UNIVERSIDAD DE LA REPÚBLICA

FACULTAD DE INGENIERIA
INSTITUTO DE COMPUTACIÓN -INCO
PEDECIBA Informática

P H D T H E S I S

to obtain the title of

PhD on Informatics

of the Universidad de la República

Defended by

Cristina Lucía MAYR TRENTINI
(mayr@fing.edu.uy)

Optimal Route Reflection Topology Design

Thesis Advisor: Claudio RISSO
Thesis Co-Advisor: Eduardo GRAMPÍN
Academic Advisor: Franco ROBLEDO

Defended on March, 2020

Jury:

Reviewers :	Dr. Luciana SALETE BURIOL	-	Universidade Federal do Rio Grande do Sul, (Brasil)
	Dr. Alberto GARCÍA MARTÍNEZ	-	Universidad Carlos III de Madrid (España)
President :	Dr. Ing. Alberto PARDO	-	Universidad de la República (PEDECIBA INFORMÁTICA)
Examinators :	Dr. Francisco BARAHONA	-	IBM Watson Research Center Yorktown Heights, (New York)
	Dr. Ing. Héctor CANCELA	-	Universidad de la República (PEDECIBA INFORMÁTICA)

Contents

Abstract

An Autonomous System (AS) is a group of Internet Protocol-based networks with a single and clearly defined external routing policy, usually under single ownership, trust or administrative control. The AS represents a connected group of one or more blocks of IP addresses, called IP prefixes, that have been assigned to that organization and provides a single routing policy to systems outside the AS.

The Internet is composed of the interconnection of several thousands of ASes, which use the Border Gateway Protocol (BGP) to exchange network prefixes (aggregations of IP addresses) reachability advertisements. BGP advertisements (or updates) are sent over BGP sessions administratively set between pairs of routers.

BGP is a path vector routing protocol and is used to span different ASes. A path vector protocol defines a route as a pairing between a destination and the attributes of the path to that destination. Interior Border Gateway Protocol (iBGP) refers to the BGP neighbor relationship within the same AS. When BGP neighbor relationship are formed between two peers belonging to different AS are called Exterior Border Gateway Protocol (eBGP). In the last case, BGP routers are called Autonomous System Border Routers (ASBRs), while those running only iBGP sessions are referred to as Internal Routers (IRs).

Traditional iBGP implementations require a full-mesh of sessions among routers of each AS. This is due to the *split horizon* rule, under which iBGP routers do not re-advertise routes learned via iBGP to other iBGP peers. As a result, a number of $\frac{n\times(n-1)}{2}$ iBGP sessions is needed for an AS with n routers. *Route Reflection* is used as an alternative to reduce BGP sessions and gain efficiency in CPU and memory usage. With Reflection, one or more routers within the AS are designated as Route Reflectors (RRs) and they are allowed to re-advertise routes learned from an internal peer to other internal peers. The rest of the routers are *clients* of some RRs. A *client* is an iBGP router that the RR will reflect routes to.

The problem studied in this work aims to minimize the number of RRs and the BGP sessions, i.e. how to design an optimal BGP Overlay, in several scenarios: pure IP networks or IP/MPLS, nominal or single link/node failure.

Another contribution is the classification of Internet prefixes into classes, which not only helps designing the BGP overlay but also is an input for traffic engineering when considering MPLS coordinated with BGP routing.

Keywords— Network Overlay Design, Route Reflection, BGP, Internet Routing, Combinatorial Optimization, BGP resilience, Network Resilience, Internet Prefix Classes,d Border Routers.

List of publications issued from this thesis work

This thesis was written using a Swedish PhD style. The chapters are based on the following published papers:

1. "Optimal Route Reflector Topology Design", Cristina Mayr, Claudio Risso, Eduardo Grampín. 10th Latinamerican Networking Conference (LANC '18). ACM, New York, NY, USA, pages 65-72 (2018).

2. "A Combinatorial Optimization Framework for the Design of Resilient iBGP Overlays", Cristina Mayr, Eduardo Grampín, Claudio Risso. 15th International Conference on the Design of Reliable Communication Networks (DRCN'19). IEEE, pages 6-10 (2019).

3. "Designing an Optimal and Resilient iBGP Overlay with extended ORRTD",Cristina Mayr, Claudio Risso, Eduardo Grampín. The Fifth International Conference on Machine Learning, Optimization, and Data Science (LOD'19) Springer International Publishing, pages 409-421 (2019).

4. "A combined BGP and IP/MPLS resilient transit backbone design", Claudio Risso, Cristina Mayr, Eduardo Grampín. 11th International Workshop on Resilient Networks Design and Modeling (RNDM'19)- October 14-16, Nicosia, Cyprus. IEEE, pages 1-8 (2019).

5. "Scalable iBGP and IP/MPLS combined resilient transit backbone design", Cristina Mayr, Claudio Risso, Eduardo Grampín. *Submitted to Computer Networks, Elesevier. September, 2019.*

Agradecimientos

Sin duda, mirando hacia atrás puedo decir que esta tesis es el resultado de un largo esfuerzo en el que varias personas me han motivado, guiado y apoyado.
Debo agradecer muy especialmente a mi director de tesis, el Dr. Claudio Risso, quien en primer lugar aceptó ser mi tutor, y en segundo lugar, además de ser un entusiasta de la Investigación de Operaciones, con su experiencia y conocimiento, y su invaluable consejo ha sido un guía e impulsor esencial de esta tesis.
Al co director de mi tesis, el Dr. Eduardo Grampín quien con su detalladas sugerencias y comentarios, su vasta experiencia y conocimiento y su visión práctica de redes ha contribuido a mi formación doctoral.
Además, ambos han sido fundamentales para armonizar dos visiones que en el mundo de la investigación en el área de redes de hoy en día son bien diferentes.
También me gustaría expresar mi gratitud a mi director académico, el Dr. Franco Robledo, quien desde su época de Director del InCo me motivó, me indujo a pensar: ¿por qué no?, y desde un comienzo me brindó su apoyo para emprender y completar este camino.
A los revisores y al resto de tribunal, quienes me honrarán con la lectura de este trabajo.
Y finalmente a mi querida familia, quienes incondicionalmente me han apoyado en todos los momentos de este camino.

Acknowledgments

Undoubtedly, looking back I can say that this thesis is the result of a long effort in which several people have motivated, guided and supported me.
I must thank my thesis advisor, Dr. Claudio Risso, who firstly accepted me as a student, and secondly, in addition to being an enthusiast of Operations Research, with his experience and knowledge, and his invaluable advice has been an essential guide and driver of this thesis.
To the co-director of my thesis, Dr. Eduardo Grampín whom with his suggestions and detailed comments, his vast experience and knowledge and his practical vision of networks has contributed to my doctoral training.
In addition, both have been fundamental to harmonize two visions that in the world of today's networks research may appear different.
I would also like to show my gratitude to my academic advisor, Dr. Franco Robledo, who from his time as Director of InCo motivated me, led me to think: why not?, and from the very beginning offered me support to undertake and complete this path.
To the reviewers and the rest of the jury, who will honor me by reading this work.
And finally my dear family, who have unconditionally supported me at all times of this path.

Chapter 1

Introduction

Every device requires an IP address to connect to the Internet. IP addresses identify endpoints and they are globally assigned to network operators (i.e. Autonomous Systems or ASes) in blocks of unique and consecutive addresses or network prefixes. The Internet is dependent upon every router in the world to get to know those prefixes, so they can determine routes towards destination for every network packet. Hence, the Internet consists of a set of autonomous systems (ASes) that exchange information about network prefixes accessibility through a standard routing protocol, the Border Gateway Protocol (BGP). Every AS in the world uses BGP to announce those prefixes assigned to it as well as those learned from other ASes. During the process, prefixes attributes are appended or modified by routers to capture how suitable they are from its perspective within the Internet. BGP peering among Autonomous System Border Routers (ASBRs) of neighbor ASes is called External BGP (eBGP), while peering among routers inside the same AS (Internal Routers-IRs) is called Internal BGP (iBGP). In order to make sure that internal transport of BGP info is loop-free (control plane), and internal routing is coherent (loop-free data plane forwarding), the following iBGP advertisement rules must be observed: 1) prefixes learned from an eBGP neighbor can be re-advertised to an iBGP neighbor, and vice versa, and 2) prefixes learned from an iBGP neighbor cannot be re-advertised to another iBGP neighbor. Whatever the rule applied, BGP routers must pre process prefixes attributes prior to relaying them, which potentially biases attributes according to its own placement in the network. A mechanism to prevent from biases within an AS is to implement a full-mesh of BGP sessions among all routers, so all of them can get complete information, but it is quadratic in complexity and causes scalability issues. A widely accepted alternative consists in implementing *Route Reflectors (RRs)*. In this case, one or more routers within the AS are designated as RRs and they are allowed to re-announce routes learned from an IR to other internal peers, while the rest of them act as RR clients of some RR. In this case the number of iBGP sessions scales linearly with the number of routers, but it can introduce reliability and biasing problems which requires a careful design of the iBGP overlay. The main drawbacks of reflection are reliability and biasing, undesirable flaws whose prevention requires a careful design of the iBGP overlay. The design of an optimal BGP overlay for an AS is a known $\mathcal{NP} - Hard$ problem.

This thesis tackles the problem of designing a consistent, reliable and yet optimal iBGP overlay of route reflectors, a problem of notorious academic and industrial relevance that mostly counts heuristic approaches before this one.

The problem is tackled by successive approximations.

Firstly, the research focus on pure IP networks, where IP routing and forwarding are essentially the same. The research work begins with the introduction and modeling of a new combinatorial optimization

problem called *Optimal Route Reflector Topology Design* (ORRTD). The problem goal is to find the minimum number of RRs and BGP sessions in an AS in the nominal case, that is, when there are no failures, and taking into account the relationship between BGP and the costs of the underlying IGP network. Besides, we consider the existence of *classes* of prefixes (groupings of IP prefixes) arriving at different ASBRs of an IP network.

Secondly, the optimization problem considering resilience for either a single router or link failure is analyzed. It can be demonstrated that this problem is $\mathcal{NP} - Hard$. In addition, an enhanced model is proposed to solve it, and ORRTD out-performance compared to other known heuristics is shown. Then a relaxation of the problem called *extended ORRTD* is presented, where ASBRs, as well as the internal routers may be eligible as RRs. Improved results with respect to the previous version of ORRTD are obtained, even in the case of bigger quantities of prefixes classes.

In the second part of the research we propose a model to minimize the RRs when Multiprotocol Label Switching (MPLS) overlay is in place, as MPLS allows traffic engineering, which let handling unexpected congestions, a better bandwidth utilization and route around failed links or nodes. MPLS is a packet-forwarding technology which uses *labels* to make data forwarding decisions, so link protection is encapsulated in the MPLS overlay. In this case an optimization model in two stages is introduced: the first stage applies a variation of ORRTD where resilience against loss of ASBRs adjacencies is addressed, while the second stage introduces MPLS for link optimization. This model allows to perform a sensitivity analysis to determine growth strategy. Finally, a full end- to-end case is presented, considering both control and forwarding plane (BGP over IP/MPLS), where a method to build the classes of prefixes is proposed and applied to a real case, showing that the complexity of millions of BGP announcements can be reduced to just twenty seven prefixes classes. In the MPLS layer, worst case demands are also studied, in each of the ASBRs adjacencies failure scenarios. We show that Traffic Engineering turn capacity deficits up to 110% when using LDP (Label Distribution Protocol) into a 40% slack scenario when coordinating with routing.

1.0.1 State of the art

To arrive to a correct, efficient and reliable network, careful design is needed. Along the papers existing research work regarding how to configure the BGP overlay is explained. Besides, there are several proposals to modify BGP, in order to avoid known possible issues. However, as the Internet is composed of thousands of routers with different capabilities, this last option do not look too attractive, as it would take a long time to adopt another standard to replace BGP. In this section a detailed analysis of those existing heuristics for the standard BGP is presented. First we have to consider the main properties of a full-mesh BGP network:

P1 *Complete visibility* For every external destination, each router picks the same route that it would have picked had it seen the best routes from every other eBGP router in the AS.

P2 *Loop-free forwarding*:1 After the dissemination of eBGP learned routes, the resulting routes (and the subsequent forwarding paths of packets sent along those routes) picked by all routers should be free of deflections and forwarding loops.

P3 *Robustness to IGP failures*: The route dissemination mechanism should be robust to node or link failures and IGP path cost changes.

The BGPSep (Vutukuru et al., 2006) algorithm builds an iBGP configuration that satisfies the complete visibility, loop-free forwarding and robust properties against IGP faults. BGPSep is based on the notion of a graph separator, a (small) set of nodes whose removal partitions a graph into roughly equal-sized connected components. The problem of finding the optimal separator set of a graph is, in general, an $\mathcal{NP} - Hard$ problem. However, fast and practical algorithms for finding small separators are known for many families of graphs. A special case is considered when the network contains interior routers that do not receive any external routes. This modified algorithm is inefficient if the number of egress routers is very small. Another case is for a backbone network and it recommends to run BGPSep on just the backbone routers to establish a route reflector (RR) hierarchy over the backbone alone. Then configure the backbone routers in each PoP as route reflectors for the access routers in the PoP.

BGPSep_D (Zhao et al., 2006a) improves BGPSep considering that in an IGP graph, vertices whose degree is one will have full visibility if they are clients of their only neighbor (assuming the neighbor has full visibility). BGPSep_D gradually removes the pendant vertices whose degree is one, that usually exist in IGP topologies of large ASes. Then it applies BGPSep, and the authors claim that the maximum degree can be reduced from 9% to 50% and the IBGP sessions by 10 to 50%, which supposedly results in less BGP sessions.

The BGPSep_S (Zhao et al., 2006b) algorithm constructs an iBGP configuration taking into account the degree of the vertices, the vertex separators and the shortest paths between the vertices in the underlying IGP graph.

Both BGPSep_D and BGPSep_S use the notion of signaling chain between two routers A and B: it is defined as a set of routers $(A = R_0)R_1, R_2, ..., R_r, B(= R_{r+1})$ such that, for $i = 1...r$, (i) R_i is a route reflector and (ii) at least one of R_{i+1} or R_{i1} is a route reflector client of R_i. The authors claim that the iBGP configuration generated with BGPSep_S decreases the maximum degree of the topology between 27% and 68% compared to the full-mesh. In addition, if a separator set can be found in an IGP graph, then any path that begins in one component and ends in a different component must pass through one or more routers in the separator set. If an iBGP topology is constructed by establishing a full-mesh between the routers of the separator set, building a full-mesh configuration within each connected component and creating other necessary iBGP sessions so that there is a shortest path signaling chain between any router in a component and a router within the separator set, then there will be a shortest signaling chain between all pairs of vertices. If one or more vertices of the related components are taken and added to the set of vertices of the separator, a super-set of the separator set is obtained, which remains a separator set.

Bates (Pelsser et al., 2010) recommends configuring the iBGP topology with one or more RRs for each point of presence (PoP - Point of Presence) in the network. All routers in the PoP are clients of the RRs of that PoP. Besides, they recommend a full-mesh of iBGP sessions among all routers in a PoP. BatesY is a variant of the Bates heuristic where the most connected router in each PoP is selected as RR. Each router is a client of the RR in its PoP. A full-mesh of iBGP sessions is established among the RRs of the different PoPs. Finally, a full-mesh of iBGP sessions is established among the routers of each PoP. BatesZ is another variant of the Bates heuristic. In order to obtain redundancy, two route reflectors are selected in each PoP. These two routers are the most connected in the PoP. All routers in the PoP are iBGP clients of the two RRs. A full-mesh of iBGP sessions is configured between the RRs. It does not require full-mesh sessions between the routers (those not selected as RRs) of the PoP.

The Zhang (Zhang and Bartell, 2003) heuristic is characterized by having multiple levels of route reflectors, thus generating a hierarchical iBGP configuration. Routers that are clients of top-level route reflectors can be RRs of routers that are at lower levels. Zhang (Zhang and Bartell, 2003) does not specify the number of RRs to be used.

