routes pour ses clients tout en respectant des chemins de signalisation valides.

Les principales contributions de cette thèse sont les suivantes:

1. Une analyse des problèmes présents dans le routage BGP montre le besoin d'une amélioration concernant la scalabilité et la diversité des routes, tout en garantissant une propagation correcte des routes. À côté de l'état de l'art, des exemples de données extraites d'un vrai réseau de transit démontrent une perte de diversité dans le cas des routes candidates à l'entrée du processus de décision BGP.
2. Le concepte oBGP est illustré dans le Chapitre 4, ainsi que la façon dont la plateforme

réussit à propager les routes Internet dans cette nouvelle architecture. Les notions de sous-plan et préfixe virtuel sont expliquées, tandis que la dernière section donne un aperçu articulé de l'ensemble.

3. La réalité opérationnelle est prise en compte dans le Chapitre 5 qui présente deux schémas possibles pour assurer la robustesse de l'architecture. Des scénarios de panne sont investigués et une implémentation est suggerée en s'appuyant sur des techniques de virtualisation.
4. Finalement, une évaluation analytique de la solution oBGP est fournie. Une meilleure compréhension des paramètres et des compromis requis par l'adoption de cette architecture est possible à travers des études sur des topologies de petits, moyens et grands réseaux.

Toutes les publications issues des travaux de recherche conduits pendant cette thèse sont disponibles en ligne à la page `www.iuniana.ro`.

Le routage dans l'Internet

Il est important de comprendre le contexte général dans lequel les traveux de cette thèse se situent. Ce chapitre explique comment la communication se fait de bout en bout et illustre des conceptes fondamentaux sur le routage en général et plus précisément sur le routage inter-domaine accompli par BGP. Les derniers paragraphes décrivent les architectures BGP internes à chaque réseau: le maillage complet, les confédérations et la réflection de routes.

En traversant l'Internet, les utilisateurs arrivent à se joindre même à de très grandes distances et ceci n'importe l'endroit géographique où ils se trouvent, pourvu d'être connecté à l'Internet. Ceci est possible car BGP permet aux différents réseaux qui constituent l'Internet de communiquer en utilisant un langage commun. En effet, BGP établit une convention de communication en spécifiant un format standard pour les messages échangés entre les différents domaines adimistratifs, dénommés également Systèmes Autonomes (AS). À l'aide de BGP, les AS apprennent des informations sur le prochain routeur par lequel le trafic doit sortir du réseau pour joindre la destination et le chemin d'AS qu'il faut traverser pour arriver à la destination, ainsi que d'autres attributs spécifiques de la route annoncée. BGP fait partie d'une catégorie de protocoles de routage dits à vecteurs de distance parce que l'information de routage qui est conviée entre deux AS a une structure à deux composantes: une *direction* c'est à dire l'AS Border Router (ASBR) de sortie et une *distance* representée par le nombre de bonds. Le vecteur de chemins est également appelé Network Layer Reachability Information (NLRI) et est obligatoirement présent dans tous les messages BGP de mise à jour (*update*). Plus précisément, BGP est un protocole à vecteurs de chemin car dans le processus de décision de la meilleure route, autres attributs que la distance jouent un rôle dans la sélection.

Non seulement BGP peut-il transporter des attributs supplémentaires concernant les des-

tinations et les routes associées, mais ces éléments permettent aux opérateurs de réseau d'appliquer des politiques de routage. Ces politiques permettent d'influencer le meilleur chemin que le protocole aurait choisi uniquement en fonction de la distance, afin de satisfaire certaines contraintes financières liées aux accords de *peering* passés avec les AS voisins. En revanche, l'utilisation de ces critères ne garantit plus un choix prévisible du plus court chemin, ce qui peut mener parfois à des anomalies dans le routage.

En pratique, BGP peut se diviser en deux protocoles: BGP intérieur (iBGP) pour gérer les messages à l'intérieur d'un AS et BGP extérieur (eBGP) pour échanger des informations sur les préfixes joignable avec les autres AS voisins. Cette différence assez nette est ce qui a permis aux opérateurs de réseaux de déployer des nouvelles architectures à l'intérieur de leurs AS sans aucun impact sur les réseaux voisins. Voilà, par la suite, quelques détails sur le déroulement du mécanisme de sélection de route en BGP.

À l'échelle d'un routeur, le processus de décision doit prendre en compte toutes les interactions avec chaque routeur voisin. En gros, s'il y a n préfixes annoncés dans l'Internet et le routeur en cause a m voisins qui lui envoient la table de routage complète, alors la table de routage iBGP nomée Routing Information Base (RIB) va contenir $m * n$ routes dans le pire cas. L'algorithme de décision BGP sélectionne le meilleur chemin vers chaque destination et l'installe dans la table de commutation nomée Forwarding Information Base (FIB) qui va l'utiliser pour commuter les paquets contenant le trafic des utilisateurs. Cette même route choisie comme étant la meilleure vers le préfixe annoncé sera aussi transmise aux voisins BGP adjacents, afin qu'ils puissent joindre cette destination.

Le processus de décision BGP se déroule pour chaque préfixe annoncé et prend comme paramètre d'entrée toutes les routes BGP disponibles vers cette destination, appellées aussi routes candidates. Un algorithme d'ordonnancement permet de choisir la meilleure route vers un NLRI donné en se basant sur les attributs de chaque route. Il y a plusieurs étapes successives et à la fin de chaque étape, les routes qui ne sont pas optimales sont éliminées. À la fin, une seule route doit rester et c'est la meilleure route. Si à l'avant-dernière étape plusieurs routes candidates sont équivalentes, une règle permet d'en élire une seule en comparant l'identifiant unique des routeurs qui l'ont annoncée et choisissant le plus petit.

Table 9.1 – Le processus de décision BGP

#	préférence	objectif
1	plus grand local_pref	relations économiques
2	plus court chemin d'AS	ingénierie du trafic
3	IGP mieux qu'EGP mieux qu'Incomplete	
4	plus petit multi_exit_disc	
5	plus petite métrique IGP vers le point de sortie EGP	
6	plus petit identifiant du routeur	éliminer l'égalité

À l'intérieur d'un AS, une seule entité administrative gère tous les équipements et met en place une seule politique de routage configurée sur les routeurs BGP. Le but d'iBGP est de redistribuer les messages de routage à l'intérieur de l'AS, tout en respectant cette politique de routage. Auparavant, iBGP demandait un maillage complet de sessions entre tous les routeurs d'un même AS afin de garantir que chaque routeur serait capable d'apprendre la meilleure route externe pour envoyer des datagrammes IP. Cette configuration est vite devenu un problème de scalabilité pour les grands réseaux car le nombre total de sessions croît en fonction du carré de nœuds impliqués. Pour pallier à ce défaut qui engendrait un fort surcoût de calcul, deux solutions alternatives ont vu le jour: les confédérations et les réflecteurs de routes (RRs), comme illustré dans la Fig. 9.1.

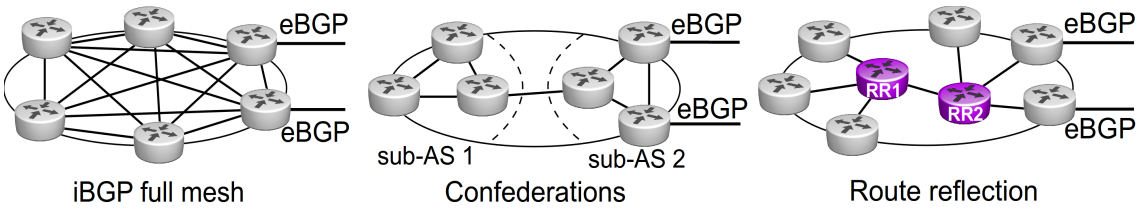


Figure 9.1 – Examples of a full mesh of iBGP sessions, confederations, route reflection

Les confédérations sont des sous-AS qui ont comme but de diviser un grand réseau dans des zones plus faciles à contrôller. Un réflecteur de routes est un routeur qui a un rôle de point central où un groupe de routeurs se connecte afin de recevoir ou envoyer des routes. Ces deux designs présentent des avantages, mais sont également accompagnés de résultats imprévisibles ou d'anomalies comme des oscillations permanentes dans le routage et des boucles qui peuvent affecter la convergence du réseau. Ces schémas peuvent subir aussi des effets de routage sous-optimal à cause d'un masquage d'information ou des décisions non-déterministes influencées par l'état du réseau lors de l'arrivée des annonces BGP.

État de l'art

Puisque BGP n'a pas été conçu avec un modèle mathématique à sa base, au fil du temps BGP a évolué et a subi des changements et des rajouts afin de pouvoir répondre aux nouvelles exigeances. En plus des besoins techniques, les opérateurs ont également exprimé une demande de moyens qui leur permette de mettre en place des politiques de routage assez complexes, parfois même contradictoires entre les différents acteurs de l'Internet.

La communauté de la recherche a investi des efforts dans les études de BGP et a investigué plusieurs aspects du routage inter-domaine, tout en proposant des solutions de contournement pour les problèmes d'architecture BGP, de politique de routage, de scalabilité, de propagation correcte et diversité des routes. Ce chapitre propose une taxonomie non-exhaustive de ces résultats, synthétisée par le tableau 9.2.

Table 9.2 – Taxonomie des solutions déjà existentes

solution proposée	aspect traité
réflection de routes & confédérations	extensibilité
filtrage des préfixes *bogon* et *martiens*, aggrégation des préfixes	
Ballani *et al.* — réduction de la taille de la FIB avec ViAggre	
Zhang *et al.* — Core Router-Integrated Overlay (CRIO)	
Dobrescu *et al.* — rajout de capacité avec RouteBricks	
Vissicchio *et al.* — migration vers plusieurs niveau de RRs (MIRTO)	
Griffin et Wilfong — définition d'une topologie iBGP correcte	propagation correcte
Griffin et Wilfong — analyse de l'oscillation due au MED	
Flavel et Roughan — iBGP stable et flexible	
Van den Schrieck *et al.* — sessions BGP légères	
always-compare-med & set-deterministic-med	
Griffin et Sobrinho — Projet Metarouting	
usage de MPLS pour éviter les déflections	
Raszuk *et al.* — draft sur la réflection de route optimale	
Rawat et Shayman — construction du graphe iBGP	
Feamster *et al.* —le modèle de propagation (up)*(over)?(down)*	
Buob *et al.* — vérification de la fm-optimalité et iBGPv2	
Sarakbi et Maag — réduction du nombre de sessions avec BGP Skeleton	
Vutukuru *et al.* — BGPSep pour une topologie RR correcte	
Uhlig et Tandel — conséquence de la réflection des routes sur la diversité	diversité des routes
Bonavelnture *et al.* — draft sur la réflection de routes intelligente	
Filsfils *et al.* — BGP Prefix Independent Convergence (PIC)	
Pelsser *et al.* — diversité du next-hop iBGP	
Walton *et al.* —draft add-paths	
Bornhauser *et al.* — extensibilité de add-paths	
Feldmann *et al.* — temps de traversée de BGP	convergence
Ben Houidi *et al.* —trous dans les transfert de tables	
Teixeira — sensibilité aux changements de routage	
maillage complet	gestion &
Feamster et Balakrishnan — vérification des configurations des routeurs	dépannage
Feamster *et al.* & Caesar *et al.* — Routing Control Platform (RCP)	plateformes
Pelsser *et al.* & Masuda *et al.* — SpliTable	de routage
Koponen *et al.* — Onix	

Les solutions présentes dans l'état de l'art sont majoritairement dirigées vers une direction établie, visant de résoudre un problème bien spécifique. En observant la liste énumérée dans le tableau 9.2, il y a beaucoup de travaux consacrés à l'extensibilité de BGP pour pouvoir suivre l'explosion de l'Internet, des études et propositions de nouveaux paradigmes de routages pour résoudre les difficultés liées à la propagation correcte des routes dans les architectures BGP et il y a aussi des solutions pour accroître la diversité des routes propagées par les RRs. Une catégorie à part est constituée par les plateformes de routage, qui visent eux, à couvrir un spectre plus large de questions liées au routage BGP. C'est dans ce cadre que la solution oBGP s'inscrit.

Les travaux décrits tout le long de ce manuscrit ont à la base l'hypothèse selon laquelle l'Internet va continuer sa croissance, tant en nombre d'utilisateurs et services, qu'en nombre de réseaux (AS) et de préfixes BGP. Prenant en compte l'évolution possible de BGP à travers des nouvelles options et ajouts, la plateforme oBGP vient pallier des besoins en terme de scalabilité du nombre de routes à transférer dans une architecture qui se veut un remplacement viable de la réflection de routes en iBGP. De la même manière, à travers le design sur lequel le modèle est construit, il est possible de résoudre d'autres ennuis comme la visibilité réduite sur les routes candidates disponibles ou bien de tirer profit d'une architecture distribuée afin d'optimiser et paralléliser le traitement des routes BGP.

La solution oBGP

Dans les réseaux d'aujourd'hui le routage est fortement distribué: chaque routeur dans l'AS prend sa propre décision concernant le chemin vers une destination. oBGP propose de séparer la sélection des routes (plan de routage ou de contrôle) de la commutation du trafic en soi (plan des données ou de forwarding) sur des équipements distincts. Débarrasser les routeurs du plan de contrôle apporte en effet de nombreux avantages décrits dans la suite du mémoire.

Ayant l'occasion de reconsidérer les fondements du design actuel, la solution oBGP met en place une séparation entre le plan de contrôle qui sera géré par un réseau superposé (de type *overlay*) de processus de routage et la transmission des paquets effectuée par les routeurs. La proposition est d'utiliser une plateforme de routage constituée d'engins de routage appelés nœuds oBGP qui fédèrent toute l'information de routage BGP disponible à l'intérieur de l'AS.

Ces nœuds oBGP se comportent comme une entité distribuée qui collecte tous les messages des routeurs de bordure (ASBRs) qui sont eux-mêmes connectés aux différents pairs (*peers*) extérieurs à travers des sessions eBGP. Cette approche permet au réseau superposé (*overlay*) de recevoir toutes les routes des AS voisins et rassembler toutes les routes annoncées de sorte à construire une vision complète et unifiée.

Le but de la plateforme oBGP est de fournir un autre choix viable, différent du routage iBGP actuel, tout en prenant en compte les évolutions futures de l'Internet en termes de taille (numéro d'AS, taille de la RIB BGP, nombre de chemins annoncés pour une même

destination) et en remédiant aux quelques anomalies qui existent en iBGP. L'objectif principal d'oBGP est de collecter les routes reçues en eBGP et de les redistribuer à l'intérieur de l'AS, permettant ainsi aux routeurs et aux hôtes de joindre les destinations externes.

À long terme, oBGP est destiné à être intégré par les équipementiers qui fournissent aujourd'hui les routeurs, mais peut également tourner sur des serveurs supplémentaires au-dessus de matériel générique. Le réseau logique superposé est composé de processus de routage (nœuds oBGP) qui sont conjointement responsables de:

- la collecte, la division et le stockage de l'ensemble complet des routes reçues en eBGP et des routes internes originées par les routeurs de l'AS
- le stockage des politiques de routage et des configurations désirées pour tous les routeurs du réseaux
- la redistribution des chemins calculés vers les routeurs clients de la plateforme oBGP.

Une des préoccupations principales transposée dans les architectures iBGP est le besoin d'extensibilité: pouvoir soutenir la croissance de la table de routage à travers le temps et traiter le nombre de messages protocolaires en augmentation continue. Pour atteindre l'extensibilité souhaitée, la solution oBGP adopte un design où l'information de routage est divisée dans plusieurs *sous-plans* de contrôle. Avec cette approche, différents sous-ensembles des nœuds oBGP gèrent chacun seulement une partie de l'ensemble complet de préfixes de la table de routage (voir Fig. 9.2).

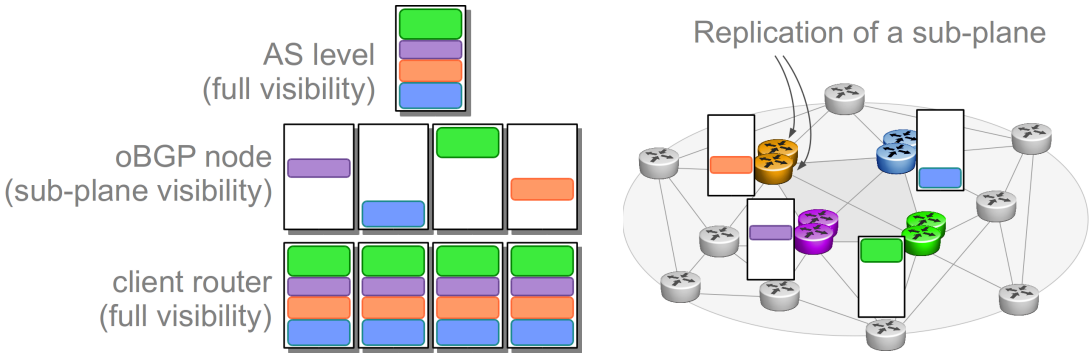


Figure 9.2 – La table de routage est partagée entre les $n = 4$ sous-plans oBGP

Après cette vue générale des idées clé de la plateforme oBGP, la section 4.3 présente les différents éléments de la solution oBGP (voir Fig. 9.3): quel est le rôle des nœuds, comment les routes circulent dans l'architecture globale et surtout quels sont les graphes utilisés pour modéliser oBGP, ainsi que certaines propriétés.