The Optimal algorithm (Buob et al., 2008) - based on the Benders decomposition framework- is another heuristic that fulfills the following requirements: Fm-optimality, correctness, reliability, robustness and scalability. They consider both the nominal case and the failure case, by adding satellite problems. A mixed-integer program (MIP) is proposed to minimize the number of IGP hops and BGP sessions.

Optimal Route Reflector Topology Design (ORRTD) is the proposed solution in the present research. The ORRTD construction process takes into account the BGP algorithm which has a set of ordered break-even rules, until the step in which the IGP metric is considered. The outstanding property of ORRTD is that the optimization considers that different prefixes may arrive at each border router, i.e., the ASBRs receive less than the total set of Internet prefixes. The optimal criteria in ORRTD is to use the minimum number of route reflectors and sessions, maintaining correctness and full-mesh optimality (Buob et al., 2008; Griffin and Wilfong, 2002; Vissicchio et al., 2012), assuming that all prefixes matching a common gateway or a set of equally preferred gateways are clustered into classes of prefixes (or labels). This is in fact a realistic consideration, since it depends on Internet Service Provider's policies.

The second distinguishing characteristic of ORRTD is that it can be coordinated with the forwarding plane in an IP/MPLS (Multiprotocol Label Switching) scenario. Nowadays most Internet providers implement their backbones by combining IP routing with MPLS for QoS-aware traffic forwarding. MPLS forwarding incorporates traffic engineering and more efficient fail-over mechanisms. The contribution of the study is on proposing an optimal and yet resilient topology design for an IP/MPLS Internet backbone, which takes advantage of traffic engineering features to optimize the demands, maintaining the aforementioned iBGP overlay optimality.

1.0.2 Complementary experimental results

Throughout the study, experimental outcomes with different network topology designs are shown. In this section the following additional information is introduced:

a) resolution time results not presented in the papers, when applying ORRTD to different topologies

b) a complete alternative scenario when applying ORRTD coordinated with IP/MPLS traffic engineering to a real world ISP case.

Resolution time study

For the first case, in table 1.1 the solver (which in this case is IBM ILOG CPLEX(R) Interactive Optimizer version 12.6.3) resolution times are shown for several networks and different quantity of prefixes classes, for the IP pure case. In this case resilience is considered for both link and node failures. This is the meaning of the table columns: the first one is the name of the network, then the number of border routers and internal routers, the number of classes of prefixes, the resulting quantity of route reflectors when applying ORRTD, the time spent by the solver to calculate them, and finally the number of BGP sessions got with ORRTD, and the number of BGP sessions if a full mesh of BGP sessions is implemented.

Note that the resolution times are very low, but prior processing of the information is needed to get a linear programming specification, such as the construction of the classes of prefixes and the construction of the auxiliary graphs as described in the next chapters. So those times should also be considered. Consequently, most of the hard processing is done previously, when pre-processing the BGP messages

Table 1.1: extended ORRTD Resolution time (sec)

						# BGP Sessions	
Network	**# BRs**	**#IRs**	**#PFs**	**#RRs**	**Time**	**eORRTD**	**FM**
AB5	3	5	4	3	0.02	18	28
AB10	3	10	4	2	0.02	23	78
Abilene	3	8	4	2	0.03	16	55
Cernet2	4	37	4	3	0.33	112	820
Forthnet	3	56	4	2	15.47	114	1711
Garr2	7	48	4	7	0.05	246	1485
Jgn2Plus2	6	11	4	5	0.31	45	136
SwitchL3	12	30	4	3	0.52	79	861
TtNew	6	94	4	6	76.05	532	4950
TtNew20	20	80	4	18	16.11	1152	4950
UniC	3	24	4	3	0.83	65	351
Uran	5	18	4	4	0.73	48	253
UsCarrier1	3	154	4	3	285.27	461	12246
WideJpn	11	19	4	5	0.36	77	435
AB5	3	5	10	3	0.05	18	28
AB10	3	10	10	2	0.23	24	78
Abilene	3	8	10	3	0.22	25	55
Cernet2	4	37	10	3	0.58	112	820
Forthnet	3	56	10	3	29.25	225	1711
Garr2	7	48	10	7	0.92	272	1485
Jgn2Plus2	6	11	10	5	0.31	47	136
SwitchL3	12	30	10	3	1.63	91	861
TtNew	6	94	10	6	228.5	561	4950
TtNew20	20	80	10	18	154.72	1470	4950
UniC	3	24	10	3	0.47	65	351
Uran	5	18	10	4	1.5	67	253
UsCarrier1	3	154	10	3	242.11	460	12246
WideJpn	11	19	10	5	0.38	89	435
AB5	3	5	50	3	0.25	19	28
AB10	3	10	50	2	0.23	24	78
Abilene	3	8	50	3	0.45	26	55
Cernet2	4	37	50	3	4.05	112	820
Forthnet	3	56	50	3	25.01	171	1711
Garr2	7	48	50	7	9.13	265	1485
Jgn2Plus2	6	11	50	6	0.66	79	136
SwitchL3	12	30	50	4	5.42	136	861
TtNew	6	94	50	6	1486.74	573	4950
TtNew20	20	80	50	18	1150.3	1430	4950
UniC	3	24	50	3	1.61	65	351
Uran	5	18	50	5	3.38	78	253
WideJpn	11	19	50	5	1.28	113	435

to reduce the amount of data (which can take a few hours) in at least one order, and after that, when building the tree of prefixes to deduce the classes of prefixes (processing about 750 thousands prefixes takes about 40 minutes). The steps to be considered are the following:

- run the BGP Path Selection Algorithm for every ASBR, using as input the Rib-In databases: From an original set of over 9 million eBGP updates, the BGP decision process produces around 800 thousand prefixes.
- build a prefixes tree and discard prefixes of low specificity, because when a range of IP addresses is spanned by more than one prefix/mask entry, the router always chooses the most specific as its gateway. This step produced around 750 thousand prefixes.
- group the Internet prefixes into classes of prefixes. This step produced some dozens classes. This and the previous step are done simultaneously and take about 40 minutes.
- build the auxiliary graphs based on the network topology and the prefixes classes.
- build the optimization model. This step and the previous one take some seconds, depending on the graph size and the number of prefixes classes.
- run the solver to obtain the minimum number of route reflectors and BGP sessions.
- Estimate the traffic for each prefixes class by using statistics per source (snmp and netflow toos)
- In the case of IP/MPLS, run the optimization process for the traffic engineering (nominal and worst case), which is explained in detail in the next section.

Full Description of the Second Scenario - ORRTD and IP/MPLS

In the paper *"Scalable iBGP and IP/MPLS combined resilient transit backbone design"* a future network to be implemented by a regional ISP is analyzed. The network design is shown in Fig. 1.1. Let's call this network N_A. On the right hand, in Fig. 1.2 there is an alternative network which will be studied in this section. Let's call this network N_B.
N_A has a node in Porto Alegre (PA) that provides alternative paths to connect to Brazil, complementing those already existing, namely: a submarine cable between Maldonado and Santos and a set of terrestrial connections through Argentina, with a higher failure rate per unit of time, but above all, of very high aggregate propagation delay (i.e. long geographical distances). Under those assumptions it was difficult to reach some of the tolerated delay limits between countries defined by the ISP. The alternative scenario, N_B, presumes another connectivity scheme with Brazil, where instead of the path through PA, an independent submarine cable capacity arriving at Santos (SS) from Toninas (TS) is considered. Additional changes can be highlighted: there is a new submarine cable between TS and Parada 5 (P5); PA node disappears and a third node in San Pablo (S3) is introduced, where peering is done with Telecom, a provider with which there was no direct connection on the N_A network. Finally, a second connection is provided with IX.br, an IXP (Internet Exchange Point) operated by NIC.br. An Internet exchange point (IXP) is a physical location through which Internet Service Providers (ISPs) and Content Delivery Networks (CDNs) connect with each other. In this way companies can shorten their path to the transit coming from other participating networks, thereby reducing latency, improving round-trip time, and potentially reducing costs.

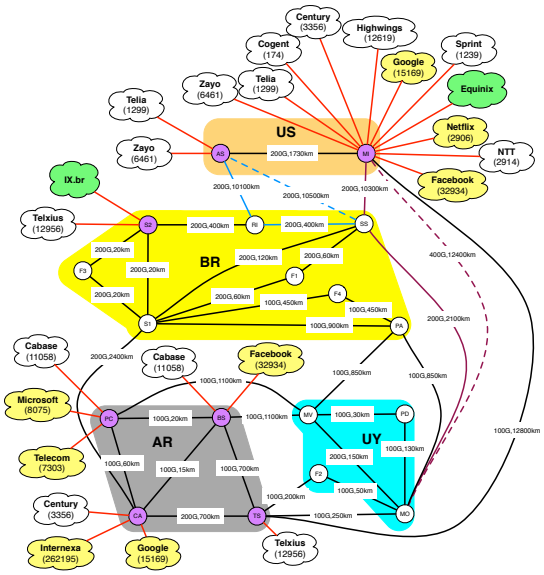


Figure 1.1: 1st scenario (N_A)

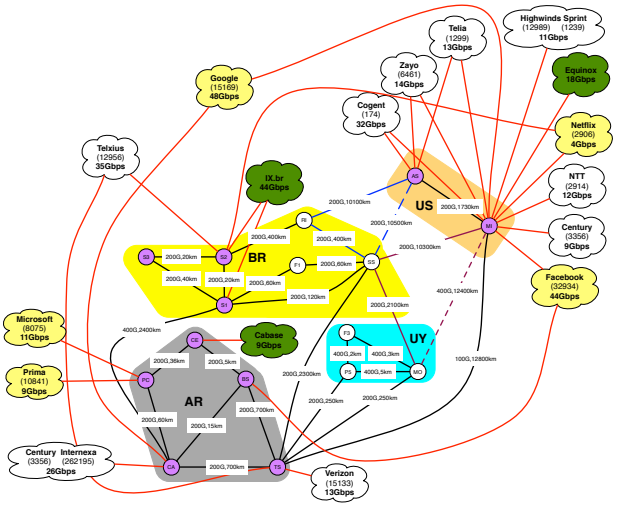


Figure 1.2: 2nd scenario (N_B)

The steps to be considered to design the MPLS overlay and recur to traffic engineering for the efficient use of the network are the following:

- Traffic estimation is adjusted by destination, seeking to reproduce traffic between countries. A representative of each class is used in an emulation environment to implement the previous iBGP overlay. Traffic over that environment replicates the current average traffic (nominal case).
- Adjacency losses are emulated to estimate the associated traffic matrices in the virtual environment.
- The *worst case* scenario is calculated taking as a reference the maximum traffic between each pair of nodes among all faults. CDNs changes may happen unexpectedly, without coordination, and a resilient network must be prepared to support it.
- The delay limits between nodes are defined by balancing the design objectives with the possibilities of the network in the case of potential simple failures.
- To maintain consistency with what would have been the iBGP optimality, those limits should be close to the IGP values, unless it results in congestion of some links.
- The network traffic engineering is optimized to find the independent pairs of paths, which must also comply with the delay limits, while ensuring that there is no congestion even after each simple failure in the links.

Label Distribution Protocol (LDP) is the simplest mechanism to signal paths in MPLS and is based on replicating the paths that pure IP routing would have chosen by the IGP. Although known for its limited efficiency in the use of resources, LDP is still popular for its simplicity and parallelism with classic IP routing. BGP and the LDP tunnels are aligned, since both use the same metric. This work explores the advantages of using optimized traffic engineering instead of LDP, by choosing physically independent

paths (primary and secondary paths), administratively set (signaled with RSVP-TE) to comply with a set of Quality of Service (QoS) and additional restrictions such as seeking to meet delays limits between countries, and minimizing congestion to any physical failure.

Starting with the eBGP route advertisements of the current network it has been simulated what the universe of eBGP updates in the network would be, and based on them the optimal iBGP overlay is shown in Fig. 1.3. For clarity, the full-mesh of adjacencies among reflectors, which is implicit, has been omitted in the figure. The fictitious nodes (F1 and F2) have also been omitted as they were included just for the purpose of supporting physically independent connections between the same nodes. Notice that the overlay to be designed must be full-mesh optimal not only in the nominal scenario, but also when a total loss of the adjacencies of any node in the network occurs. It is first observed that an overlay with these characteristics must have at least 6 reflectors, and not less than 40 iBGP sessions. Twenty-five of them are seen in Figure 1.3; the remaining fifteen constitutes the full-mesh among the resulting RRs. This BGP overlay was found with exact solver (IBM CPLEX) and it is optimal, but not necessarily unique. This means there cannot be another overlay with these characteristics (resilient and full-mesh optimal) with less than six RRs or with less than forty sessions, but there can be a different configuration (i.e. other nodes playing the RR role) with those same values.

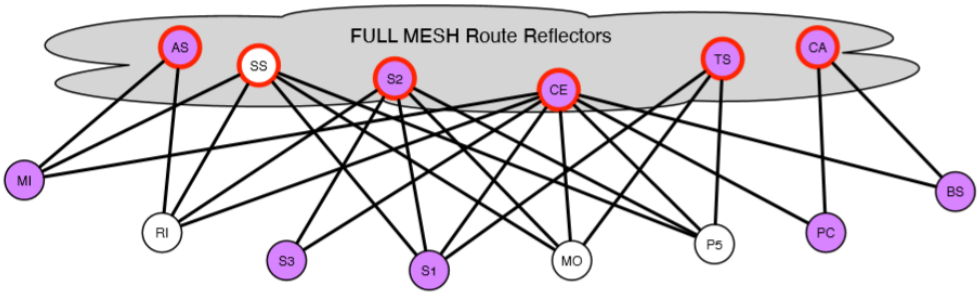


Figure 1.3: Optimal iBGP Overlay for the 2nd scenario

As already described, structuring the BGP prefixes into equivalence classes is an essential component in the preparation of the combinatorial optimization problem to solve the iBGP overlay. Starting with over 9 million updates that this network would receive, the BGP selection algorithm in the first place, and an ad-hoc filtering tool (developed as part of this work) as a second step, reduced that number to 730043 essential updates, that is, those that concentrate all the routing information that the international ISP network on this scenario needs in order to have optimal resilient connectivity to the world. The number is still huge to be considered in an exact combinatorial formulation. Another important step in the process is the clustering of the resulting prefixes into equivalence classes defined by the combination of border routers through which the associated updates enter. The idea is that guaranteeing the optimality of only one of the prefixes learned from that combination of ASBRs, implies the optimality for other prefixes entering through that same combination. This idea allows to reduce the 730043 prefixes to 66 equivalence classes, shown in Table 1.2. The network N_B could potentially accept more than thousand prefixes classes ($2^n - 1$, where n is the number of ASBRs). Nonetheless, the combined results of the emulation environment to pre-process the prefixes, and the construction of the prefixes tree produced just 66 classes of prefixes.