Afin de garantir que la quantité de routes correspondant à chaque sous-plan reste équilibrée, le concept de *préfixe virtuel* est introduit pour une plus fine granularité dans la gestion des préfixes alloués aux sous-plans. En supposant que l'Internet continue à se comporter comme aujourd'hui, une structure de sous-plans figée pourrait devenir déséquilibrée. Au fur et à mesure que les préfixes évoluent, il peut y avoir plus ou moins de routes associées pour les atteindre que ce qui était prévu lors de la mise en place des sous-plans oBGP.

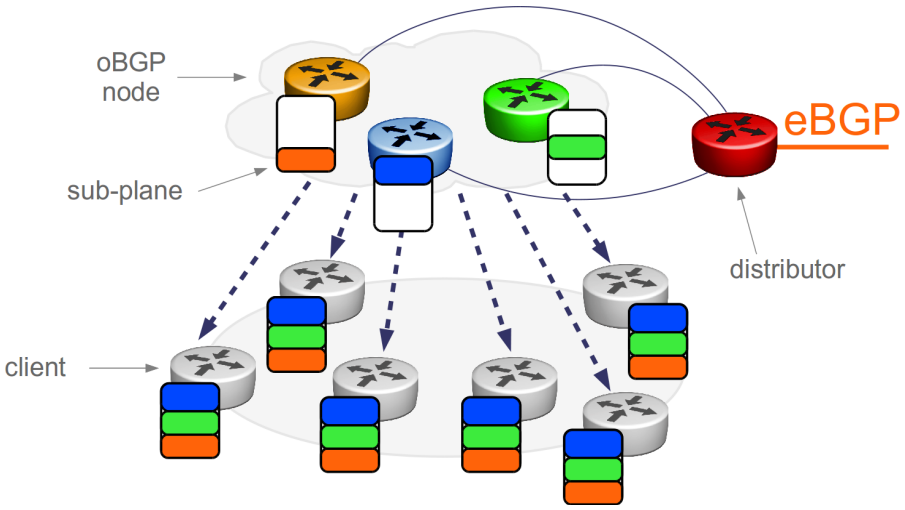


Figure 9.3 – Les différents éléments de la plateforme oBGP (exemple avec 3 sous-plans)

Pour arriver à avoir une répartition équilibrée dans le temps, un algorithme d'allocation et ré-allocation est présenté dans la section 4.3.3.

La partie finale du chapitre, la section 4.4 donne un panorama de la solution oBGP à travers trois visions articulées: comment le modèle oBGP fonctionne au niveau du réseau entier, comment il se décline pour chaque sous-plan et finalement, quel est le point de vue du routeur client et quelles sont ses interactions avec les autres éléments.

Architectures robustes

L'architecture décrite dans le chapitre antérieur est plutôt une base théorique, un modèle pour les différentes architectures qu'il est possible d'instancier dessus. Le cinquième chapitre de ce manuscrit contient des exemples d'architectures oBGP qui prennent en compte des contraintes opérationnelles comme la redondance et le besoin de survivre des cas de panne assez complexes.

Le chapitre débute avec une explication de la réalité du terrain en ce qui concerne les règles d'ingénierie pour le maillage des sessions dans le paradigme actuel: avec la réflexion de route, chaque routeur client a besoin de maintenir au moins deux sessions iBGP vers un réflecteur de route primaire et vers un secondaire. Cette double attache est nécessaire pour pouvoir faire face aux cas de panne qui n'affectent pas directement le routeur client, mais un autre équipement ou lien du réseau.

Il s'agit également d'expliquer certains aspects de la topologie qui sont des conséquences directes engendrées par la nécessité de robustesse au niveau physique, pour pouvoir résister aux pannes électriques, inondations, tremblement de terre etc. Quand une panne de site survient, si les deux réflecteurs de route sont hébergés sur le même site, les effets négatifs

de la panne vont atteindre ces deux réflecteurs et leurs clients correspondants perdent toute communication vers le primaire et vers le secondaire. Pour qu'un client donné ne se retrouve pas complètement isolé de ses deux réflecteurs de route, les deux équipements sont idéalement placés sur des sites distincts.

La deuxième partie du chapitre décrit une architecture de réseau qui peut se coller sur le modèle oBGP générique. La solution 1:1 consiste d'un assemblage des éléments de la plateforme oBGP qui met en œuvre la séparation du plan de contrôle dans plusieurs sous-plans, tel que spécifié par le design global d'oBGP. Cette solution a l'avantage de pouvoir être appliquée pour n'importe quelle valeur paire ou impaire du nombre de sous-plans. Par exemple, le plan de routage peut être divisé en trois parties, comme c'est le cas pour le réseau illustré dans la Fig. 9.4

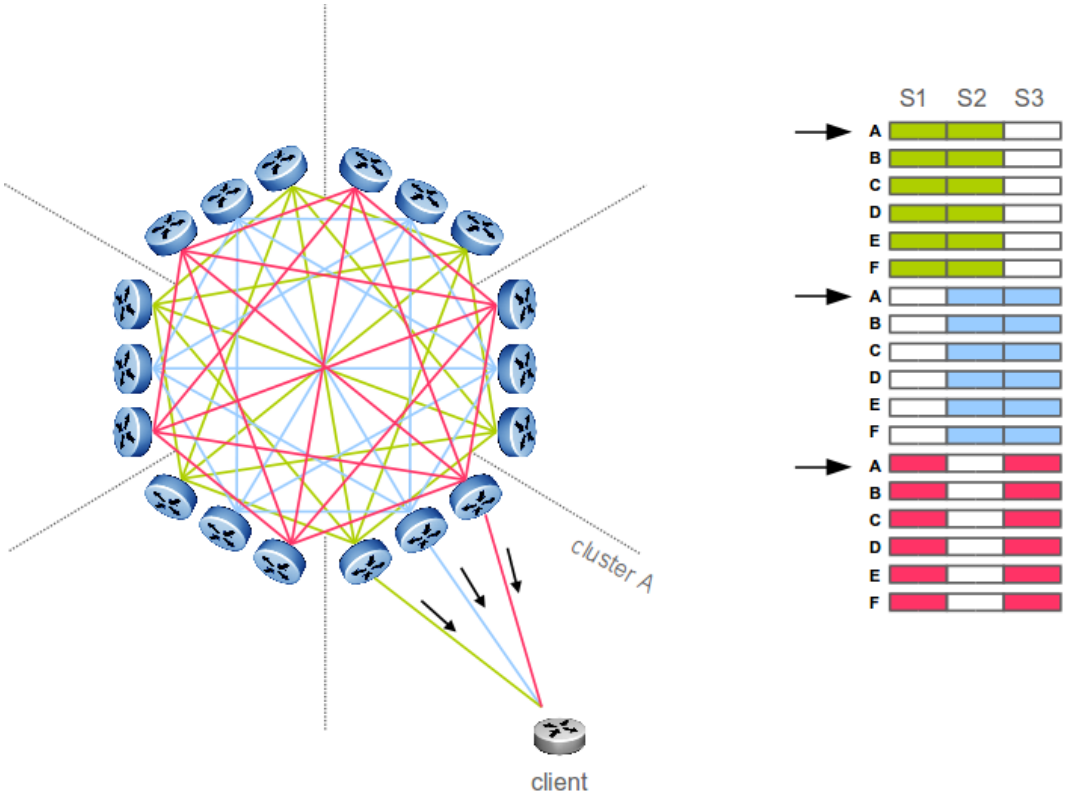


Figure 9.4 – Une architecture oBGP de type 1:1 avec trois sous-plans

Cependant, il existe des cas de double panne qui font que certaines parties de la table de routage ne peuvent pas être globalement propagées. Pour lutter contre ces cas spécifiques, la nouvelle architecture 1+1 introduit un autre type de maillage des sessions à l'intérieur des sous-plans. Bien qu'applicable uniquement pour un nombre pair de sous-plans, la solution 1+1 résiste à certains cas de panne supplémentaires auxquels la précédente architecture 1:1 ne pouvait pas s'opposer.

Le schéma de réseau dans la Fig. 9.5 montre une topologie concrète de l'architecture 1+1

avec deux sous-plans oBGP, six PdPs (Points de Présence), un routeur client et trois distributeurs.

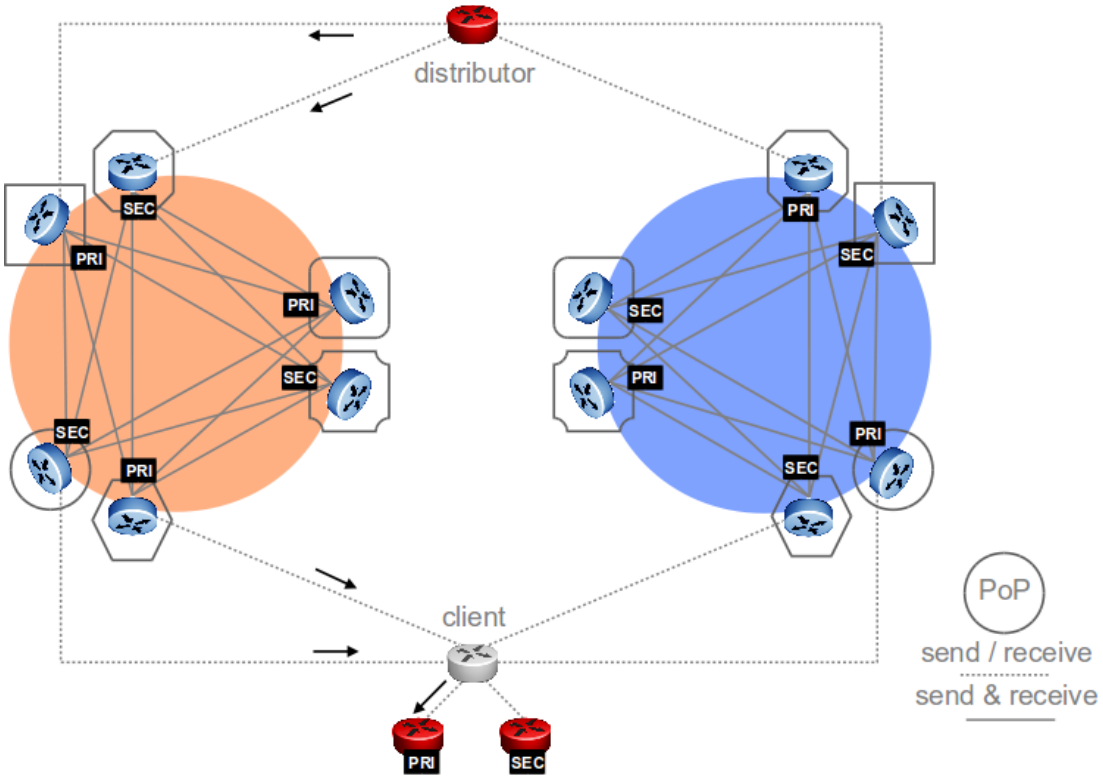


Figure 9.5 – Une vue globale de l'architecture 1+1 avec deux sous-plans

En s'appuyant sur l'exemple de la figure 9.5, la dernière partie de ce chapitre propose d'étudier les différents cas de panne possibles: indisponibilité d'un nœud oBGP (primaire ou secondaire), cas de double panne sur les nœuds oBGP (un nœud dans un sous-plan donné et n'importe quel autre nœud dans le deuxième sous-plan, les deux nœuds dans le même sous-plans mais dans des groupements différents, deux nœuds dans le même sous-plan et dans le même groupement). Au final, les dernières sections sont dédiées aux pannes physiques qui impactent un site entier et à la panne des distributeurs de la plateforme oBGP.

oBGP en pratique

Puisque la plateforme oBGP requiert quelques additions aux logiciels de routage BGP, le sixième chapitre du manuscrit offre une direction possible pour une implémentation de test, en s'appuyant sur les techniques de virtualisation système. Une mise en œuvre complète du modèle oBGP demande quelques changements au niveau des distributeurs et des nœuds oBGP, c'est donc pour cette raison que la solution proposée ici repose sur une

approche logicielle.

Dans un premier temps, le chapitre décrit la plateforme de test dVirt qui a été conçue comme un environnement virtuel distribué utilisé pour les analyses et essais de différentes architectures de routage. La suite contient des détails sur la structure et l'organisation des multiples parties de dVirt, ainsi que les paramètres que l'utilisateur peut contrôler et un exemple d'utilisation.

En dépit d'une large palette d'outils de simulation, très peu d'applications offrent la possibilité d'instancier une plateforme de test de façon automatique pour faire des simulations fidèles, tout en prenant en compte les interactions des protocoles avec les couches en-dessous. dVirt offre une fédération de multiples fonctionnalités dans un outil flexible qui permet à l'utilisateur de déployer et d'évaluer une topologie de réseau.

L'objectif de dVirt est d'être capable de cloner un réseau d'un FAI au-dessus d'une infrastructure virtualisée qui tourne sur un nombre réduit de serveurs. dVirt cherche à reproduire les événements qui pourraient survenir dans un réseau réel et donne à l'utilisateur les moyens pour faire des expérimentations avec les vrais configurations et plan d'adressage.

Le principe de base utilisé par dVirt est la virtualisation: permettre à plusieurs systèmes d'exploitation de s'exécuter au-dessus d'une même machine physique, de manière complètement isolée. Pour cela, l'hyperviseur Xen est utilisé afin de supporter plusieurs machines virtuelles qui sont en fait des distributions Linux Debian avec leurs propres applications. L'exemple dans la figure 9.6 illustre un cas générique.

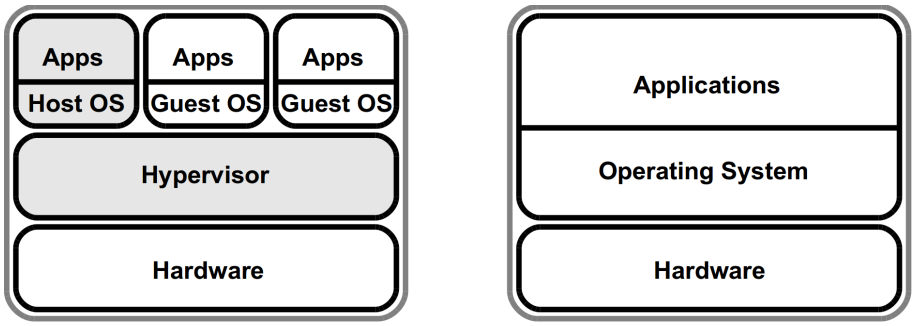


Figure 9.6 – Une comparaison entre deux systèmes: celui de gauche contient trois machines virtuelles au-dessus d'un hyperviseur et celui de droite a un système d'exploitation de type classique.

La virtualisation intervient pour la simulation de plusieurs routeurs. Cela est possible en faisant appel à un logiciel de routage appelé Quagga qui permet d'instancier des engins simulant des routeurs OSPF, IS-IS, RIP ou BGP. De cette manière, plusieurs routeurs virtuels peuvent co-exister sur le même serveur physique. À l'aide des tunnels GRE, plusieurs machines physiques peuvent être reliées de façon transparente et le trafic de la plateforme de test peut être routé entre les différents serveurs, en augmentant ainsi la taille de la plateforme de test.

En prenant comme paramètre d'entrée un fichier qui décrit la topologie à recréer, un

scénario de simulation qui indique les événements réseau à simuler et et s'appuyant sur une bibliothèque de fonctions déjà existantes, dVirt peut construire de façon automatique la plateforme de test (voir Fig. 9.7). dVirt exécute la simulation spécifiée et collecte les résultats sur l'ordinateur qui pilote l'expérimentation.

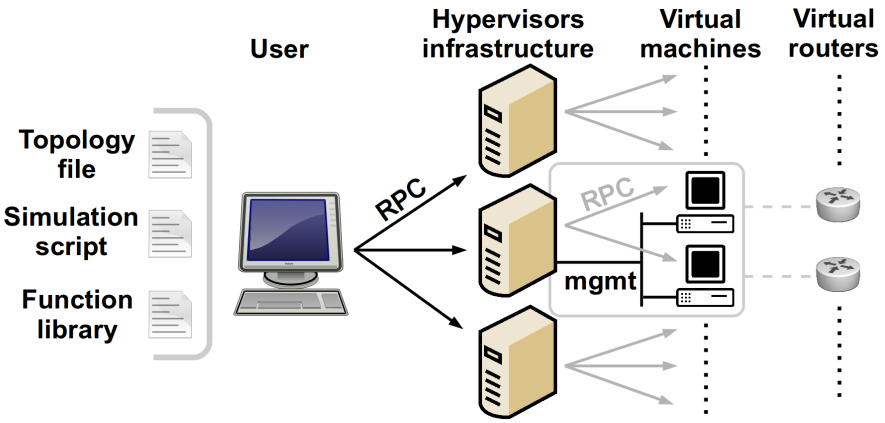


Figure 9.7 – Un aperçu global des composants de dVirt: l'utilisateur interagit à travers un RPC avec les hyperviseurs distants et avec les machines virtuelles correspondantes.