Table 1.3 presents the quantity of prefixes classes and the solver time for the IP/MPLS case for the topologies N_A and N_B. A comparison of the results obtained by optimizing only the number of RRs with

Table 1.2: Equivalence Classes of Prefixes

Class Id	Combination of ASBRs	#Prefixes	Cumulated %
Class 1	TS CA S2 MI S1	316317	43,33%
Class 2	TS CA S2 AS MI S1	295723	83,84%
Class 3	TS CA S2 MI S1 S3 CE	48677	90,50%
Class 4	TS CA S2 AS MI S1 S3 CE	35296	95,34%
Class 5	MI	8469	96,50%
Class 6	S2 S1	6752	97,42%
Class 7	TS CA S2 AS MI S1 CE	4033	97,98%
Class 8	PC	3111	98,40%
Class 9	TS PC CA S2 AS MI S1 S3 CE	2715	98,77%
Class 10	CE	1637	99,00%
Class 11	TS S2 MI	959	99,13%
Class 12	AS MI	854	99,25%
Class 13	TS S2	779	99,35%
Class 14	TS CA S2 MI S1 CE	769	99,46%
Class 15	S2 S1 CE	602	99,54%
Class 16	TS S2 S1	434	99,60%
Class 17	TS PC CA S2 AS MI S1 CE	424	99,66%
Class 18	CA CE	289	99,70%
Class 19	TS PC CA S2 AS MI S1	273	99,74%
Class 20	CA MI	265	99,77%
Class 21	TS S2 AS MI	239	99,80%
Class 22	TS PC CA S2 MI S1 S3 CE	180	99,83%
Class 23	CA AS MI	172	99,85%
Class 24	TS CA S2 S1	163	99,88%
Class 25	TS S2 S3 CE	121	99,89%
Class 26	TS S2 MI S1	110	99,91%
Class 27	CA S2 AS MI S1	97	99,92%
Class 28	TS S2 S1 S3 CE	79	99,93%
Class 29	TS CA S2 AS MI	67	99,94%
Class 30	TS S2 AS MI S1	62	99,95%
Class 31	TS PC CA S2 MI S1 CE	60	99,96%
Class 32	CA S2 MI S1	54	99,96%
Class 33	CA S2 S1	52	99,97%
Class 34	CA MI S3 CE	24	99,97%
Class 35	CA AS MI S3 CE	23	99,98%
Class 36	TS CA S2 S1 S3 CE	22	99,98%
Class 37	TS PC CA S2 MI S1	16	99,98%
Class 38	TS BS CA S2 MI S1	16	99,99%
Class 39	CA S2 S1 CE	12	99,99%
Class 40	TS CA S2 MI S3 CE	10	99,99%
Class 41	CA	9	99,99%
Class 42	TS S2 AS MI S1 S3 CE	9	99,99%
Class 43	TS BS CA S2 AS MI S1	8	99,99%
Class 44	TS CA S2 MI	6	99,99%
Class 45	S2 MI S1	6	99,99%
Class 46	CA S2 AS MI S1 S3 CE	6	99,99%
Class 47	TS CA S2 AS MI S3 CE	5	99,99%
Class 48	TS S2 MI S1 S3 CE	4	100,00%
Class 49	S2 AS MI S1	4	100,00%
Class 50	TS BS CA S2 MI S1 CE	3	100,00%
Class 51	CA S2 MI S1 S3 CE	3	100,00%
Class 52	TS BS CA S2 AS MI S1 CE	3	100,00%
Class 53	TS CA S2	2	100,00%
Class 54	TS S2 MI S3 CE	2	100,00%
Class 55	TS CA S2 CE	2	100,00%
Class 56	TS CA S2 S1 CE	2	100,00%
Class 57	CA S2 AS MI S1 CE	2	100,00%
Class 58	CA AS MI CE	2	100,00%
Class 59	TS BS S2 MI CE	1	100,00%
Class 60	TS BS S2 MI	1	100,00%
Class 61	TS BS CA S2 MI S1 S3 CE	1	100,00%
Class 62	PC CA MI S3 CE	1	100,00%
Class 63	TS S2 S1 CE	1	100,00%
Class 64	TS S2 AS MI S3 CE	1	100,00%
Class 65	PC CA S2 S1 CE	1	100,00%
Class 66	PC S2 S1 CE	1	100,00%

those obtained when optimizing both the number of RRs and the BGP sessions is also shown. Note that having less BGP sessions has a positive effect on administration time, and Rib-In table size for the involved routers.

Table 1.3: ORRTD and Prefixes classes - IP/MPLS case

Network	**#BRs**	**#IRs**	**Max classes**	**#Clases**	**#RRs**	**BGP sessions**	**Time (sec)**
N_A-only RRs	7	11	127	27	6	42	0,55
N_A-RRs and sessions	7	11	127	27	6	22	0,66
N_B-only RRs	10	6	1023	66	6	62	1,36
N_B-RRs and sessions	10	6	1023	66	6	40	4,31

Similar to the case examined in the scenario described in *"Scalable iBGP and IP/MPLS combined resilient transit backbone design"* for N_A, few classes concentrate most of the prefixes: the first ten classes concentrate more than 90% of the prefixes to be learned by the network, and a comparable number of classes concentrate most of the traffic. Table 1.4 presents the estimation of nominal peak traffic, according to the new adjacency map (network N_B) and available statistics.

Table 1.4: Peack traffic - nominal case

	US	**BR**	**AR**	**UY**
US	0	24	24	93
BR	24	0	15	49
AR	24	15	0	159
UY	93	49	159	0

Table 1.5: Peack traffic - worst case

	US	**BR**	**AR**	**UY**
US	0	36	36	166
BR	36	0	31	106
AR	36	31	0	206
UY	166	106	206	0

The nominal peak traffic is the maximum expected traffic when all adjacencies in the network are operational, and when content providers balance their traffic following the distribution revealed in the current statistics. The nominal traffic for the network N_B now is 364Gbps, almost the same value as the 346Gbps of the network N_A, as shown in Table 1.6.

Simulations and optimizations are performed using the nominal matrix, and the so-called *worst-case matrix*. This matrix captures different traffic variants, exploring combinations of adjacency losses of different types and simulating how traffic would be redistributed. Subsequently, the highest traffic between each pair of nodes is taken and a new matrix is built with the maximum in each case. The worst-case matrix for N_A added up 495 Gbps, 43% more traffic than in the nominal case. The worst case matrix for network N_B totals 581 Gbps and is presented in Table 1.5.

Table 1.6: Comparison of traffic

	Network N_A	**Nework N_B**
nominal traffic	346Gb	364Gb
worst-case	495Gb	581Gb
increment	43%	60%

The new worst-case matrix is significantly higher (almost 60%) than the nominal one. Much of that difference is explained by the IX.br. In the new scenario, the IX.br connects with S1 and S2. In the case of network N_A IX.br is only connected to S2. Given the eventual loss of that adjacency, its 44 Gbps would be received from regional transit providers, in a distributed manner, which would not significantly

Table 1.7: Target Distances among Countries

	US	BR	AR	UY
US	1730	12430	14565	14133
BR	12430	560	3201	2712
AR	14565	3201	796	1048
UY	14133	2712	1048	8

Table 1.8: Feasible Distance Bounds

	US	BR	AR	UY
US	1730	12430	15290	14840
BR	12430	560	3201	2840
AR	15290	3201	796	1048
UY	14840	2840	1048	8

increase the peak in any of the cases. In the new topology and under the design logic for the worst-case, the network must be prepared to receive 44 Gbps for both S1 and S2, because if the adjacency is lost in one of those nodes, all IX.br traffic would be received by the other node. It is noted that as a result of the worst-case design, the network is prepared for these failure scenarios. Unlike the network N_A, where in case of failure, IX.br traffic would pass through more than one AS before reaching destination, with the consequent degradation in performance, in N_B, traffic is received by a node that is 20km away from the previous one, which represents less than 1 msec of additional delay, over a network that is also prepared not to congest in the event of failure. The worst-case traffic scenario actually generates much better quality solutions, where several combinations of events or failures could go unnoticed by most users.

The model supports setting specific delay limits for routes between any pair of nodes. For practical reasons, the ISP decided to set limits between the countries. The underlying premise is that, given a tunnel between two nodes from different countries, both the point-to-point delay of the primary and secondary paths do not exceed the threshold between those countries. The goal is that the values of these thresholds are those of the nodes that are at a greatest distance between those countries in the event of a single link failure. The values for the network N_B are those presented in Table 1.7.

Distance is used to represent *delay*. Table 1.7 should be read as follows: if all possible physical link failures are simulates one by one, the shortest active path between the nodes of US and BR with greatest distance is 12430km. It is clear that as a reference of delay between countries, a resilient solution with values below those values is not possible. As expected, the distances within the countries are significantly shorter than the international distances.

As previously indicated, the values in Table 1.7 are the theoretical lower limits for delays between countries. They may not be reachable when implementing a redundancy scheme with active / standby traffic engineering, because the latter requires physical independent paths, while some links of the shorter paths may be repeated in several failure scenarios. However, Table 1.8, which has the effective values used for traffic engineering, is nearly the same, except in three of them, and in those cases the differences are under 5%, which are always imperceptible in terms of milliseconds of additional delay. As the first practical result on traffic engineering, it should be noted that: the network N_B with the nominal demand (Table 1.4) and the delay limits (Table 1.8), is feasible and with a slack of capacity that would allow it to increase demand evenly by up to 25.79% without congesting any internal link in case of a simple physical failure. If instead of the nominal demand the worst case is considered, the network is no longer feasible, although by a small margin. In fact, a reduction of only 3% of the traffic would have been enough to make the instance feasible. In any case, the problem is solved with a 100 Gbps expansion in the capacity of the TS-MO and TS-P5 cables. From the previous extension, which represents 0.33% in terms of the total kilometers of 100Gbps links, the resulting network has a minimum slack of 11% in the worst combination of a link and adjacencies failures case. The full detail is shown in Table 1.9.

Now compare the results obtained when applying traffic engineering with those obtained by using

Table 1.9: Minimum slacks for the worst case demand

Node 1	Node 2	Length	Capacity	Min - slack - worst case	
AS	MI	1730	200	62	31%
AS	RI	10100	200	142	71%
AS	SS	10500	200	78	39%
MI	SS	10300	200	124	62%
MI	TS	12800	100	30	30%
MI	MO	12400	400	234	59%
RI	SS	400	200	154	77%
RI	S2	400	200	131	66%
SS	S1	120	200	82	41%
SS	F1	60	200	70	35%
SS	TS	2300	200	38	19%
SS	MO	2100	200	64	32%
S1	S2	20	200	107	54%
S1	S3	40	200	107	54%
S1	F1	60	200	70	35%
S1	CA	2400	400	333	83%
S2	S3	20	200	55	28%
BS	CA	15	200	45	23%
BS	TS	700	200	42	21%
BS	CE	5	200	156	78%
CA	PC	60	200	156	78%
CA	TS	700	200	42	21%
PC	CE	36	200	165	83%
TS	MO	250	300	61	20%
TS	P5	250	300	44	15%
MO	P5	5	400	226	57%
MO	F2	3	400	42	11%
P5	F2	2	400	42	11%

LDP. The network N_B with the nominal demand of Table 1.4 is no longer feasible, in the sense that it suffers from congestion. The network capacities were increased by iteratively repeating the process of selective expansion of these links until getting a network without congestion in the event of any simple failure, when all tunnels use the active path of the shortest possible distance. The result is shown in Table 1.10.

Table 1.10: LDP slacks after updating capacities

Node 1	Node 2	Length	Capacity	Min - slack - worst case	
AS	MI	1730	200	62	31%
AS	RI	10100	200	142	71%
AS	SS	10500	200	78	39%
MI	SS	10300	300	108	36%
MI	TS	12800	200	26	13%
MI	MO	12400	400	234	59%
RI	SS	400	200	154	77%
RI	S2	400	200	131	66%
SS	S1	120	200	82	41%
SS	F1	60	200	70	35%
SS	TS	2300	300	28	9%
SS	MO	2100	300	28	9%
S1	S2	20	200	107	54%
S1	S3	40	200	107	54%
S1	F1	60	200	70	35%
S1	CA	2400	400	333	83%
S2	S3	20	200	55	28%
BS	CA	15	200	45	23%
BS	TS	700	200	42	21%
BS	CE	5	200	156	78%
CA	PC	60	200	156	78%
CA	TS	700	200	42	21%
PC	CE	36	200	165	83%
TS	MO	250	300	94	31%
TS	P5	250	500	82	16%
MO	P5	5	400	226	57%
MO	F2	3	400	42	11%
P5	F2	2	400	42	11%

The most compromised links in the new configuration are SS-TS and SS-MO, reaching in some failure scenarios a traffic that is 91% (9% slack) of capacity. In general, regarding the performance of the solutions it can be appreciated that traffic engineering with capacities shown in Table 1.9 and LDP with capacities shown in Table 1.10 are equivalent. The difference lies in the reference cost in both cases: while an additional investment of 0.33% (kilometers of 100 Gbps links) was required to arrive at Table 1.9, the network capacity expansions of Table 1.10 represent 18.7% over the original network. This verifies the theoretical advantage of off-line and centralized traffic engineering over dynamic protocols.

1.1 Structure of the Thesis

This thesis follows the Swedish style, and it is organized in three parts.

All the chapters have been ordered according to the logic of studying the optimal BGP overlay design for the nominal case, then reliability in pure IP networks, and finally the case of a resilient iBGP and IP/MPLS transit backbone design.

1. Part I introduces the iBGP Overlay design problem and proposes an integer programming approach to solve the problem for the nominal case where only internal routers (IRs) can be route reflectors, which was called *Optimal Route Reflector Topology Design*. After that, in Chapter 3 constrains are introduced to contemplate reliability when one node or link fails, and the concept of *prefixes classes* is introduced. Chapter 4 presents a relaxation, where ASBRs are allowed to act as RRs.

2. In Part II the focus is on optimization of both control and forwarding plane in a combined BGP and IP/MPLS scenario. Chapter 5 introduces the problem and the conceptual model. In Chapter 6 the prefixes classes construction process and the case when applying the two stage model to a real world transit Internet Service Provider (ISP) which uses BGP over IP/MPLS is thoroughly examined.

3. In Part III conclusions and future research work are presented.

Each chapter includes a corresponding peer-reviewed article. They are all accepted and published (except the article from Chapter 6 which is submitted and at the time of writing this thesis there was no acceptance notification yet).

1.2 Bibliography

MiniNExT (Mininet ExTended). https://github.com/USC-NSL/miniNExT. Accessed 2018-06-01.

Quagga Routing Suite. https://www.quagga.net/. Accessed 2018-06-01.

Afek, Y., Ben-Shalom, O., and Bremler-Barr, A. (2002). On the structure and application of bgp policy atoms. In *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, pages 209–214, New York, NY, USA. ACM.

Andersson, L., Doolan, P., Feldman, N., Fredette, A., and Thomas, B. (2001). LDP Specification. RFC 3036 (Proposed Standard). Obsoleted by RFC 5036.

Awduche, D. (2006). GP/MPLS IP Virtual Private Networks (VPNs). RFC4364 (Proposed Standard), Updated by RFCs 4577, 4684, 5462.

Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and Swallow, G. (2001). RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209 (Proposed Standard). Updated by RFCs 3936, 4420, 4874, 5151, 5420, 5711, 6780, 6790, 7274.

Awduche, D., Malcolmm, J., Agogbua, J., O'Dell, M., and McManus, J. (1999). Requirements for Traffic Engineering Over MPLS. RFC2702 (Informational Standard).

Awduche, D. O. (1999). MPLS and Traffic Engineering in IP Networks. *IEEE Communications Magazine*, 37(12):42–47.

Bashandy, A., Filsfils, E., and Mohapatra, P. (2018). BGP Prefix Independent Convergence.

Basu, A., Chih-Hao, L. O., Rasala, A., Shepherd, B., and Wilfong, G. (2002). Route Oscillation in iBGP with Route Reflection. In *Proceedings of SIGCOMM 2002 Conference in Pittsburgh*, pages 235–247, New York, NY, USA. ACM.

Bates, T., Chen, E., and Chandra, R. (2006). GP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP. RFC4456.

Broido, A. , k. (2001). Analysis of routeviews bgp data: policy atoms, cooperative association for internet data analysis. In *NRDM workshop Santa Barbara*, San Diego Supercomputer Center, University of California, San Diego. CAIDA.

Buob, M. O., Lambert, A., and Uhlig, S. (2016). IBGP2: A scalable iBGP redistribution mechanism leading to optimal routing. In *35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016*, volume 2016-July, pages 1–9. IEEE.

Buob, M. O., Meulle, M., and Uhlig, S. (2007). Checking for optimal egress points in iBGP routing. In *2007 6th International Workshop on Design and Reliable Communication Networks*, pages 1–8.