Ensuite, il est possible de ré-utiliser la plateforme dVirt et la convertir afin de tester le modèle oBGP proposé. La dernière partie du chapitre explique comment l'architecture oBGP peut être simulée sur la plateforme dVirt et comment ses éléments sont transposés à l'aide des moyens offerts par dVirt. Il s'agit ici d'une implémentation minimaliste qui ne peut certainement pas explorer toutes les caractéristiques d'oBGP et qui vise surtout à offrir un point de départ solide pour une méthode de test.

Évaluation

Le chapitre d'évaluation propose une mesure analytique des résultats principaux d'oBGP. Il traite les points majeurs que la solution oBGP résout à travers son design et les différentes architectures présentées: le compromis entre la taille de la table de routage et le nombre de sessions, la propagation correcte des chemins étroitement liée au temps de convergence et finalement la complexité d'une éventuelle migration d'iBGP vers une plateforme oBGP.

Les règles de dimensionnement dans un réseau sont en fait des contraintes qui façonnent l'architecture et la topologie. Dans ce chapitre, l'évaluation est basée sur deux types de réseaux: un réseau fortement maillé appelé réseau dense et un autre réseau ayant le même nombre de nœuds, mais connecté à travers moins de sessions que le premier. Ce dernier type de réseau est considéré comme étant un réseau éparse. Certains des graphes et des figures présentés le long de ce chapitre prennent en compte des comparaisons entre des réseaux de taille différente et de maillage dense ou éparse. Les investigations portent sur des topologies de 50, 200 et 500 nœuds, ce qui peut correspondre à des petits réseaux, des

réseaux de taille moyenne et de grands réseaux.

Un tableau résume les paramètres et notations pour les formules à venir et les calculs pour le nombre de sessions correspondant à chacune de ces topologies et le maillage dense ou éparse. En ce qui suit, l'évaluation porte sur la taille de la table de routage en terme de nombre de routes à traiter avec un seul meilleur chemin par préfixe BGP, mais également un scénario où la robustesse est prise en compte et un deuxième chemin vient renforcer la diversité en utilisant l'option add-paths. Un volet est dédié aux réseaux qui transportent non seulement des préfixes Internet, mais supportent aussi un service VPN.

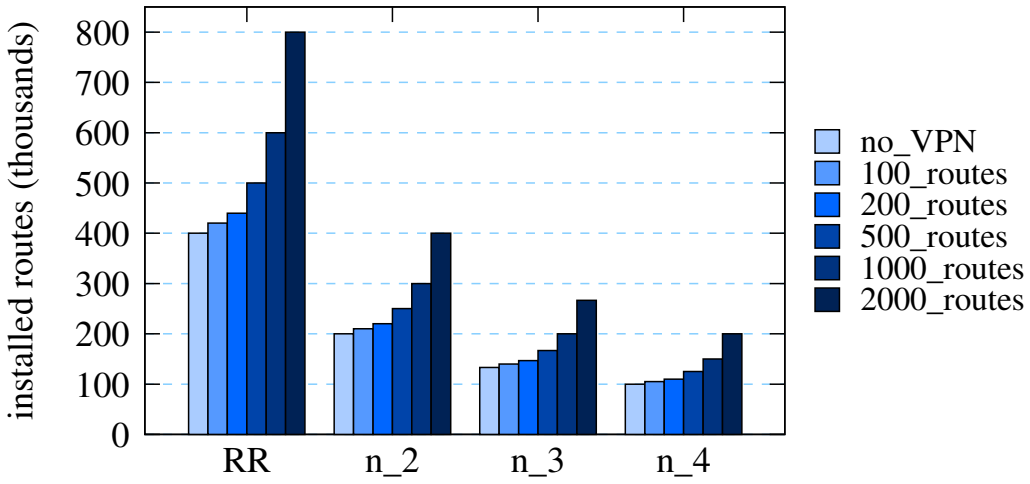


Figure 9.8 – Une comparaison entre les tailles des tables de routage ayant un nombre variable de routes VPN dans le cadre d'une topologie moyenne de 200 nœuds.

Une section spéciale est dédiée à l'étude du nombre de sessions, en comparant les différentes topologies des réseaux utilisant la plateforme oBGP avec d'autres solutions de la littérature telles que SpliTable et la classique réflexion de routes. Ici il faut noter la discussion sur le compromis entre le gain en terme de taille de la table de routage au détriment du nombre de sessions et vice-versa lors de l'augmentation du nombre de sous-plans oBGP.

Les prochains paragraphes portent sur la complexité nécessaire pour l'installation de nouveaux équipements au sein du réseau utilisant le modèle oBGP, ainsi que sur un scénario de migration d'une architecture classique utilisant la réflexion de routes vers la mise en place d'oBGP.

Un autre paragraphe est dédié aux preuves mathématiques qui viennent appuyer les garanties de propagation correcte et le temps de convergence fini dans le cas d'une architecture oBGP.

Conclusion

Cette thèse présente un nouveau modèle d'architecture extensible pour remplacer le routage iBGP actuel. L'état de l'art et une brève étude présentée dans la section 3.1.3 montrent que le routage iBGP est réellement affecté par des défauts tels que le manque d'extensibilité qui empêche le passage à l'échelle, une diversité des routes sévèrement diminuée à l'intérieur des AS et une propagation incorrecte des routes BGP dans certaines conditions qui pourrait éventuellement avoir des conséquences sur la convergence du protocole. Suite aux multiples problèmes d'iBGP identifiés le long du chapitre 3, il devient évident qu'une amélioration est possible. Pour pallier à ces manques identifiés dans le routage iBGP, le modèle oBGP est introduit: il s'agit d'un réseau logique superposé à la topologie physique, basé sur des nœuds intelligents qui constituent ensemble la plateforme de routage oBGP. Cette plateforme est responsable du processus de décision pour chacun des routeurs clients présents dans le réseau, tel un réflecteur de routes global, ayant la vision de toutes les routes annoncées dans l'AS.

Les principes du modèle oBGP, ainsi que les avantages fournis par cette nouvelle architecture de routage sont présentés dans le chapitre 4. Pour mieux répondre aux besoins en termes de passage à l'échelle, le modèle oBGP envisage une séparation du plan de contrôle du plan de données et une division de l'information de routage en plusieurs sous-ensembles disjoints. S'appuyant sur le concept de préfixe virtuel, il est possible d'utiliser la méthode présentée dans la sections 4.3.3 pour arriver à un partage équilibré de la charge sur les nœuds des différents sous-plans oBGP.

Une deuxième conséquence directe de ce partage est le fait que chaque nœud oBGP est responsable d'une partie réduite de l'information de routage, ce qui permet de pouvoir faire des calculs plus complexes à base des messages reçus, en s'appuyant sur des routeurs à capacité de calcul équivalente. En plus, il est possible d'accélérer le processus de convergence puisque la sélection des meilleures routes se déroule en parallèle, dans les différents sous-plans oBGP.

La décorrélation entre la sélection des chemins et leur propagation permet aux routeurs d'acquérir également une visibilité plus large: le processus de décision prend en compte des informations sur les annonces BGP, mais aussi la topologie IGP, permettant ainsi d'aboutir à une sélection optimale au niveau de l'AS. Les choix des routes sont basés sur une connaissance totale des chemins reçu ou générés à l'intérieur de l'AS, ce qui permet d'éviter certaines anomalies de routage qui apparaissent localement.

Un autre avantage est l'augmentation de la diversité des routes qui peuvent être reçues par le routeur client. En effet, il est possible à travers des extensions comme add-paths de recevoir plusieurs routes BGP vers une même destination. Avoir une deuxième route par préfixe garantit une certaine redondance qui peut accélérer le processus de reconvergence. Ce gain ne viens pas sans un effet d'augmentation de la taille des tables de routage: pour une évaluation correcte il faut regarder le compromis à faire entre la taille de la table de routage et le nombre de sessions nécessaires comme illustré dans le chapitre 7.

Dans un sens plus large, oBGP se distingue comme étant une plateforme de routage qui permet de résoudre en même temps plusieurs sujets majeurs discutés dans la littérature et l'état de l'art. Ce nouveau paradigme de routage permet de combiner les avantages de plusieurs solutions déjà citées et d'ouvrir un nouveau point de vue sur le thème du routage iBGP. Les concepts du calcul distribué s'y retrouvent et il est possible d'encadrer oBGP dans les nouvelles direction qui prennent contour dans le monde de la recherche.

Travaux à venir

Les objectifs d'oBGP sont évalués d'une manière analytique dans le chapitre 7, mais une mise en œuvre à plus grande échelle pourrait offrir plus de connaissance sur les caractéristiques dynamiques du protocole qui sont difficilement analysables sans un déploiement réel. Dans les vrais réseaux, des variations peuvent intervenir au niveau du graphe représentant la topologie et il est possible que des situations limite et des cas tordus apparaissent. D'un point de vue de la conception, la plateforme oBGP est censée pouvoir faire face à ces cas, mais seulement une mise en œuvre pourrait confirmer cette supposition. De même, dans le modèle oBGP décrit et les architectures présentées, les nœuds oBGP ne sont pas inclus dans le plan de données, ce qui veut dire qu'ils ne traitent pas le trafic utilisateur. Bien que plus simple, il pourrait être judicieux de tester l'option de la séparation complète des nœuds oBGP du trafic.

Comme il a déjà été mentionné, l'avantage de la division de la table de routage peut être dépassé par la surcharge de calcul nécessaire au niveau de la plateforme oBGP. Un des points essentiels de la solution proposée consiste à déterminer le seuil au-delà duquel il est convenable d'utiliser oBGP pour calculer les meilleurs chemins. Après avoir identifié les différents compromis requis pour la mise en route d'une telle architecture, il est utile d'observer le comportement de la couche applicative des routeurs, ainsi que l'usage du CPU dans le cas d'un usage plus intensif tel que des calculs plus complexes dans le cadre de la réflexion de routes optimale sur les nœuds oBGP. Cependant, il peut s'avérer d'intérêt de faire une analyse de nouvelles méthodes d'optimisation du stockage des structures de données au niveau des routeurs, surtout pour compacter la Adj-RIB-In sur les nœuds oBGP. À présent, des pointeurs et des moyens de compression sont utilisés dans les implémentations classiques des routeurs pour profiter des récurrences multiples des mêmes informations dans les différentes parties des tables de routage qu'un même routeur doit traiter. Le temps de convergence du protocole est un autre aspect étroitement lié à la réalisation de la couche applicative sur les routeurs; en fonction des choix des constructeurs d'équipement, le comportement des routeurs peut varier et donc avoir des conséquences importantes sur la durée de la période de convergence au niveau global du réseau.

En tenant compte des considérations plus pratique, il serait intéressant d'observer le comportement de la plateforme oBGP face à des erreurs de configuration qui arrivent assez fréquemment dans les réseaux opérationnels. Les conséquences de telles fautes peuvent avoir parfois des répercussions au niveau de l'Internet entier, comme il a été le cas pour les incidents provoqués par un opérateur télécom du Pakistan qui avait aspiré tout le trafic

Internet à destination du site YouTube.

Bien que conçu pour ne pas avoir d'influence sur les pairs eBGP, le déploiement d'une plateforme oBGP pourrait engendrer des motifs récurrents dans les annonces BGP. Ces informations pourraient soulever des suspicions des AS voisins et des questions concernant les interactions avec la plateforme oBGP. Dans la plupart des cas, il est difficile d'obtenir des conclusions concernant la topologie d'un AS donné seulement basé sur les messages BGP échangés; cette propriété devrait rester inchangée avec oBGP. De surcroît, la plateforme de routage ne devrait pas engendrer des instabilités dans le routage vers les préfixes joignables ou le nombre de messages de contrôle échangés lors d'une erreur. En effet, le nombre de messages de protocole à l'intérieur de l'AS est garanti d'être fini dans le cas d'une erreur qui pourrait rendre une destination injoignable.

Le modèle oBGP pourrait également être utilisé pour tester des nouvelles options dans les algorithmes de routage, avec des conséquences minimales sur les clients ou même en mode transparent. À travers la construction de cette approche, oBGP offre un terrain prêt à accueillir des nouvelles mises en œuvre des nouvelles caractéristiques dans le routage et propose une direction supplémentaire pour les études de passage à l'échelle du plan de contrôle iBGP.

Appendix

BGP Duplicate Routes

In certain redundant architectures as the one presented in Fig. 9.9, a phenomenon of duplication of the BGP routes can occur when using specific software versions from an equipment vendor on the routers acting as route reflectors. As discussed in chapter 5, operational networks are usually built with redundancy in mind so that they can handle failures, so each client router of the iBGP mesh is connected to a primary and a backup route reflector. The network represented in Fig. 9.9 has a set of three clients that send routes to a cluster of route reflectors which is connected through an iBGP mesh with another cluster of route reflectors in the same AS; it is the second cluster that feeds the routes received from the first cluster to the final client. Indeed, due to the double meshing, the client router is likely to receive the duplicate routes.

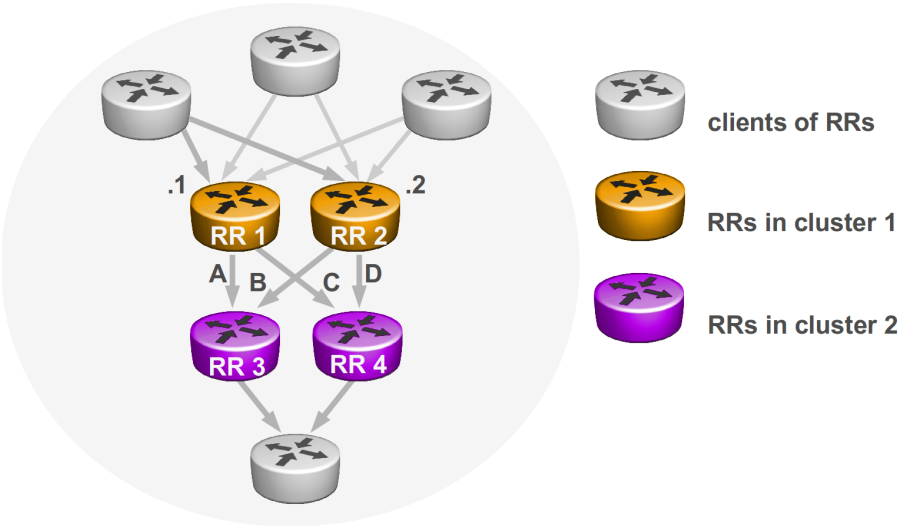


Figure 9.9 – An example topology in which duplicate routes may occur.

Since no mechanism is enforced that will allow the route reflectors in the central mesh to distinguish between the two routes coming from the same client, the same route that is in fact propagated to the primary (router .1) and the backup (router .2) is considered as being in fact two distinct routes. Furhter in the iBGP mesh, each route reflector duplicates again its own route, sending it to the set of iBGP peers from cluster 2: the route-reflector RR1 sends routes A and B and the route-reflector RR2 sends C and D.

Knowing that each route reflector sends only a single best route for each destination, there are multiple possible options depending on the order of arrival of the different routes. As-

sume that B is the best route selected by RR3 and D is the best route for RR4. Depending on the moment of reception of the routes, the following order is possible:

- **A** B **C** D: this is the worst case scenario, as the client router receives first the least preferred routes: the route A comes first, then B arrives and overrides the selection of the route A. The C route comes before the D route which is in fact the route that ends up being selected by the client. Since the client receives all four routes and ends up selecting only two of them (corresponding to the upstream route reflectors), this means that the network incurs a duplication of all the useful routes.
- **A** B D: here the order of reception makes it so that the route D is received by the upstream RR4 before the less preferred C route, avoiding thus the duplication. However, a duplication of the routes still exists because the route reflector RR3 first receives the route A which is not the best, and then the route B arrives.
- B **C** D: same phenomenon as above, only happening on RR3; there is no duplication for the route received by RR3 because the preferred route is received first.
- B D: best case scenario, both preferred routes arrive before the other candidates and there is no duplication of routes.

Note that the four cases above cover all the possible options and that statistically, there are equal chances for any of the four scenarios to occur with the same probability. However, the amount of duplicate routes represents only a small fraction of the network load when compared to the actual payload data.

Bibliography

[Ballani *et al.*, 2008] Hitesh Ballani, Paul Francis, Tuan Cao, and Jia Wang. ViAggre: Making Routers Last Longer! In *Proc. of workshop on Hot Topics in Networks (HotNets-VII)*, Oct 2008.

[Bates *et al.*, 2000] T. Bates, R. Chandra, and E. Chen. BGP Route Reflection - An Alternative to Full Mesh IBGP. IETF RFC2796, April 2000.

[Ben Houidi and Meulle, 2010] Zied Ben Houidi and Mickaël Meulle. A new VPN routing approach for large scale networks. In *IEEE International Conference on Network Protocols (ICNP)*, pages 124 –133, October 2010.

[Ben Houidi *et al.*, 2007] Zied Ben Houidi, Renata Teixeira, and Marc Capelle. Origin of Route Explosion in Virtual Private Networks. In *Proceedings of ACM CoNEXT Student workshop*, December 2007.