Buob, M.-O., Uhlig, S., and Meulle, M. (2008). Designing optimal iBGP route-reflection topologies. In *Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet*, NETWORKING'08, pages 542–553, Berlin, Heidelberg. Springer-Verlag.

Chen, R., Shaikh, A., Wang, J., and Francis, P. (2011). Address-based Route Reflection. In *Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies*, CoNEXT '11, pages 5:1–5:12, New York, NY, USA. ACM.

Cittadini, L., Vissicchio, S., and Di Battista, G. (2010). Doing don'ts: Modifying BGP attributes within an autonomous system. In *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, pages 293–300.

Cristel Pelsser, Akeo Masuda, and Kohei Shiomoto (2010). A novel internal BGP route distribution architecture. In *IEICE General Conference*.

Dakshayini, S. S. M. (2016). Effect of route reflection on iBGP convergence and an approach to reduce convergence time. In *International Journal of scientific research and management (IJSRM)*, volume 4.

Elmokashfi, A., Kvalbein, A., and Dovrolis, C. (2012). BGP churn evolution: A perspective from the core. *IEEE/ACM Transactions on Networking*, 20(2):571–584.

F. Skivee, S. B. and Leduc, G. (2006). A scalable heuristic for hybrid IGP/MPLS traffic engineering - Case study on an operational network. In *14th IEEE International Conference on Networks*, pages 1–6.

Farrel, A., Vasseur, J.-P., and Ash, J. (2006). A Path Computation Element (PCE)-Based Architecture. RFC4655 (Informational Standard).

Feamster, N., Balakrishnan, H., Rexford, J., Shaikh, A., and van der Merwe, J. (2004). The case for separating routing from routers. In *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, FDNA '04, pages 5–12, New York, NY, USA. ACM.

Feamster, N. and Rexford, J. (2007). Network-wide prediction of BGP routes. *IEEE/ACM Transactions on Networking*, 15(2).

Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J., and True, F. (2001). Deriving traffic demands for operational IP networks: methodology and experience. *IEEE/ACM Transactions on Networking*, 9(3):265–279.

Flavel, A. (2009). *BGP, Not As Easy As 1-2-3*. PhD thesis, University of Adelaide, Australia.

Flavel, A. and Roughan, M. (2009). Stable and flexible iBGP. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pages 183–194, New York, NY, USA. ACM.

Fuller, V., Li, T., Yu, J., and Varadhan, K. (1993). Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy. RFC1519.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability. A Guide to the Theory of NPCompleteness.* Freemann.

Godán, F., Colman, S., and Grampín, E. (2016). Multicast BGP with SDN control plane. In *2016 7th International Conference on the Network of the Future (NOF)*, pages 1–5.

Grampin, E. and Serrat, J. (2005). Cooperation of control and management plane for provisioning in MPLS networks. In *9th IFIP/IEEE International Symposium on Integrated Network Management*, pages 281–294.

Griffin, T. G. and Wilfong, G. (2002). On the Correctness of IBGP Configuration. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '02, pages 17–29, New York, NY, USA. ACM.

Gutiérrez, E., Agriel, D., Saenz, E., and Grampín, E. (2014). RRLOC: A tool for iBGP Route Reflector topology planning and experimentation. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, volume 3, pages 1–4.

Khosla, R., Fahmy, S., and Hu, Y. C. (2011). Bgp molecules: Understanding and predicting prefix failures. In *2011 Proceedings IEEE INFOCOM*, pages 146–150.

Klockar, T. and Carr-Motyckov, L. (2004). Preventing oscillations in route reflector-based I-BGP. In *Proceedings of 13th International Conference on Computer Communications and Networks - ICCCN 2004*, pages 53–58. IEEE.

Knight, S., Nguyen, H., Falkner, N., Bowden, R., and Roughan, M. (2011). The Internet Topology Zoo. *IEEE Journal on Selected Areas in Communications*, 29(9):1765 –1775.

Lantz, B., Heller, B., and McKeown, N. (2010). A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pages 19:1–19:6, New York, NY, USA. ACM.

Li, T. and Smit, H. (2008). IS-IS Extensions for Traffic Engineering. RFC 5305 (Proposed Standard).

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

Long, H. e. a. (2019). OSPF Traffic Engineering (OSPF-TE) Link Availability Extension for Links with Variable Discrete Bandwidth. RFC 8330 (Standards Track).

Marques, P., Fernando, R., Chen, E., Mohapatra, P., and Gredler, H. (2017). Advertisement of the best external route in BGP, draft-ietf-idr-best-external-05.

McPherson, D., Gill, V., Walton, D., and Retana, A. (2002). Border Gateway Protocol (BGP) Persistent Route Oscillation Condition. RFC 3345.

Mereu, A., Cherubini, D. , F. and Frangioni, A. (2009). Primary and backup paths optimal design for traffic engineering in hybrid IGP/MPLS networks. In *7th International Workshop on Design of Reliable Communication Networks*, pages 273–280.

Musunuri, R. and Cobb, J. A. (2005). Comprehensive Solution for Anomaly-Free BGP. In Magedanz, T., Madeira, E. R. M., and Dini, P., editors, *Operations and Management in IP-Based Networks*, pages 130–141, Berlin, Heidelberg. Springer Berlin Heidelberg.

Obradovic, D. (2002). Real-time Model and Convergence Time of BGP. volume 2.

Oprescu, I., Meulle, M., Uhlig, S., Pelsser, C., Maennel, O., and Owezarski, P. (2011). oBGP: an Overlay for a Scalable iBGP Control Plane. In *NETWORKING 2011 - 10th International IFIP TC 6 Networking Conference, Spain, May 9-13, 2011*, Valencia, Spain.

Papadimitriou, C. and Yannakakis, M. (1991). Optimization, Approximation, and Complexity Classes. *Journal of Computer System Sciences*, pages 425–440.

Park, J. H. (2011). *Understanding the Impact of Internal BGP Route Reflection*. PhD thesis, University of California.

Pelsser, C., Uhlig, S., Takeda, T., Quoitin, B., and Shiomoto, K. (2010). Providing scalable NH-diverse iBGP route re-distribution to achieve sub-second switch-over time. *Comput. Netw.*, 54(14):2492–2505.

Raszuk, R., Cassar, C., Aman, E., Decraene, B., and Wang, K. (2019). BGP Optimal Route Reflection (BGP-ORR). Expires January, 2020.

Raszuk, R., Fernando, R., Patel, K., McPherson, D., and Kumaki, K. (2012). Distribution of Diverse BGP Paths. RFC6774.

Raza, M. H., Kansara, A. K., Nafarieh, A., and Robertson, W. (2014). Central Routing Algorithm: An Alternative Solution to Avoid Mesh Topology in iBGP. In *The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks*, pages 85–91. Procedia Computer Science 37.

Raza, M. H., Kansara, A. K., and Robertson, W. (2016). Effective IBGP Operation without a Full Mesh topology. *International Refereed Journal of Engineering and Science (IRJES)*, 5(8):16–23.

Rekhter, Y. and Li, T. (1995). A Border Gateway Protocol 4 (BGP-4). RFC 1771 (Draft Standard), Obsoleted by RFC 4271.

Risso, C. (2014). *Using GRASP and GA to design resilient and cost-effective IP/MPLS networks*. PhD thesis, UdelaR/INRIA.

Risso, C., Nesmachnow, S., and Robledo, F. (2018). Metaheuristic approaches for IP/MPLS network design. *International Transactions in Operational Research*, 25(2):599–625.

Rosen, E., Viswanathan, A., and Callon, R. (2001). Multiprotocol Label Switching Architecture. RFC3031 (Draft Standard).

Schuller, T., Aschenbruck, N., Chimani, M., Horneffer, M., and Schnitter, S. (2018). Traffic engineering using segment routing and considering requirements of a carrier IP network. *IEEE/ACM Trans. Netw.*, 26(4):1851–1864.

Solla, V., Jambrina, G., and Grampín, E. (2017). Route reflection topology planning in service provider networks. In *2017 IEEE URUCON, URUCON 2017*, volume 2017-December, pages 1–4.

Sun, Xiaomei; Li, Q., Xu, M., and Yang, Y. (2016). Achieving Stable iBGP with Only One Add-Path. In *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, pages 688–696. IEEE.

Van den Schrieck, V., Francois, P., and Bonaventure, O. (2010). Bgp add-paths: The scaling/performance tradeoffs. *Selected Areas in Communications, IEEE Journal on*, 28(8):1299 –1307.

Vissicchio, S., Cittadini, L., Vanbever, L., and Bonaventure, O. (2012). iBGP Deceptions: More Sessions, Fewer Routes. In *INFOCOM, 2012*. IEEE.

Vutukuru, M., Valiant, P., Kopparty, S., and Balakrishnan, H. (2006). How to Construct a Correct and Scalable iBGP Configuration. In *Proceedings of the 25th IEEE International Conference on Computer Communications.*, INFOCOM 2006, pages 1–12.

Walton, D., Retana, A., Chen, E., and Scudder, J. (2016). Advertisement of Multiple Paths in BGP. RFC7911.

Xiao, L., Wang, J., and Nahrstedt, K. (2003). Reliability-aware IBGP route reflection topology design. *Proceedings - 11th IEEE International Conference on Network Protocols, ICNP*, 2003-January:180–189.

Yang, D. and Rong, Z. (2015). Evolution of the Internet at the autonomous system level. In *2015 34th Chinese Control Conference (CCC)*, pages 1313–1317.

Zhang, R. and Bartell, M. (2003). *BGP Design and Implementation*. Cisco Press, Indianapolis.

Zhao, F., Lu, X., Zhu, P., and Zhao, J. (2006a). BGPSep_D: An Improved Algorithm for Constructing Correct and Scalable IBGP Configurations Based on Vertexes Degree. In *Proceedings of the Second International Conference on High Performance Computing and Communications*, HPCC'06, pages 406–415, Berlin, Heidelberg. Springer-Verlag.

Zhao, F., Lu, X., Zhu, P., and Zhao, J. (2006b). Bgpsep_S: An algorithm for constructing IBGP configurations with complete visibility. In *Distributed Computing and Networking, 8th International Conference, ICDCN 2006, Guwahati, India, December 27-30, 2006.*, pages 379–384.

Part I

Optimal Route Reflection Topology Design

Chapter 2

Optimal Route Reflection Topology Design

In the first place the objective is, for the nominal case, to optimize (minimize) the quantity of Route Reflectors (RRs) within the AS, such that no sub-optimal route is chosen. In other words, the routes selected with the designated RRs are the same that would have been selected if instead of having RRs, the iBGP speakers were fully meshed. Experimental results are shown when applied to known network topologies, achieving notorious improvement when compared to other existing heuristics approaches.

Optimal Route Reflection Topology Design

Cristina Mayr
Universidad de la República
Montevideo, Uruguay
mayr@fing.edu.uy

Eduardo Grampín
Universidad de la República
Montevideo, Uruguay
grampin@fing.edu.uy

Claudio Risso
Universidad de la República
Montevideo, Uruguay
crisso@fing.edu.uy

ABSTRACT

Autonomous Systems (ASes) exchange routing information about networks they can reach in the Internet, and the most widely extended way to connect them is by means of Border Gateway Protocol (BGP) sessions. ASes set up external BGP (eBGP) sessions between the AS border routers (ASBR) of neighboring ASes, and the routing information learned by ASBRs is then redistributed inside the AS through internal BGP (iBGP) sessions. In order to avoid loops, iBGP can not re-advertise prefixes learned from an iBGP neighbor to another iBGP neighbor. To have complete visibility, routers within the same AS are required to be connected in full-mesh. This causes scaling problems, since the number of required sessions grows quadratically with the number of routers involved. For large networks this can lead to administration issues, and therefore, in order to manage scalability, Route Reflection is generally accepted as an alternative to full-mesh. Even though Route Reflectors (RRs) simplify administration, they also introduce new problems such as: routing sub-optimality, increased probability of loops, poor route diversity, among others.

The objective of the present work is to optimize (minimize) the number of Route Reflectors (RRs) within the AS, such that no sub-optimal route is chosen. In other words, the routes selected with the designated RRs are the same that would have been selected if the iBGP speakers were fully meshed.

CCS CONCEPTS

• **Networks** → *Logical / virtual topologies*; • **Theory of computation** → *Mathematical optimization*;

KEYWORDS

Internet Routing, BGP, Route Reflection, Network Design, Combinatorial Optimization

ACM Reference Format:
Cristina Mayr, Eduardo Grampín, and Claudio Risso. 2018. Optimal Route Reflection Topology Design. In *Latin America Networking Conference (LANC '18), October 3–4, 2018, São Paulo, Brazil.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3277103.3277124

LANC '18, October 3–4, 2018, São Paulo, Brazil

ACM ISBN 978-1-4503-5922-1/18/10…$15.00
https://doi.org/10.1145/3277103.3277124

1 INTRODUCTION

The inter-domain routing is supported by the Border Gateway Protocol [22], which is used to exchange reachability information among Autonomous Systems (ASes). BGP is a path-vector, policy-routing protocol, with capabilities to express network operator commercial requirements and policies (attaching several attributes to network prefixes). Routing scalability has been a a matter of deep concern in the Internet community for the last ten years [14], and involves both *inter-domain* and *intra-domain* issues. Intra-domain routing is composed by the interaction of the Interior Gateway Protocol (IGP) and Internal BGP (iBGP); while the IGP builds connectivity for internal prefixes, iBGP is needed to determine the exit gateway for packets with destination to external ASes. The interaction of the IGP and iBGP for large, transit ASes, usually follows the *Pervasive BGP* model: all routers in an AS are iBGP speakers; a router running external BGP (eBGP) sessions is called Autonomous System Border Router (ASBR), while a router running only iBGP sessions is called Internal Router (IR).

The AS_PATH attribute, attached to every reachable prefix by BGP, plays two fundamental roles. On the one hand, disregarding administrative issues, is the most important decision metric, i.e., for a given prefix, the shortest AS_PATH is chosen. On the other hand, the AS_PATH attribute is used to avoid loops (i.e., if a router finds its own ASN in a BGP update, it must be discarded). In order to make sure that internal transport of BGP info is loop-free (control plane), and internal routing is coherent (loop-free data plane forwarding), the following iBGP advertisement rules must be observed: 1) prefixes learned from an eBGP neighbor can be re-advertised to an iBGP neighbor, and vice versa, and 2) prefixes learned from an iBGP neighbor cannot be re-advertised to another iBGP neighbor (the *split-horizon rule*). While the first rule makes sure that the complete routing information is disseminated, the second rule prevents BGP announcements from looping, since iBGP cannot rely on the AS_PATH attribute to detect loops, because this attribute remains unchanged intra-domain. The practical implication of this rule is that a full mesh of iBGP sessions between each pair of routers in the AS is required, resulting in $\frac{n\times(n-1)}{2}$ iBGP sessions for a domain of n routers. Furthermore, the routing state (i.e. the size of iBGP Rib-In routing table) can be n times larger than the number of best routes (i.e., for T best routes, Rib-In size can reach up to $n \times T$ entries), imposing large CPU and memory requirements to every router in the AS. Another aspect of iBGP scalability that need to be considered is the number of BGP messages generated intra-domain by external BGP updates.

An extensively used alternative to tackle down the scalability concerns is *route reflection* [2]. In this case, one or more routers within the AS are designated as Router Reflector (RRs), while the rest plays the RR client role (client for short). RRs are allowed to

infringe the split-horizon rule, and therefore they can re-advertise routes learned from an internal peer to other internal peers. With route reflection the number of iBGP sessions can be as low as $n-1$ when using a sole RR, where n is the number of iBGP routers. Since a unique RR in an AS constitutes a single point of failure, at least two routers are selected as RRs. There is a body of work regarding the election of the RRs, which can be defined as the RR topology, or *iBGP overlay design problem*. There exist many algorithms and heuristics to solve this problem, including hierarchical proposals. We review some of such proposals throughout this paper; the interested reader may further refer to [7, 17], among many other authoritative studies.

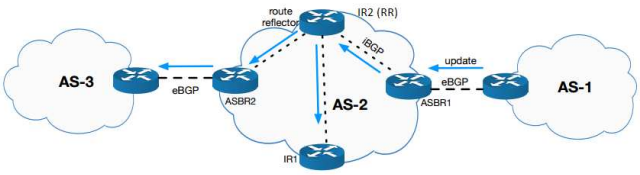
This work proposes a novel combinatorial optimization approach to undertake the problem of designing a consistent and yet optimal iBGP overlay for an AS. The optimal criteria is using the minimum number of route reflectors and sessions, maintaining *correctness* and *full-mesh optimality* [4, 9, 24], assuming that all prefixes matching a common gateway or a set of equally preferred gateways are clustered into classes of prefixes (or labels).

The document is organized as follows: Section 2 reviews issues when using route reflection and some existing alternatives to reflection; Section 3 explains what is an iBGP overlay and proposes a mathematical approach for route reflectors selection; Section 4 presents experimental results over some network topologies; and finally, Section 5 summarizes our main conclusions and lines for future work.

2 KNOWN ISSUES FOR REFLECTION

Although Route Reflectors improve BPG scalability, there are several issues to consider at the time of designing an iBGP overlay that includes RRs. When more than one update for the same destination (prefix) is received by a RR, it only reflects routes considered as best routes based on its local routing information. Although efficient in terms of hardware processing, when mis-configured, this could be a major drawback, resulting into sub-optimal routing, because clients will have less routing options (path diversity). For example, in Figure 1, if $ASBR_2$ is closer than $ASBR_1$ to IR_2 (the router reflector), then routes having $ASBR_1$ as next hop will never reach IR_1. Then a sub-optimal route is installed in its routing table.

Figure 1: Route Reflection - several reflectors



The following is a list of some of the typical problems that might arise from the usage of route reflection:

(1) **Less robustness -** If a RR fails, its client routers also become disconnected, which affects network availability and routing stability.
(2) **Slow convergence -** An update message may take several hops before reaching the iBGP destination router.
(3) **Sub-optimal routes -** As each RR has a partial view of the network topology, and only propagates its better option, it might select a best path different from the choice in full mesh.
(4) **Increased probability of loops -** In general there is more than one RR, to avoid single points of failure. So clients connect to several RRs, which could potentially introduce data plane loops.
(5) **Non deterministic behavior -** This happens when routing decisions depend on the arrival order of announcements.

2.1 Dealing with route reflection problems

Pelsser et al. [6] propose to implement an iBGP route distribution architecture relying on Route Servers (RS), each one responsible for a subset of the external destinations. A similar strategy is presented in [20] and [21] where the root node in an AS is responsible for all the control and management operations such as maintaining routing tables and calculating paths. In [5], Address-Based Route Reflection (ABRR) concept is presented. With this approach each RR is responsible for a portion of prefixes from all routers, and there is no constraint on RR placement. There is also another approach, described in [15] which proposes to use an *overlay* of routing instances (oBGP) responsible for performing the BGP decision process on behalf of the client routers within the AS. A more recent work by Buob et al. [3] proposes iBGP2, which completely avoid route reflection, building a dissemination spanning tree for every prefix, rooted in the ASBRs. Therefore, each BGP speaker in the AS has to compute shortest paths from each of its iBGP neighbors towards candidate egress points, where a router u only advertises a route towards an egress point s to a neighbor v if and only if u belongs to the shortest path from v to s. This is a variant of the idea of disseminating updates using multicast [18], implemented using Software Defined Networking ideas in [8].

Complementary, practical alternatives or variations to classical Route Reflectors architecture have also been proposed to improve BGP reliability, including the following, among others:

Multi-path - RFC7911 [26] defines a BGP extension that allows the advertisement of multiple paths for the same address prefix without the new paths implicitly replacing any previous ones.

BGP Prefix-Independent Convergence (PIC) - Given the large scaling targets, it is desirable to restore traffic after a failure within a time period that does not depend on the number of BGP prefixes. BGP PIC is based on a shared hierarchical forwarding chain (taking into account that multiple destinations are reachable via the same list of next-hops), and a forwarding plane that supports multiple levels of indirection. This is a recent proposal described in [1].

BGP Advertise-Best-External provides the network with a backup external route to avoid loss of connectivity with the primary external route. It is useful when an ASBR chooses a path received over an iBGP session (of another border router) as the best path for a prefix even if it has a path learned from eBGP. In this case the router can advertise one externally learned path called the *best external path*. There is a draft

issued by IETF [13] but no updates ever since. However, main router providers do implement this feature.

Add-Path - Described in RFC7911 [26]. It defines a BGP extension that allows the advertisement of multiple paths for the same address prefix without the new paths implicitly replacing previous ones. The objective is to reduce persistent route oscillations and improve routing convergence.

Diverse-Path (Shadow Router) - BGP can distribute an alternative path other than the best path between BGP speakers: for each RR in the AS, a *shadow route reflector* is added to distribute the second best path, or *diverse path*. It is described in the informational RFC6774 [19].

3 IBGP OVERLAY DESIGN BASED ON THE REFLECTION OF ROUTES

Along Section 2 we reviewed known problems derived from route reflection, as well as several strategies for either: introduce changes in the standards to mitigate those issues, or directly to propose alternate architectures to deal with the lack of scalability of iBGP. Most of the state of the art regarding the reflection problem undertakes the path of updating or changing technology. On the other hand, the work presented in this document and other related works, rely on a careful design of the iBGP overlay assembled up from existing and widely used standards. Prefixes information is exchanged between peers through *BGP updates*.

Route selection process in BGP consists in applying a sequence of rules in hierarchical order to determine the best path, when alternatives exist for a given prefix. Those rules inspect the characteristics and attributes of each BGP route (which consist of a prefix plus a number of attributes), in order the find the *best route* for each prefix.

After processing the highest priority attributes (either administrative or structural) for each prefix, if there is no winner yet, the path selection process takes in to account the IGP information, and therefore BGP route selection depends on interactions with intra-domain routing protocols. Whereas the IGP can be modeled as stand alone, the selection of the best BGP route at each router also depends on the IGP path cost to the BGP next hop announcing the route. *Hot potato* routing, where a router prefers the route with the shortest IGP path (the closest exit point), introduces a tight relationship between BGP and the underlying IGP.

As we mentioned before, when deployed in a full mesh scheme the iBGP selection process counts with the whole pre-filtered set of entries; in parallel it knows the details of the internal topology from the IGP (of which is a member), so it can determine the optimal route for every prefix. On the other hand, iBGP route reflection provides network operators with good scalability at the cost of possibly introducing routing and forwarding anomalies when misconfigured, as those enumerated in Section 2.

There are some previous research works about RR selection, focusing mainly in reliability. For example, in [27] the authors address the problem of finding reliable route reflection (RR) topologies for iBGP networks, by means of the concepts of iBGP expected lifetime and expected session loss. They propose to consider three major criteria in designing a reliable iBGP route reflection topology:

- the number of clusters needed[1]
- how to choose the route reflectors (RR)
- how to assign clients to route reflectors (RR)

Park [16] focuses on the impact of route reflection and presents an evaluation and analysis results on its impact on two important metrics of BGP performance: BGP path diversity and convergence delay inside an ISP. He also studied *hierarchical route reflection (HRR)*, a common technique that implements RR in several layers and found that although HRR brings an increase in the routing update counts, this overhead is not significant in most cases, and can be mitigated through a carefully engineered iBGP topology. In his work it is described how some ISP configured a pair of RRs in each of its major POPs (point of presence, an access point from one place to the rest of the Internet), so that client routers connect to the RRs residing in the same POP, making the logical iBGP topology follow the underlying geographic locations. But there is no proposal on how those RRs should be selected. Zhang et al. [28] propose a hierarchical design with different route reflectors levels to reduce the number of sessions: route reflectors that are clients of higher level reflectors can reflect routes for routers of lower ones. The upper level reflectors must be fully meshed. The recommendation is that if the number of full mesh sessions in the top level iBGP mesh is administratively unmanageable, one should consider introducing RR hierarchy. In [24], Vissicchio et al. show that iBGP route propagation can trigger unexpected side effects like forwarding loops. They define *dissemination correctness* to model visibility issues caused by iBGP route propagation rules. They also show that the addition of just a single iBGP session can affect the ability of iBGP to correctly distribute routing information within an AS. This contradicts prior researchers who had proposed to optimize iBGP routing by adding extra sessions. Moreover, they also prove that deciding whether an iBGP configuration is *dissemination correct* is computationally intractable, and the problem of determining whether the addition of a single iBGP session can adversely affect dissemination correctness of an iBGP configuration is also computationally intractable.

Conversely to the previously referred works, which mainly focus upon reliability issues associated to Route Reflection, this work aims on those problems derived from the lack of optimality. That problem is far from being easy, and in fact some versions of it can be proven NP-Complete.

3.1 Objects and premises of the technique

The object of this work is to design an iBGP overlay with minimum number of route reflectors and sessions, and yet *route optimal* for a steady configuration of eBGP messages upon an internal given topology. We consider only one cluster, and propose a methodology to choose the route reflectors (RR), and how to assign clients to route reflectors (RR). A fundamental premise along this work is a fixed set of eBGP adjacencies, i.e., a known set of ASBRs, whose internal topology particulars are known to all other routes within the AS up from the IGP. Besides, we assume that either from a real or forecasted state of adjacencies, the sets of prefixes (attributes included) learned by each ASBR are known in advance. Finally, we assume that a configuration of BGP rules has been set to fulfill some strategy about the way traffic is exchanged with other ASes, so

[1] A cluster is composed by a RR and its clients, and permit to scale RR deployment.

after filtering those rules under the paradigm of a full meshed iBGP overlay, we can deduce the optimal route for each prefix. At this level of detail and without loss of generality, we can think of those routes as grouped into classes, whose prefixes are indifferent at the time of choosing one ASBR over another, to all purposes except for the BGP optimality.

The technique introduced in this work is called *Optimal Route Reflector Topology Design* (ORRTD). It does not regard with resilience issues, but aims instead on being optimal in terms of routing, and in the number of reflectors and sessions to keep. As we see later on, the technique relies upon an integer programming problem formulation, whose constraints have been chosen to always select IGP optimal routes. Hence, among the issues enumerated in Section 2, a feasible solution cannot fall down onto those numbered as: (2) slow convergence, (3) sub-optimal routes and (4) increased probability of loops. The certainty of not falling in problem (3) situations comes from the fact that the iBGP routes are optimal by construction. Remains to be seen that they effectively apply (hop-by-hop) inside the AS in conjunction with the hot-potato paradigm, which will be seen during the analysis of problem number (4).

Besides and also by construction, the topology of adjacencies for each class of prefixes is a clique-star, that is, a full mesh among route reflectors combined with ASBR-to-RR sessions or CLI_IR-to-RR (i.e. Internal not Route Reflector to Internal Route Reflector). Thus, the diameter of the iBGP sub-overlay for each class is 3 at most, and messages are to be rapidly propagated through the overlay (do not incur in problem type number (2)).

Finally, problems of type (4) -loops- cannot happen either. As a basis, there is a hypothesis regarding the correct setting of the split-horizon rule, which is deactivated only among RRs. The route optimality argument guarantees that if any, a routing loop cannot be generated at the BGP level. Hence, the existence of loops is bound up with an inconsistency between routes at the BGP and the IGP levels. However, such inconsistencies cannot exist because of the *optimality principle*, an intrinsic characteristic of the *shortest-path problem*, upon which link-state IGPs are based. Observe that prior to the lowest IGP metric step in BGP's path selection, there exists a prioritization of eBGP over iBGP paths. Therefore, among all those prefixes matching a list of attributes, it is not possible to get to a point where an ASBR relies on the IGP metric to learn an external route that has also been learned from a remote peer. In other words, there cannot be loops between border routers. There cannot be loops at the level of internal addresses either, because of the IGP consistency, i.e., because of the optimality of its cost.

So, to complete the argument we must prove that loops cannot happen between internal and external addresses. Actually, we will prove that packets must follow an optimal path towards its border gateway, which closures the route optimality (problem number 3), a result beyond the absence of loops. Suppose that an internal router IR_1 (reflector or not) learns that its optimal hop for a given class of prefixes *ClassA* is $ASBR_1$. Later, a packet is forwarded from IR_1 with destination to *ClassA*, for which is sent to IR_2 as the next-hop towards $ASBR_1$. If IR_2 has selected a border router other than $ASBR_1$ (let $ASBR_2$ be that router), the path the packets follow could be different from the originally intended by IR_1. The selection however, must also be optimal (i.e. with equal cost), otherwise IR_1 would not have identified the border properly. Indeed, since the *optimality principle* applies to all internal routers (by construction), if IR_2 is an intermediate node in the optimal path from IR_1 to a target *ClassA* (learned from $ASBR_1$), and the optimal target from IR_2 to the same destination is through $ASBR_2$, the costs from IR_2 to $ASBR_1$ and to $ASBR_2$ must match. Otherwise, the path $IR_1 - IR_2 - ASR_2$ should be optimal rather than $IR_1 - IR_2 - ASR_1$, or $IR_2 - ASR_1$ should be the sole optimal for IR_2. As a corollary, since IGP costs are positive and internal-to-external traffic is cost optimal, loops can not happen.

3.2 From the raw problem to its Integer Programing Formulation

We consider a scenario with the following characteristics: i) there is an Autonomous System with a collection of BGP speaking routers connected by a pure IP network (i.e. hop-by-hop routing); ii) there is also a set of Autonomous System Border Routers (ASBRs) that send and receive routing information to/from other ASes, which have come to a steady state of prefixes database; and iii) we want to determine which of the internal routers will be designated as Route Reflectors (RR), with certain constraints:

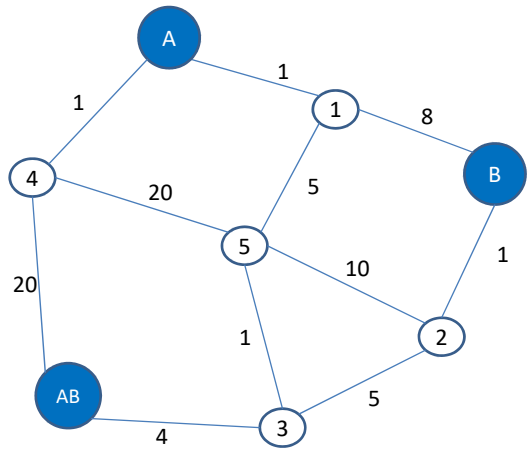
(1) The routers are either internal routers (IR) or ASBRs;
(2) Only IRs can be RRs (an ASBR cannot be a RR);
(3) Every Client-IR (i.e. not RR) must be a peer of a unique RR per class of prefixes;
(4) Whenever optimal for some IR, every ASBR must be connected to at least one RR per class of prefixes;
(5) An ASBR cannot peer with a Client-IR: it can only peer with RRs.

We also assume that: all external prefixes are learned through BGP; that they have been filtered according to the path selection algorithm to get to a set of classes of prefixes; ant that the complete AS topology (costs included) is known. BGP updates from ASBRs will be with *next-hop self*, as internal network does not know about external routes. In the present work no additional functionality is needed, no additional BGP sessions, and no changes to BGP process are suggested. Finally, we remark that resilience is not considered, so correctness and optimality of solutions are only guaranteed for a non-faulty state. Nevertheless, that does not mean that solutions are not resilient, but that the *full mesh optimality* might be lost in some failure scenarios.

The process to go from an instance so defined to an Integer Programing Problem is as follows. Suppose we have the graph associated with the network of some AS, like that represented in graph of Figure 2. Vertices correspond to routers while edges do to links. The graph is undirected and weighted, being the IGP cost the weight of each link. Different classes of prefixes come from each ASBR; the other routers are internal. For instance, Figure 2 shows an example AS with eight routers, three of which are ASBRs (A, B and AB). In the example we suppose we have two prefix classes, labeled as *A* and *B*. ASBRs A and AB are potential gateways for prefix class *A*, and ASBR B and AB are potential gateways for prefix class *B*.