[Ben Houidi *et al.*, 2009] Zied Ben Houidi, Mickaël Meulle, and Renata Teixeira. Understanding Slow BGP Routing Table Transfers. In *Proceedings of ACM Internet Measurement Conference (IMC)*, November 2009.

[Ben Houidi, 2010] Zied Ben Houidi. *Scalable Routing in Provider Provisioned Virtual Private Networks*. PhD thesis, Université Pierre et Marie Curie Sorbonne, December 2010.

[Bonaventure *et al.*, 2004] Olivier Bonaventure, Steve Uhlig, and Bruno Quoitin. The case for more versatile BGP Route-Reflectors. Internet draft, draft-bonaventure-bgp-route-reflectors-00, work in progress, July 2004.

[Bornhauser *et al.*, 2010] Uli Bornhauser, Peter Martini, and Martin Horneffer. Root Causes for iBGP Routing Anomalies. In *Proceedings of the 35th IEEE Conference on Local Computer Networks (LCN)*, pages 488–495, October 2010.

[Bornhauser *et al.*, 2011] Uli Bornhauser, Peter Martini, and Martin Horneffer. Scalability of iBGP Path Diversity Concepts. In *Proceedings of the 10th IFIP Networking*, pages 432–443. Springer, May 2011.

[Buob *et al.*, 2007] Marc-Olivier Buob, Mickael Meulle, and Steve Uhlig. Checking for optimal egress points in iBGP routing. In *International Workshop on Design and Reliable Communication Networks (DRCN)*, October 2007.

[Buob *et al.*, 2008] Marc-Olivier Buob, Steve Uhlig, and Mickael Meulle. Designing optimal ibgp route-reflection topologies. In *Networking*, pages 542–553, 2008.

[Buob, 2008] Marc-Olivier Buob. *Routage interdomaine et intradomaine dans les réseaux de cœur*. PhD thesis, Université d'Angers, October 2008.

[Caesar *et al.*, 2005] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe. Design and implementation of a routing control platform. In *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, Berkeley, CA, USA, 2005.

[Cazaux *et al.*, 2009] Christophe Cazaux, Laurent Piget, and Guillaume Gaulon. VPN Partitionning – HLD and LLD. France Telecom Group Restricted document, 2009.

[Dobrescu *et al.*, 2009] Mihai Dobrescu, Norbert Egi, Katerina Argyraki, Byung-Gon Chun, Kevin Fall, Gianluca Iannaccone, Allan Knies, Maziar Manesh, and Sylvia Ratnasamy. RouteBricks: Exploiting Parallelism To Scale Software Routers. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, SOSP '09, pages 15–28, New York, NY, USA, 2009. ACM.

[Euro-IX, 2011] Euro-IX. European Internet Exchange Association. http://www.euro-ix.net/, 2011.

[Fabrikant *et al.*, 2011] A. Fabrikant, U. Syed, and J. Rexford. There's something about MRAI: Timing diversity can exponentially worsen BGP convergence. In *Proceedings of IEEE INFOCOM*, pages 2975 –2983, april 2011.

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

[Feamster *et al.*, 2004a] Nick Feamster, Hari Balakrishnan, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. The case for separating routing from routers. In *FDNA '04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pages 5–12, New York, NY, USA, 2004. ACM.

[Feamster *et al.*, 2004b] Nick Feamster, Jared Winick, and Jennifer Rexford. A Model of BGP Routing for Network Engineering. In *in Proc. ACM SIGMETRICS*, pages 331–342, 2004.

[Feldmann *et al.*, 2004] Anja Feldmann, Hongwei Kong, Olaf Maennel, and Alexander Tudor. Measuring BGP Pass-Through Times. In *Passive and Active Measurement Workshop (PAM)*, pages 267–277, 2004.

[Filsfils *et al.*, 2011] Clarence Filsfils, Pradosh Mohapatra, John Bettink, Pranav Dharwadkar, Peter De Vriendt, Yuri Tsier, Virginie Van den Schrieck, Olivier Bonaventure, and Pierre François. BGP Prefix Independent Convergence (PIC) Technical Report. Technical report, Cisco, 2011. http://www.cisco.com/en/US/prod/collateral/routers/ps5763/bgp_pic_technical_report.pdf.

[Flavel and Roughan, 2009] Ashley Flavel and Matthew Roughan. Stable and Flexible iBGP. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pages 183–194, New York, NY, USA, 2009. ACM.

[Francis *et al.*, 2011] Paul Francis, Xiaohu Xu, Hitesh Ballani, Dan Jen, Robert Raszuk, and Lixia Zhang. FIB Suppression with Virtual Aggregation, July 2011. Internet draft, draft-ietf-grow-va-05, work in progress.

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

[Gao and Rexford, 2000] Lixin Gao and Jennifer Rexford. Stable Internet Routing without Global Coordination. In *IEEE/ACM Transactions on Networking*, pages 681–692, 2000.

[Gao, 2001] Lixin Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking (TON)*, 9(6):733–745, 2001.

[Griffin and Sobrinho, 2005] Timothy G. Griffin and Joäo Luís Sobrinho. Metarouting. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '05, pages 1–12, New York, NY, USA, 2005. ACM.

[Griffin and Wilfong, 2002a] Timothy Griffin and Gordon T. Wilfong. Analysis of the MED Oscillation Problem in BGP. In *Proceedings of the 10th IEEE International Conference on Network Protocols*, ICNP '02, pages 90–99, Washington, DC, USA, 2002. IEEE Computer Society.

[Griffin and Wilfong, 2002b] Timothy G. Griffin and Gordon Wilfong. On the correctness of iBGP configuration. *SIGCOMM Computer Communications Review*, 32(4):17–29, 2002.

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

[Huston, 2011a] Geoff Huston. BGP AS count. http://www.cidr-report.org/cgi-bin/plot?file=%2fvar%2fdata%2fbgp%2fas2.0%2fbgp-as-count.txt, 2011.

[Huston, 2011b] Geoff Huston. BGP Routing Table Analysis Reports. http://bgp.potaroo.net/, 2011.

[Inter-Domain Routing Workgroup, 2011] IETF Inter-Domain Routing Workgroup. http://datatracker.ietf.org/wg/idr/, 2011.

[Internet World Stats, 2011] Internet World Stats. http://www.internetworldstats.com/stats.htm, 2011.

[Ishiguro, 1991] Kunihiro Ishiguro. Quagga Software Routing Suite. http://www.quagga.net/, 1991.

[Jasinska *et al.*, 2011] Elisa Jasinska, Nick Hilliard, Robert Raszuk, and Niels Bakker. Internet Exchange Route Server. Internet draft, draft-jasinska-ix-bgp-route-server-02, March 2011.

[Juniper Networks, 2002] Juniper Networks. Differences Between BGP Route Reflectors and Confederations. https://www.juniper.net/customers/csc/documentation/techdocs/downloads/pdf/350010.pdf, 2002.

[Koponen *et al.*, 2010] Teemu Koponen, Martin Casado, Natasha Gude, Jeremy Stribling, Leon Poutievski, Min Zhu, Rajiv Ramanathan, Yuichiro Iwata, Hiroaki Inoue, Takayuki Hama, and Scott Shenker. Onix: A Distributed Control Platform for Large-scale Production Networks. In *OSDI*, 2010.

[Labovitz *et al.*, 2000] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. In *in Proc. ACM SIGCOMM*, pages 175–187, 2000.

[Le *et al.*, 2010] Franck Le, Geoffrey G. Xie, and Hui Zhang. Theory and new primitives for safely connecting routing protocol instances. In *Proceedings of ACM SIGCOMM*, 2010.

[Libvirt - The virtualization API] Libvirt - The virtualization API. http://www.libvirt.org/.

[Masuda *et al.*, 2011] Akeo Masuda, Cristel Pelsser, and Kohei Shiomoto. Splitable: Toward routing scalability through distributed bgp routing tables. *IEICE TRANSACTIONS on Communications*, E94-B(01), January 2011.

[McPherson *et al.*, 2002] Danny McPherson, Vijay Gill, Daniel Walton, and Alvaro Retana. Border Gateway Protocol (BGP) Persistent Oscillation Condition, August 2002.

[Metarouting, 2011] Metarouting. http://www.cl.cam.ac.uk/~tgg22/metarouting/, 2011.

[Meulle, 2007] Mickaël Meulle. *Inférence des accords économiques et des politiques de routage dans l'Internet*. PhD thesis, Université Blaise Pascal de Clermont-Ferrand, March 2007.

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

[Oprescu *et al.*, 2011a] Iuniana Oprescu, Mickaël Meulle, and Philippe Owezarski. dVirt: a Virtualized Infrastructure for Experimenting BGP Routing. In *IEEE Local Computer Networks*, October 2011.