Up from that information, an optimal internal-to-border distance matrix can be built. Following with the example of Figure 2 and after running Dijkstra's shortest path algorithm from each ASBR to the set of internal nodes, we find for each class of prefixes the

Figure 2: Graph with 2 prefix classes (A and B)



optimal IGP distance towards each ASBR. Results are shown in Table 1.

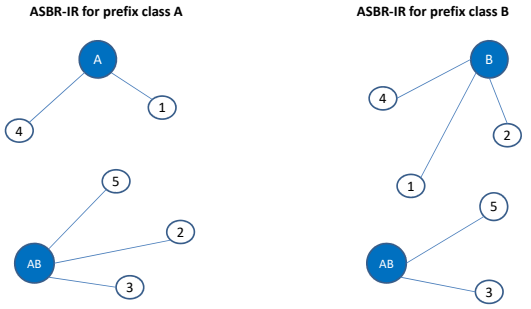
Table 1: Per-class optimal distance to ASBRs

IR to Class A		
	A	**AB**
IR_1	1	10
IR_2	10	9
IR_3	7	4
IR_4	1	12
IR_5	6	5

IR to Class B		
	B	**AB**
IR_1	8	10
IR_2	1	9
IR_3	6	4
IR_4	10	12
IR_5	7	5

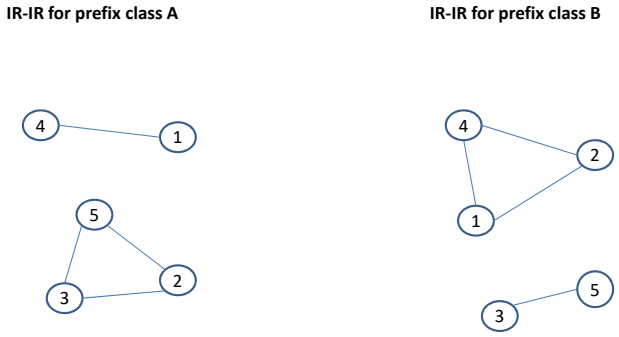
After applying the optimal IGP distance filter to class A prefixes, the BGP's path selection algorithm should conclude that: ASBR router A is the next-hop for IR_1 and IR_4, while AB should be for IR_2, IR_3 and IR_5. For prefixes class B on the other hand, the optimal IGP metric should result in: B being the next-hop for IR_1, IR_2 and IR_4, while AB is the preferred option for IR_3 and IR_5. Such internal-to-border affinities define a graph for each class of prefixes, which are respectively sketched on the left and rightmost of Figure 3.

Figure 3: IR-ASBR Adjacency graphs - 2 prefix classes (A and B)



Complementary, internal routers that share a common ASBR for a common class of prefixes, could serve as the reflector of each other for that class. We represent such relation by a second graph of affinities, in this case among internal routers (see Figure 4). The

Figure 4: Internal to Internal IR affinities graphs for prefixes classes A and B



previous concepts set the grounds to formalize our problem.

Problem Statenment

Formally, a problem instance is defined by the weighted IGP graph $G = (V, E, W)$, for a partition of the nodes $V = ASBR \cup IR$ ($ASBR \cap IR = \emptyset$), $E \subseteq V \times V$, $W : E \rightarrow \mathbb{N}^+$ and a pre-filtered set of prefixes represented by finite set of classes C with a function $ASBR2CLASS : ASBR \rightarrow 2^C$, where 2^C is the set of all the subsets of C. This represents which prefix classes are received by each ASBR.

Up from this data, the problem is transformed into two unweighted undirected families of graphs:

- $ASBR2IR4C : C \times ASBR \rightarrow 2^{IR}$. This function represents, for each prefix class, the association of IRs to preferred ASBRs, as shown intuitively in Figure 3.
- $IR2IR4C : C \times IR \rightarrow 2^{IR}$. This means: for each prefix class, which IRs share a common ASBR? In the previous example, for prefix class A, IR_1 and IR_4 have a common preferred ASBR (A), while IR_2, IR_3 and IR_5 have as preferred ASBR (B), as shown in the left side table of Figure 4.

The computation of both function relies on the shortest path algorithm so they are of polynomial time complexity. From now on, the original graph is no longer needed.

The next step consists in assembling all these pieces into a single combinatorial optimization problem. The formulation is as follows:

$$
\begin{cases}
\min \sum_{i \in IR} x_i & \\
\text{Subject to :} & \\
\sum_{(ij) \in S^k} y_{ij}^k \geq 1, & \forall i \in BR, k \in C, \quad S^k \neq \emptyset \quad (i) \\
x_j - y_{ij}^k \geq 0, & \forall i \in BR, k \in C, \quad (ij) \in S^k \quad (ii) \\
x_j + \sum_{(ij) \in T^k} z_{ij}^k \geq 1, & \forall j \in IR, k \in C \quad (iii) \\
x_i + x_j - z_{ij}^k \geq 0, & \forall i \in IR, k \in C \quad (ij) \in T^k \quad (iv) \\
x_i + x_j + z_{ij}^k \leq 2, & \forall i \in IR, k \in C \quad (ij) \in T^k \quad (v) \\
\sum_{(jh) \in S^k} y_{jh}^k - z_{ih}^k \geq 0, & \forall j \in IR, k \in C \quad (ih) \in T^k \quad (vi) \\
x_i, y_{ij}^k, z_{ij}^k \in \{0,1\}, & \forall i, j \in V, k \in C
\end{cases}
\tag{1}
$$

Equation 1 has the following **parameters**:

- BR : set of all Autonomous System Border Routers
- IR : set of all Internal Routers
- C : set of prefixes classes
- $\{S^k\}$: set of border-to-internal BGP affinity matrices, i.e., $S_{ij}^k = 1$ if and only if $j \in ASBR2IR4C(k, i)$, with $k \in C$, $i \in$ BR, $j \in$ IR
- $\{T^k\}$: set of internal-to-internal BGP affinity matrices, i.e., $T_{ij}^k = 1$ if and only if $j \in IR2IR4C(k, i)$, with $k \in C$, $i \in$ IR, $j \in$ IR

and the following boolean **variables**:

- x_i : 1 if router i is to be a RR and 0 otherwise;
- y_{ij}^k : 1 if ASBR i is to be iBGP adjacent to IR j for prefixes class k and 0 otherwise;
- z_{ij}^k : 1 if IR i is to be iBGP adjacent to IR j for prefix class k and 0 otherwise.

The objective function in Equation 1 pushes down to get the minimum number of route reflectors. Constraints in Equation 1 deserve a more detailed analysis. Equations in group (i) force every ASBR full-mesh optimal for some IR and prefixes class, to be adjacent to at least one IR. Equations in group (ii) impose for those internal routers iBGP adjacent to an ASBR to be reflectors. That is, if internal router j is adjacent to an ASBR i, then j must be an RR.

It is worth mentioning that Route Reflectors are globally selected, that is, they are common to all prefixes classes.

Equations in group (iii) guarantee that each internal router is a route reflector, or - for every class of prefixes- is iBGP adjacent to another internal router. We remark that for this model, the only adjacencies considerer are those in $S = \cup_k S^k$ and $T = \cup_k T^k$. Moreover, adjacencies between route reflectors are implicit, so they are ignored during the optimization.

That is the reason why equations (iv) and (v) combined force to 1 the number of reflectors in an internal-to-internal adjacency. Indeed, if $z_{ij}^k = 1$ (IR i is iBGP adjacent to IR j for prefix class k) for some $k \in C$ then $x_i + x_j = 1$ must hold, so either $x_i = 1$ or $x_j = 1$ but not both (either internal router i or internal router j is a RR). Observe that in those cases where variables z_{ij}^k indicate multiple sessions between a pair of internal routers (i.e. for different classes), they must be replaced by a single iBGP adjacency. The path selection algorithm is in charge of taking the optimal one among all updates.

Furthermore, since the objective of the optimization only accounts the number of RRs, numerical solutions to Equation 1 could include unnecessary ASBR-to-RR or CLI_IR-to-RR adjacencies. They, however, can be easily (polynomially) post-filtered.

Finally, to be consistent in the IGP optimality of the next hop, equations (vi) force every internal router i to get its optimal gateway towards prefixes class k, from a reflector h that is connected in turn to some border router j, optimal for that class from the perspective of i. That is because if z_{ih}^k=1 then there exists $h \in IR$ such that y_{jh}^k=1.

For small networks, solutions can be easily found by brute-force or quasi-exhaustive methods. For more complex networks the problem can be solved with any popular solver like GLPK or CPLEX, even for hundreds of nodes, while the number of prefixes classes is limited. For even more complex problems, with hundreds of classes, a heuristic approach should be used.

4 EXPERIMENTAL EVALUATION

In this section we present early results obtained in the emulation environment proposed by [23] which is based on Quagga[2], MiniNExT[3] (an extension layer to build complex networks in Mininet [12]) and ExaBGP[4] for injecting BGP messages. The benefits of this environment are that it supports IPv6 and 32 bits Autonomous System Number (ASN4), it also supports routing protocols such as OSPF and BGP, and it is possible to inject BGP traces to test the dissemination of the routing information inside the AS under test. The topologies were taken from "The Internet Topology Zoo" repository [11] and slightly adapted. We compare the output of the solution obtained with ORRTD, with that obtained with full-mesh. We also solve the RR location using other algorithms implemented by the RRLOC tool [10].

Table 2: Comparison of algorithms

Topology	# IRs	# ASBRs	RRs ORRTD	RRs BGPSep	BGPSepS	RRs Zhang
Abilene	8	3	2	5	4	4
AB5	5	3	3	5	4	4
AB10	10	3	2	10	5	4
Airtel	3	6	1	4	3	4
Garr	47	7	4	12	20	4
UniC	24	3	2	8	10	4
Uninett	67	3	2	27	26	4

[2]Quagga Routing Suite. Available at: https://www.quagga.net/. Accessed: 2018-06-01
[3]MiniNExT (Mininet ExTended). Available at: https://www.quagga.net/. Accessed: 2018-06-01
[4]https://github.com/Exa-Networks/exabgp

We first show a comparison resulting of several known algorithms that locate RRs: BGPSep [25] , an algorithm based on the notion of a graph separator and that claims to ensure loop-free forwarding and complete visibility properties, another version of BGPSep called BGPSep_S [29] for which the authors claim that produces a smaller number of iBGP sessions, and Zhang [28], which focus on hierarchical route reflection. In all cases our proposed algorithm (ORRTD) results in a reduced number of RRs. This can be seen in Table 2. The BGP decision taken by ORRTD algorithm applied to the example shown in Figure 2 is presented in Table 3. We took two prefixes representing each one different prefix classes, and emulated BGP behavior both with a ORRTD topology (left side table) and full mesh (right side table). Both tables correspond to the resulting LOC_RIB table, indicating, for each router, the chosen next_hop. For example, for router 1 and prefix class represented by 177.10.158/24, in both cases next_hop is 192.168.0.1, which correspond to the ospf identifier assigned by the emulator to one of the border routers. It can be seen that the next_hop is the same for FM and ORRTD, i.e. the proposed algorithm is taking the same decision as if it network had been configured in full mesh (preserving full mesh optimality).

Table 3: Comparison of next hop

ORRTD Model loc_rib_ipv4 table			
prefix	next_hop	router	id
177.10.158.0/24	192.168.0.1	B	1
202.70.88.0/21	172.16.2.2	B	2
177.10.158.0/24	192.168.0.1	4	4
202.70.88.0/21	192.168.0.7	4	5
177.10.158.0/24	192.168.0.8	2	7
202.70.88.0/21	192.168.0.7	2	8
177.10.158.0/24	192.168.0.8	5	10
202.70.88.0/21	192.168.0.8	5	11
177.10.158.0/24	172.16.1.2	A	13
202.70.88.0/21	192.168.0.7	A	14
177.10.158.0/24	172.16.3.2	AB	16
202.70.88.0/21	172.16.3.2	AB	17
177.10.158.0/24	192.168.0.1	1	19
202.70.88.0/21	192.168.0.7	1	20
177.10.158.0/24	192.168.0.8	3	22
202.70.88.0/21	192.168.0.8	3	23

FM model loc_rib_ipv4 table			
table	next_hop	router	id
177.10.158.0/24	192.168.0.1	B	37
202.70.88.0/21	172.16.2.2	B	38
177.10.158.0/24	192.168.0.1	4	40
202.70.88.0/21	192.168.0.7	4	41
177.10.158.0/24	192.168.0.8	2	43
202.70.88.0/21	192.168.0.7	2	44
177.10.158.0/24	192.168.0.8	5	46
202.70.88.0/21	192.168.0.8	5	47
177.10.158.0/24	172.16.1.2	A	49
202.70.88.0/21	192.168.0.7	A	50
177.10.158.0/24	172.16.3.2	AB	52
202.70.88.0/21	172.16.3.2	AB	53
177.10.158.0/24	192.168.0.1	1	55
202.70.88.0/21	192.168.0.7	1	56
177.10.158.0/24	192.168.0.8	3	58
202.70.88.0/21	192.168.0.8	3	59

Several prefix classes were also considered. Recall that in this paper we define a prefix class as a group of prefixes, and the decision of how to form these groups is upon the ISP. The results for 2 prefix classes are shown in table Table 4. In the studied case there are ASBRs that receive only prefixes of class A, ASBRs that receive only prefixes of class B, and ASBRs that receive prefixes from both classes. In our example, border routers A and AB receive prefixes of class A, while B and AB receive prefixes of class B. For the purpose of the test, we take a single prefix to represent a prefix class, as the behavior would be the same. Choosing the right prefix classes is not a trivial job, but for the purpose of this article we assume that prefix classes categorization is a given input for the optimization process, and is done based on ISP policies.

5 CONCLUSION AND FUTURE WORK

BGP is an essential protocol supporting Internet connectivity, and BGP scalability is one of the most prominent issues of this protocol, particularly in the intra-domain scope, as a full mesh among the routers is required. Most efforts to deal with the problem consist in

Table 4: RRs for two prefix classes

			# routers receiving prefix class	
Topology	**RRs**	**IRs**	**A**	**B**
Abilene	2	8	2	2
AB5	3	5	2	2
AB10	2	10	2	2
Airtel	2	3	4	4
Garr	4	47	4	4
UniC	2	24	2	2
Uninett	2	67	2	2

expanding the stack of technologies (i.e., proposing IETF RFCs to augment the protocol capabilities and improve its behavior). Among them, route reflection is a classic and simple approach, widely standardized over the Internet infrastructure. However, when not used properly, reflection could lead to other kinds of issues. Based on the concept of overlay networks, this paper proposes a novel mathematical approach to tackle down several known problems of refection, by means of a design that optimizes the scalability. The technique has been called Optimal Route Reflector Topology Design, or ORRTD for short. Among other advantages, with ORRTD there is no need to modify or augment existing BGP standards. Early experimental results corroborate not only the theoretical consistency of the ORRTD technique, but its outperformance over other alike heuristic approaches.

As a drawback, the current version of ORRTD is not fully resilient. The technique is quite flexible, so many resiliency improvements could be easily introduced. For instance, by adding $\sum_{i \in IR} x_i \geq 2$ to Equation 1 we can force the existence of two route reflectors per client. We can also force each ASBR to be connected to two reflectors by increasing to 2 the right-hand side of Equation 1-(i), or simply by adding sessions in a post-processing stage. Those changes to the model are insufficient nonetheless. IGP routing optimality is the cornerstone quality assurance of this technique, and that is only guaranteed for a fully operational network. Either by dropping nodes or links, it is possible to find counterexamples where the lack of IGP optimality leads (on faulty states) to misbehaviors like those we pursued to avoid for the normal operation (described in Section 2). An improved version of this model, fully resilient against single node or link failures, is one of our current lines of future work.

Until now we have relied on standard solvers, but this combinatorial formulation of the problem is numerically hard. Therefore, the strategy of using exact methods to solve ORRTD will eventually fail for larger instances. Another line of work goes by characterizing the intrinsic complexity of the problem. If proven NP-Hard we should implement heuristic approaches to tackle down larger instances of the problem.

It is also worth to mention that we assume that prefix classes categorization is a given input for the optimization process, and is done based on ISP policies, either static or dynamically. This classification may constitute a whole line of research, for example, using machine learning or other techniques to build the prefixes classes based on the dynamics of BGP updates.

Finally, we remark that this promising technique is being used in a real-world application, by means of a joint project between our University and ANTEL (national telecommunications operator of Uruguay), with the aim of designing portions of the infrastructure that supports Internet services of the company.

REFERENCES

[1] A. Bashandy, Ed Filsfils, and P. Mohapatra. 2018. BGP Prefix Independent Convergence. Expires: October, 2018. https://datatracker.ietf.org/doc/draft-bashandy-rtgwg-segment-routing-uloop/

[2] T. Bates, E. Chen, and R. Chandra. 2006. BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP). RFC4456.

[3] Marc Olivier Buob, Anthony Lambert, and Steve Uhlig. 2016. IBGP2: A scalable iBGP redistribution mechanism leading to optimal routing. In *35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016*, Vol. 2016-July. IEEE, 1–9. https://doi.org/10.1109/INFOCOM.2016.7524409

[4] Marc-Olivier Buob, Steve Uhlig, and Mickael Meulle. 2008. Designing optimal iBGP route-reflection topologies. In *Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet (NETWORKING'08)*. Springer-Verlag, Berlin, Heidelberg, 542–553. http://dl.acm.org/citation.cfm?id=1792514.1792575

[5] Ruichuan Chen, Aman Shaikh, Jia Wang, and Paul Francis. 2011. Address-based Route Reflection. In *Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies (CoNEXT '11)*. ACM, New York, NY, USA, Article 5, 5:1–5:12 pages. https://doi.org/10.1145/2079296.2079301

[6] Cristel Pelsser, Akeo Masuda, and Kohei Shiomoto. 2010. A novel internal BGP route distribution architecture. In *IEICE General Conference*.

[7] Ashley Flavel. 2009. *BGP, Not As Easy As 1-2-3*. Ph.D. Dissertation. University of Adelaide, Australia. Available at https://digital.library.adelaide.edu.au/dspace/bitstream/2440/60002/8/02whole.pdf.

[8] F. Godán, S. Colman, and E. Grampín. 2016. Multicast BGP with SDN control plane. In *2016 7th International Conference on the Network of the Future (NOF)*. 1–5. https://doi.org/10.1109/NOF.2016.7810140

[9] Timothy G. Griffin and Gordon Wilfong. 2002. On the Correctness of IBGP Configuration. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '02)*. ACM, New York, NY, USA, 17–29. https://doi.org/10.1145/633025.633028

[10] E. Gutiérrez, D. Agriel, E. Saenz, and E. Grampín. 2014. RRLOC: A tool for iBGP Route Reflector topology planning and experimentation. In *2014 IEEE Network Operations and Management Symposium (NOMS)*. 1–4. https://doi.org/10.1109/NOMS.2014.6838346

[11] S. Knight, H.X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. 2011. The Internet Topology Zoo. *Selected Areas in Communications, IEEE Journal on* 29, 9 (october 2011), 1765 –1775. https://doi.org/10.1109/JSAC.2011.111002

[12] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX)*. ACM, New York, NY, USA, Article 19, 6 pages. https://doi.org/10.1145/1868447.1868466

[13] P. Marques, R. Fernando, E. Chen, P. Mohapatra, and H. Gredler. 2017. Advertisement of the best external route in BGP draft-ietf-idr-best-external-05.

[14] D. Meyer (Ed.), L. Zhang (Ed.), and K. Fall (Ed.). 2007. Report from the IAB Workshop on Routing and Addressing. RFC 4984 (Informational). , 39 pages. https://doi.org/10.17487/RFC4984

[15] Iuniana Oprescu, Mickael Meulle, Steve Uhlig, Cristel Pelsser, Olaf Maennel, and Philippe Owezarski. 2011. oBGP: an Overlay for a Scalable iBGP Control Plane. In *NETWORKING 2011 - 10th International IFIP TC 6 Networking Conference, Spain, May 9-13, 2011*. Valencia, Spain.

[16] Jong Han Park. 2011. *Understanding the Impact of Internal BGP Route Reflection*. Ph.D. Dissertation. University of California.

[17] Cristel Pelsser, Steve Uhlig, Tomonori Takeda, Bruno Quoitin, and Kohei Shiomoto. 2010. Providing scalable NH-diverse iBGP route re-distribution to achieve sub-second switch-over time. *Comput. Netw.* 54, 14 (Oct. 2010), 2492–2505. https://doi.org/10.1016/j.comnet.2010.04.007

[18] Kedar Poduri, Cengiz Alaettinoglu, and Van Jacobson. 2003. BST - BGP Scalable Transport. http://www.nanog.org/meetings/nanog27/presentations/van.pdf

[19] R. Raszuk, R. Fernando, Keyur Patel, Danny McPherson, and Kenji Kumaki. 2012. Distribution of Diverse BGP Paths. RFC6774.

[20] Muhammad H. Raza, Ankit K. Kansara, Aliraza Nafarieh, and William Robertson. 2014. Central Routing Algorithm: An Alternative Solution to Avoid Mesh Topology in iBGP. In *The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks*. Procedia Computer Science 37, 85–91. https://doi.org/10.1016/j.procs.2014.08.016

[21] Muhammad H. Raza, Ankit K. Kansara, and William Robertson. 2016. Effective IBGP Operation without a Full Mesh topology. *International Refereed Journal of Engineering and Science (IRJES)* 5, 8 (2016), 16–23.

[22] Y. Rekhter and T. Li. 1995. A Border Gateway Protocol 4 (BGP-4). RFC 1771 (Draft Standard). , 57 pages. https://doi.org/10.17487/RFC1771 Obsoleted by RFC 4271.

[23] V. Solla, G. Jambrina, and E. Grampín. 2017. Route reflection topology planning in service provider networks.. In *2017 IEEE URUCON, URUCON 2017* (2017-december ed.), Vol. 2017-December. 1–4.

[24] Stefano Vissicchio, Luca Cittadini, Laurent Vanbever, and Olivier Bonaventure. 2012. iBGP Deceptions: More Sessions, Fewer Routes. In *INFOCOM, 2012*. IEEE.

[25] M. Vutukuru, P. Valiant, S. Kopparty, and H. Balakrishnan. 2006. How to Construct a Correct and Scalable iBGP Configuration. In *Proceedings of the 25th IEEE International Conference on Computer Communications. (INFOCOM 2006)*. 1–12. https://doi.org/10.1109/INFOCOM.2006.122

[26] D. Walton, A. Retana, E. Chen, and J. Scudder. 2016. Advertisement of Multiple Paths in BGP. RFC7911.

[27] Li Xiao, Jun Wang, and K. Nahrstedt. 2003. Reliability-aware IBGP route reflection topology design. *Proceedings - 11th IEEE International Conference on Network Protocols, ICNP* 2003-January (2003), 180–189.

[28] Randie Zhang and Micah Bartell. 2003. *BGP Design and Implementation*. Cisco Press, Indianapolis. 264–266 pages.

[29] Feng Zhao, Xicheng Lu, Peidong Zhu, and Jinjing Zhao. 2006. BGPSepD: An Improved Algorithm for Constructing Correct and Scalable IBGP Configurations Based on Vertexes Degree. In *Proceedings of the Second International Conference on High Performance Computing and Communications (HPCC'06)*. Springer-Verlag, Berlin, Heidelberg, 406–415. https://doi.org/10.1007/11847366_42

Chapter 3

A Combinatorial Optimization Framework for the Design of Resilient iBGP Overlays

In this chapter the research goes one step further designing an optimal topology when one link or node fails. In addition, experimental results with known network topologies are shown and compared with the nominal case, and the full-mesh design. Moreover, we demonstrate that designing an optimal route reflector topology is a $\mathcal{NP}$-hard problem.

A Combinatorial Optimization Framework for the Design of Resilient iBGP Overlays

Cristina Mayr
Instituto de Computación
Universidad de la República
Montevideo, Uruguay
mayr@fing.edu.uy

Claudio Risso
Instituto de Computación
Universidad de la República
Montevideo, Uruguay
crisso@fing.edu.uy

Eduardo Grampín
Instituto de Computación
Universidad de la República
Montevideo, Uruguay
grampin@fing.edu.uy

***Abstract*—The Internet is an aggregation of Autonomous Systems (ASes) which exchange network prefixes reachability advertisements using the Border Gateway Protocol (BGP). ASes set up external BGP (eBGP) sessions between the AS border routers (ASBR) of neighboring ASes, while internal BGP speakers establish internal Border Gateway Protocol (iBGP) sessions to learn reachability for external prefixes. In order to avoid loops in the control and forwarding plane, and to ensure complete visibility and path diversity, routers within the same AS must deploy full-mesh BGP sessions, which causes scalability problems, both in the number of sessions and the resources (memory, CPU) consumed by BGP routers. Route Reflection is a widely accepted alternative to improve scalability, but requires careful design, as new issues may be introduced, such as: increased probability of loops, divergence and routing sub-optimality. In our previous work we presented Optimal Route Reflector Topology Design (ORRTD), a combinatorial optimization approach to tackle the problem of designing a consistent and yet optimal iBGP overlay, which minimizes the number of Route Reflectors (RRs), guaranteeing that no sub-optimal route is chosen, i.e., the routes selected with the designated RRs are those that would have been selected if instead of having RRs, the iBGP speakers were fully meshed. In this paper we propose a modification to ORRTD that addresses resilience, i.e., survivability to node or link failures.**

***Index Terms*—Internet Routing, BGP, Route Reflection, Network Design, Combinatorial Optimization, BGP resilience**

I. Introduction

The inter-domain routing is supported by the Border Gateway Protocol (BGP, [16]), which is used to exchange reachability information among Autonomous Systems (ASes). Intra-domain routing is fulfilled by the interaction of the Interior Gateway Protocol (IGP) and Internal BGP (iBGP): while the IGP builds connectivity for internal prefixes, iBGP is used to determine the exit gateway for those packets whose destination is external to the AS. A router running external BGP (eBGP) sessions is called Autonomous System Border Router (ASBR), while a router running only iBGP sessions is called Internal Router (IR).

As described in our previous works [4], [5] and related work op.cit., to avoid BGP loops and make sure that the complete routing information is disseminated, a full mesh of iBGP sessions between each pair of routers in the AS is required, resulting in $\frac{n\times(n-1)}{2}$ iBGP sessions for a domain of n routers, imposing large CPU and memory requirements to hold the Rib-In tables.

Route reflection [1] is an alternative in which one or more routers within the AS are designated as Route Reflectors (RRs) and they are allowed to re-advertise routes learned from an internal peer to other internal peers, while the rest plays the RR client role. With route reflection the number of iBGP sessions decreases to $n-1$ when using a sole RR, where n is the number of iBGP routers. Since a unique RR in an AS constitutes a single point of failure, at least two routers are to be selected as RRs.

BGP route selection process combines IGP and BGP routing information, and consequently RRs decisions are influenced by their locations within the AS. The problem of selecting which routers will have the RR role, following a consistent set of client-RR adjacencies (i.e., the RR topology), is known as *iBGP overlay design problem.* and has been extensively explored as in [3], [7].

In previous works [4], [5] we presented *Optimal Route Reflector Topology Design* (ORRTD), a novel combinatorial optimization approach to tackle the problem of designing a consistent and yet optimal iBGP overlay for an AS. The optimality criterion is to use the minimum number of route reflectors (RRs) and sessions, maintaining *correctness* and *full mesh optimality* [2], [9], [19], assuming that all prefixes matching a common gateway, or a set of equally preferred gateways are clustered into *classes* of prefixes (or labels).

This paper complements [4], introducing resilience, and is organized as follows: Section II explains what is an iBGP overlay, describes some design solutions, and explains the basis of ORRTD, our novel solution to design the overlay, section III proposes a mathematical approach for RRs selection and explains how to introduce resilience in the model, Section IV presents experimental results over some network topologies, Section V discusses the problem complexity and finally, Section VI summarizes our main conclusions and lines for further research.

II. iBGP overlay design based on route reflection

The selection of the best BGP route at each router depends on the IGP path cost to the BGP next hop announcing the route due to the *Hot potato* routing, where the preferred route is the one with shortest IGP path (the closest exit point).

978-1-5386-8461-0/19/$31.00 ©2019 IEEE

Previous research works about RR selection, focus mainly in reliability, such as [22], [12], [19], or in reducing the number of sessions [24], or in modifying RR behavior or avoid them, as in [3], [14], [6]. Alternatives or variations to classical RR architecture have also been proposed to improve BGP reliability, including Multi-path [21], BGP Advertise-Best-External [10], Add-Path [21] and Diverse-Path [13].

In addition to the previously referred works, which mainly focus upon reliability issues associated with Route Reflection, this work also aims on those problems derived from the lack of optimality.

Control variables for designing a reliable iBGP route reflection topology should answer the following questions:

1) Which routers are to be chosen as route reflectors;
2) How clients are to be connected with route reflectors.

The objective function to be minimized counts the number of RRs, which also determines the number of BGP sessions. Constraints are introduced to avoid the problems described in [4], not only for steady/non-faulty state, but also to preserve such attributes after each possible single node or link failure. We will show in section V that the problem is NP-Complete.

The object of this work is to design a reliable iBGP overlay with minimum number of RRs and sessions, resilient to single node or link failure, and yet *route optimal* for a steady configuration of eBGP messages upon an internal given topology.

The technique introduced in this work is called *Optimal Route Reflector Topology Design* (ORRTD). It aims on being optimal in terms of routing, and in the number of RRs and sessions to keep. As we see in section III, the technique relies upon an integer programming problem formulation, whose constraints have been chosen to always select IGP optimal routes. In [4], [5], we demonstrate that this technique preserves correctness, and besides, *optimality principle* applies to all internal routers (by construction).

III. From the raw problem to its Integer Programing Formulation

This section introduces a mathematical formulation of the problem in two steps. The first (simpler) approach focuses upon optimization concerns of the problem. The second extends the basic formulation to integrate resilience to the design, which constitutes the main contribution of this article.

With ORRTD no additional functionality or BGP sessions are needed, and no changes to BGP process are suggested.

We consider an AS with a collection of BGP speaking routers, either IRs or ASBRs, connected by a pure IP network (i.e. hop-by-hop routing); and we want to determine which of the IRs will be designated as RRs. We assume that only an IR can be RR, every Client-IR (i.e. not RR) must be peer of a unique RR per class of prefixes, whenever optimal for some IR, every ASBR must be connected to at least one RR per class of prefixes and an ASBR cannot peer with a Client-IR. We also assume that all external prefixes are learned through BGP and they have been filtered according to the path selection algorithm to get to a set of prefix classes;

Suppose we have the graph associated with the network of some AS, like that represented in graph in Fig. 1.

This graph is undirected and the weight of each link is the IGP cost. ASBRs *A* and *AB* receive prefixes class *A*, while *B* and *AB* receive prefixes class *B*.

From that information, an optimal internal-to-border router graph can be build ([4], [5]) for each class of prefixes, which are respectively sketched on the left and right on Fig. 2. Complementary, IRs that share a common ASBR for a common prefix class, could serve as the reflector of each other for that class (Fig. 3).

A. Resilience considerations

Failures can occur at the links, or at the nodes (IRs or ASBRs). Even if the failure is in any IR or link, they could be in the shortest path calculated for deriving the graph in Fig. 2 and Fig. 3. To really ensure resilience, disjoint paths between each ASBR and the corresponding RR are needed.