[Oprescu *et al.*, 2011b] Iuniana Oprescu, Mickaël Meulle, Steve Uhlig, Olaf Maennel, Cristel Pelsser, and Phillippe Owezarski. oBGP : an Overlay for a Scalable iBGP Control Plane. In *Proceedings of the 10th IFIP Networking*, number i, May 2011.

[Oran, 1990] D. Oran. OSI IS-IS Intra-domain Routing Protocol. RFC 1142, IETF, February 1990.

[Pei and Van der Merwe, 2006] Dan Pei and Jacobus Van der Merwe. BGP Convergence in Virtual Private Networks. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 283–288, New York, NY, USA, 2006. ACM.

[Pelsser *et al.*, 2008] Cristel Pelsser, Tomonori Takeda, Eiji Oki, and Kohei Shiomoto. Improving Route Diversity through the Design of iBGP Topologies. In *IEEE International Conference on Communications (ICC '08)*, pages 5732 –5738, may 2008.

[Pelsser *et al.*, 2009] Cristel Pelsser, Akeo Masuda, and Kohei Shiomoto. Scalable support of interdomain routes in a single as. In *Proceedings of the 28th IEEE conference on Global telecommunications*, GLOBECOM'09, pages 2785–2792, Piscataway, NJ, USA, 2009. IEEE Press.

[Pelsser *et al.*, 2011] Cristel Pelsser, Olaf Maennel, Pradosh Mohapatra, Randy Bush, and Keyur Patel. Route flap damping made usable. In *Proceedings of the 12th international conference on Passive and active measurement*, PAM'11, pages 143–152, Berlin, Heidelberg, 2011. Springer-Verlag.

[Raszuk *et al.*, 2011] Robert Raszuk, Christian Cassar, Erik Aman, and Bruno Decraene. BGP Optimal Route Reflection (BGP-ORR). Internet draft draft-ietf-idr-bgp-optimal-route-reflection-00, June 2011.

[Rawat and Shayman, 2006] Anuj Rawat and Mark A. Shayman. Preventing Persistent Oscillations and Loops in iBGP Configuration with Route Reflection. *Comput. Netw.*, 50(18):3642–3665, 2006.

[Rekhter and Li, 1995] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, IETF, March 1995.

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

[Rexford *et al.*, 2002] Jennifer Rexford, Jia Wang, Zhen Xiao, and Yin Zhang. BGP Routing Stability of Popular Destinations. In *Internet Measurement Workshop*, pages 197–202, 2002.

[Rexford, 2011] Jennifer Rexford. http://www.cs.princeton.edu/~jrex/, 2011.

[Sarakbi and Maag, 2010] Bakr Sarakbi and Stephane Maag. BGP Skeleton: An Alternative to iBGP Route Reflection. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 301–305, 2010.

[Teixeira *et al.*, 2007] Renata Teixeira, Steve Uhlig, and Christophe Diot. Bgp route propagation between neighboring domains. In *Proceedings of the 8th international conference on Passive and active network measurement*, PAM'07, pages 11–21, Berlin, Heidelberg, 2007. Springer-Verlag.

[Teixeira, 2005] Renata Teixeira. *Network Sensitivity to Intradomain Routing Changes*. PhD thesis, University of California San Diego, August 2005.

[Traina *et al.*, 2001] P. Traina, D. McPherson, and J. Scudder. Autonomous System Confederations for BGP. RFC 3065, IETF, February 2001.

[Uhlig and Tandel, 2005] Steve Uhlig and Sébastien Tandel. On the impact of route-reflection on route diversity. France Telecom Tech Report, November 2005.

[Uhlig and Tandel, 2006] Steve Uhlig and Sébastien Tandel. Quantifying the BGP Routes Diversity Inside a Tier-1 Network. In *Networking*, pages 1002–1013, 2006.

[Van den Schrieck and François, 2009] Virginie Van den Schrieck and Pierre François. Analysis of paths selection modes for add-paths. Internet draft draft-vvds-add-paths-analysis-00, July 2009.

[Van den Schrieck *et al.*, 2006] Virginie Van den Schrieck, Pierre François, Sébastien Tandel, and Olivier Bonaventure. Let BGP speakers configure their iBGP sessions on their own. Position Paper, Wired 2006 Workshop, Atlanta, October 2006.

[Villamizar *et al.*, 1998] Curtis Villamizar, Ravi Chandra, and Ramesh Govindan. BGP Route Flap Damping. RFC 2439, IETF, November 1998.

[Vissicchio, 2012] Stefano Vissicchio. *Governing Routing in the Evolving Internet.* PhD thesis, Universita' degli Studi di Roma "Roma Tre", Dottorato di Ricerca in Ingegneria, Sezione Informatica ed Automazione, XXIV Ciclo, 2012.

[Vutukuru *et al.*, 2006] Mythili Vutukuru, Paul Valiant, Swastik Kopparty, and Hari Balakrishnan. How to Construct a Correct and Scalable iBGP Configuration. In *IEEE INFOCOM*, Barcelona, Spain, April 2006.

[Walton *et al.*, 2011] Daniel Walton, Alvaro Retana, Enke Chen, and John Scudder. Advertisement of Multiple Paths in BGP. Internet draft, draft-ietf-idr-add-paths-05, July 2011.

[Wenhua *et al.*, 2009] Wang Wenhua, Shen Qinguo, Song Yu, and Han Chunyong. New mrai setup mechanism for enhancing bgp route convergence. In *Information, Communication and Automation Technologies, 2009. ICAT 2009. XXII International Symposium on*, pages 1 –7, oct. 2009.

[Wu *et al.*, 2005] 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 *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 1–14, Berkeley, CA, USA, 2005. USENIX Association.

[Zhang *et al.*, 2006] Xinyang Zhang, Paul Francis, Jia Wang, and Kaoru Yoshida. Scaling IP Routing with the Core Router-Integrated Overlay. In *Proceedings of the 2006 IEEE International Conference on Network Protocols (ICNP)*, pages 147–156. IEEE Computer Society, 2006.

Abstract

The Internet is organized as a collection of networks called Autonomous Systems (ASes). The Border Gateway Protocol (BGP) is the glue that connects these administrative domains. Communication is thus possible between users worldwide and each network is responsible of sharing reachability information to peers through BGP. Protocol extensions are periodically added because the intended use and design of BGP no longer fit the current demands. Scalability concerns make the required internal BGP (iBGP) full mesh difficult to achieve in today's large networks and therefore network operators resort to confederations or Route Reflectors (RRs) to achieve full connectivity. These two options come with a set of flaws of their own such as route diversity loss, persistent routing oscillations, deflections, forwarding loops etc.

In this dissertation we present oBGP, a new architecture for the redistribution of external routes inside an AS. Instead of relying on the usual statically configured set of iBGP sessions, we propose to use an overlay of routing instances that are collectively responsible for (I) the exchange of routes with other ASes, (II) the storage of internal and external routes, (III) the storage of the entire routing policy configuration of the AS and (IV) the computation and redistribution of the best routes towards Internet destinations to each client router in the AS.

Résumé

L'Internet est organisé sous la forme d'une multitude de réseaux appelés Systèmes Autonomes (AS). Le Border Gateway Protocol (BGP) est le langage commun qui permet à ces domaines administratifs de s'interconnecter. Grâce à BGP, deux utilisateurs situés n'importe où dans le monde peuvent communiquer, car ce protocole est responsable de la propagation des messages de routage entre tous les réseaux voisins. Afin de répondre aux nouvelles exigences, BGP a dû s'améliorer et évoluer à travers des extensions fréquentes et de nouvelles architectures.

Dans la version d'origine, il était indispensable que chaque routeur maintienne une session avec tous les autres routeurs du réseau. Cette contrainte a soulevé des problèmes de scalabilité, puisque le maillage complet des sessions BGP internes (iBGP) était devenu difficile à réaliser dans les grands réseaux. Pour couvrir ce besoin de connectivité, les opérateurs de réseaux font appel à la réflection de routes (RR) et aux confédérations. Mais si elles résolvent un problème de scalabilité, ces deux solutions ont soulevé des nouveaux défis car elles sont accompagnées de multiples défauts; la perte de diversité des routes candidates au processus de séléction BGP ou des anomalies comme par exemple des oscillations de routage, des déflections et des boucles en font partie.

Les travaux menés dans cette thèse se concentrent sur oBGP, une nouvelle architecture pour redistribuer les routes externes à l'intérieur d'un AS. À la place des classiques sessions iBGP, un réseau de type *overlay* est responsable (I) de l'échange d'informations de routage avec les autres AS, (II) du stockage distribué des routes internes et externes, (III) de l'application de la politique de routage au niveau de l'AS et (IV) du calcul et de la redistribution des meilleures routes vers les destinations de l'Internet pour tous les routeurs clients présents dans l'AS.