In the present work we propose a solution when any element at a time fails, either a link, an IR, or an ASBR. We are interested in computing the smallest number of additional links (or nodes) that need to be added in order to increase the resilience of a network against random failures.

In order to make it possible to propose a resilient solution, we assume the original IGP graph is at least 2-node-connected, which translates into the existence of two node (and link) independent paths between every pair of nodes.

This guarantees in turn the existence of a detour against every possible single failure. More generally, k-edge (node) connectivity refers to the minimum number of edges (nodes) to be removed so that the graph becomes disconnected. Both problems are NP-complete.

If a graph is k-node-connected it can be proved that there are k node-disjoint paths between any pair of nodes.

B. Resilient ORRTD

Suppose we have a best p path from certain $u \in IR$ to $v \in ASBR$ for prefix class A. Let $p = u, x_1, ..., x_h, v$.

To add resilience to ORRTD we consider every type of failure:

1) *link failure* - an edge $e = (x_i, x_{i+1})$ fails. Suppose that, without this edge, the new closest ASBR from u is w. Then create a fictitious prefix class C_l advertised by w.

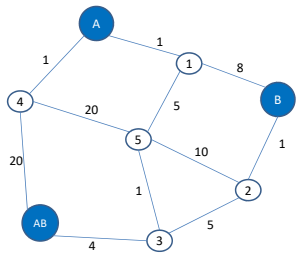


Fig. 1. Graph with 2 prefix classes (A and B)

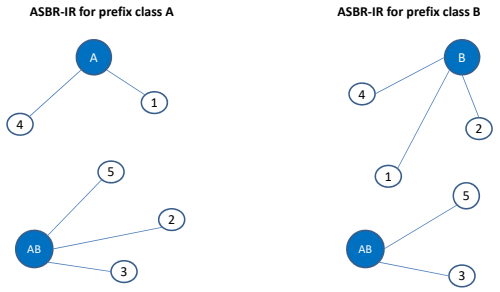


Fig. 2. IR-ASBR Adjacency graphs - 2 prefix classes (A and B)

2) *internal router failure* - if $x_i \in IR$ fails and it belongs to some best path p from another $u \in IR$ to $v \in ASBR$, then a new best path to some ASBR must be calculated. If the new best path ends in a different node $w \in ASBR$, proceed as in the previous case.
3) *border router failure* - if $v \in ASBR$ fails and it is the best exit for some $u \in IR$, proceed as in the first case.

We will analyze the case depicted in Fig. 1. The best ASBR for IR 5 and prefix class B is AB, by using the path $5-3-AB$. If the link $(5,3)$ from IR 5 to ASBR AB fails, then the best ASBR for IR 5 and prefix class B in $G' = (V, E \setminus (5,2))$ is B, an the path is $5-2-B$ (Fig. 4). Then we add a prefix class B_j advertised by B, and an affinity set of nodes corresponding to the new best path, similar to those presented in Fig . 2. Observe that we add only one *fictitious prefixes class* for each combination of: IR $\times$ original prefix class $\times$ new ASBR in failure scenario. This guarantees that each IR gets optimal prefixes for all of those ASBRs for which it is necessary to keep optimality after each possible node or link failure. Although now we have to ensure that this does not introduce sub-optimal paths to other IRs. This can be achieved by ensuring that the fictitious prefix classes appear only in the routing tables of the nodes belonging to the alternative path considered. Note that in this path, an ASBR can appear as an intermediate node. This does not introduce any problem, as the fictitious prefix class is advertised by eBGP, so the next hop remains unaltered. When considering the next steps, the prefix class comes through iBGP, so, unless the receiving router is a RR, it cannot re-advertise that prefix class. The next step consists in assembling all these pieces into a single combinatorial optimization problem.

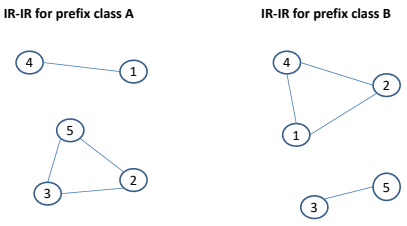


Fig. 3. Internal to Internal IR affinities graphs for prefixes classes A and B

$$
\begin{cases}
\min \sum_{i \in IR} x_i & & \\
\text{Subject to :} & & \\
\sum_{(ij) \in S'^k} y_{ij}^k \geq 1, & \forall i \in BR, k \in \mathcal{C}', S'^k \neq \emptyset & (i) \\
x_j - y_{ij}^k \geq 0, & \forall i \in BR, k \in \mathcal{C}', (ij) \in S'^k & (ii) \\
x_j + \sum_{(ij) \in T'^k} z_{ij}^k \geq 1, & \forall j \in IR, k \in \mathcal{C}' & (iii) \\
x_i + x_j - z_{ij}^k \geq 0, & \forall i \in IR, k \in \mathcal{C}' (ij) \in T'^k & (iv) \\
x_i + x_j + z_{ij}^k \leq 2, & \forall i \in IR, k \in \mathcal{C}' (ij) \in T'^k & (v) \\
\sum_{(jh) \in S'^k} y_{jh}^k - z_{ih}^k \geq 0, & \forall j \in IR, k \in \mathcal{C}' (ih) \in T'^k & (vi) \\
\sum_{i \in IR} x_i \geq 2, & \forall i \in IR & (vii) \\
w_{gh}^l \geq y_{ij}^k, & \forall i \in BR, j \in IR, k \in \mathcal{C}', g, h \in FC^l & (viii) \\
\sum_{(ij) \in P^l} y_{ij}^l \geq 1, & \forall i \in BR, l \in \mathcal{FC}, & (ix) \\
x_i, y_{ij}^k, z_{ij}^k, w_{gh}^l \in \{0,1\}, & \forall i,j \in V, k \in \mathcal{C}', l \in FC &
\end{cases}
\tag{1}
$$

Equations (1) have the following **input sets**:

$\mathcal{C}$	:	set of prefixes classes
$\{S^k\}$	:	set of border-to-internal BGP affinity matrices $S_{ij}^k = 1$ if and only if $j \in$ ASBR-to-IR for prefix class k, with $k \in \mathcal{C}$, $i \in$ BR, $j \in$ IR
$\{T^k\}$	:	set of internal-to-internal BGP affinity matrices
FC	:	set of fictitious prefix classes
$\{P^l\}$	:	set of new BGP best path nodes from internal-to-BR affinity matrices
$\{Q^l\}$	:	set of new BGP best path IR-to-IR affinity matrices

Equations (1) have the following **parameters** to support resilience:

BR	:	set of all Autonomous System Border Routers
IR	:	set of all Internal Routers
S'	:	$\{S^k\} \cup \{FC^l\}$
C'	:	$C \cup FC$
T'	:	$\{T^k\} \cup \{Q^l\}$

and the following boolean **variables**:

x_i	:	1 if router i is to be a RR and 0 otherwise;
y_{ij}^k	:	1 if ASBR i is to be iBGP adjacent to IR j for prefixes class k and 0 otherwise;
z_{ij}^k	:	1 if IR i is to be iBGP adjacent to IR j for prefix class k and 0 otherwise;
w_{gh}^l	:	1 if nodes $g, h \in P^l$, i.e., the alternative best path

The objective function in (1) pushes down to get the minimum number of RRs. But this objective has several

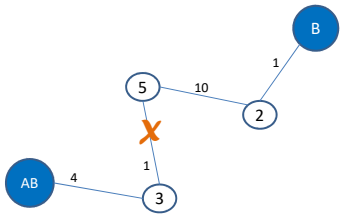


Fig. 4. New path in failure case, for prefix class B

constraints, stated in equation groups (i) to (ix). It is worth mentioning that RRs are globally selected, that is, they are common to all prefixes classes.

Equation groups (i) to (vi) are similar to those described in [4] and [5], but now considering the input sets and parameters described above; the rest of the equations are added to force resilience.

Equations (vii) ensure there is more than one RR, so the RR in not a single point of failure. Finally, equations $(viii)$ and (ix) ensure that only the nodes in the best alternative path learn the fictitious prefix classes.

For small networks, solutions can be easily found by brute-force or quasi-exhaustive methods. For more complex networks the problem can be solved with any popular solver like GLPK or CPLEX, even for hundreds of nodes, while the number of prefixes classes is limited. For even more complex problems, with hundreds of classes, a heuristic approach should be used.

In summary, the problem formulation has an augmented set of prefix classes. The new quantity of classes k' is k plus all the combinations of links and routers that can fail for each prefix class. In this new scenario, solution might not be found, given the increase on the number of restrictions. Anyway, in dense graphs, paths tend to repeat, so the problem can be preprocessed to eliminate redundant conditions. It is expected that there will not be too many additional prefix classes and so the number of constraints will not grow excessively.

IV. Experimental results

In this section we present early results obtained in the emulation environment proposed by [17] which is based on Quagga[1], MiniNExT[2] and ExaBGP[3] for injecting BGP messages. Some of the topologies were taken from "The Internet Topology Zoo" repository ([18]) and slightly adapted, for example, to ensure no vertex has degree one, as this makes finding a resilient topology design non-viable, and other topologies are theoretical cases.

For the purpose of this test we assume there are two prefix classes, and we know in advance which ASBRs advertise each prefix class. BGP updates from ASBRs will be set with the *next-hop self* option, as internal network does not know about external routes. The experimental results for the nominal case were presented in our previous work [4], [5]. In all cases ORRTD results in a reduced number of RRs.

In Table I we show, for two prefix classes, the resulting RRs applying ORRTD for the nominal case, and for the resilient case applied to different network topologies. We also show the number of equations needed for each topology. It can be easily seen that it quickly increases as the network becomes bigger. These early results show that in many cases the final quantity of RRs remains the same, and the change is about which IRs are chosen as RR, and an increased number of iBGP sessions established among the routers, as can be seen in Fig. 5. In other cases the number of RRs does increase. We observe that this strongly depends on the underlying topology. We remark that we assume RRs are in fact connected in a full-mesh, as it is the standard, so it is not introduced in the model as a constraint, and so it is not shown in Fig. 5. In Table II we show the reduction in the number of BGP sessions in the resilient version of ORRTD compared to full mesh.

V. Problem complexity

We show that finding a minimal solution in $ORRTD$ is at least as hard as finding a solution for Minimum Vertex Cover (MVC) problem, which is known to be NP-complete [8] and in fact APX-complete ([11]).

Formally, a vertex cover S of an undirected graph $G = (V, E)$ is a subset of V such that $uv \in E \Rightarrow u \in S \vee v \in S$. We consider ORRTD where there is just one prefix class, as it seems natural that if there are more prefix classes, the problem will be even more difficult. The decision version of both problems is as follows:

1) π' - Given an undirected graph $G' = (V', E')$ and a constant k, is there a subset S of V' such that $uv \in E' \Rightarrow u \in S \vee v \in S$ with size $\leq k$?
2) π - Given a weighted undirected graph $G = (V, E)$, where $V = IR \cup BR$ and a constant k, is there a subset $RR \subseteq IR$, with size $\leq k$ constructed with ORRTD?

A reduction from π' to π can be built as follows:

- for each vertex $v \in V'$ of π' there will be a vertex $v \in IR$
- for each edge $uv \in E'$: add an edge $uv \in E$ between a pair of vertices $u, v \in IR$ with weight 1, add a new vertex x_{uv} to V and a pair of edges from x_{uv} to u and from x_{uv} to v to E with weight 1.
- let $x_{uv} \in BR$. Then by construction, every IR is at distance 1 to some ASBR.

TABLE I
ORRTD - Comparison of Nominal and Resilient Case

Topology	# IRs	# ASBRs	RRs Nom.	RRs Resil.	Eqs. Nom.	Eqs. Resil.
Abilene	8	3	2	2	114	419
AB5	5	3	2	3	55	131
AB10	10	3	2	2	250	478
Airtel	3	6	1	2	32	131
Garr	47	7	4	4	3080	3852
UniC	24	3	2	2	902	944
Uran	18	5	3	5	599	687
Jgn2Plus	11	6	2	6	399	453

[1]Quagga Routing Suite. Available at: https://www.quagga.net/. Accessed: 2018-09-01

[2]MiniNExT (Mininet ExTended). Available at: shttps://www.quagga.net/. Accessed: 2018-09-01

[3]https://github.com/Exa-Networks/exabgp

TABLE II
COMPARISON OF BGP SESSIONS

Topology	resilient ORRTD	Full Mesh
Abilene	20	55
AB5	18	28
AB10	23	78
Airtel	26	36
Garr	331	1431
UniC	24	351
Uran	57	253
Jgn2Plus	64	136

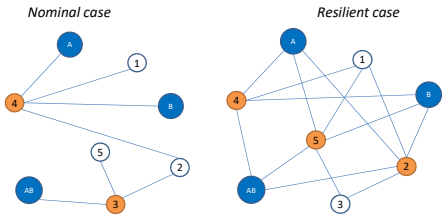


Fig. 5. More connections in resilient case

This graph in π has been designed to have a set of RRs if and only if a set cover exist in π', so $\pi' \preccurlyeq \pi$. Besides, this is a polynomial reduction.

VI. CONCLUSION

In this article we focus on the efficient usage of BGP, particularly in the intra-domain scope, though it suffers from serious scalability issues. With Route Reflection, a classic and simple approach, widely standardized over the Internet infrastructure, but requiring careful design, as it could lead to other kinds of issues, as described in section II. We based our proposal on overlay networks and present a novel mathematical approach to tackle several known problems of reflection, by means of a design that optimizes the scalability. The technique has been called Optimal Route Reflector Topology Design, or ORRTD for short. Among other advantages, with ORRTD there is no need to modify or augment existing BGP standards. Early experimental results in emulation environments demonstrate the theoretical consistency of ORRTD, even in the event of fails over single nodes or links. Besides, ORRTD outperforms other heuristic approaches, and according to our experimental results with known topologies, the number of RRs does not increase significantly, and even remains the same, while augmenting the BGP sessions needed.

It is also worth to mention that we assume that prefix classes categorization is a given input for the optimization process, and is done based on ISP policies, either static or dynamically. This classification may constitute a whole line of research, for example, using machine learning or other techniques to build the prefixes classes based on the dynamics of BGP updates. We also prove ORRTD is a NP-hard problem, which implies that when considering larger instances of the problem, some heuristic approaches should be considered to solve it, which introduces a new line for future research.

REFERENCES

[1] T. Bates, E. Chen, and R. Chandra, "BGP route reflection: an alternative to full mesh internal bgp (iBGP)," RFC4456 (Draft Standard), 2006.
[2] M. O. Buob, S. Uhlig, and M. Meulle, "Designing Optimal iBGP Route-Reflection Topologies," LNCS4982, 2008.
[3] C. Pelsser, A. Masuda, and K. Shiomoto, "A novel internal BGP route distribution architecture," in IEICE General Conference, 2010.
[4] C. Mayr, and E. Grampín, and C. Risso,"Optimal route reflection topology design," in 10th Latinamerican Networking Conference (LANC '18). ACM, New York, NY, USA, pp. 65-72, 2018.
[5] C. Mayr, and E. Grampín, and C. Risso,"Optimal route reflection topology design," *Technical Report*, https://www.fing.edu.uy/~crisso/Optimal_Route_Reflection_Topology_Design.pdf, 2018.
[6] S. S. M Dakshayini,"Effect of route reflection on iBGP convergence and an approach to reduce convergence time," in International Journal of scientific research and management (IJSRM), 2016, vol. 4 (8).
[7] A. Flavel,"BGP, not as easy As 1-2-3," Ph.D. Dissertation, 2009, University of Adelaide, Australia.
[8] M. R. Garey and D. S. Johnson,"Computers and intractability: a guide to the theory of NP-completeness," W. H. Freeman, San Francisco, 1979.
[9] T. G. Griffin and G. Wilfong,"On the correctness of iBGP configuration," in SIGCOMM '02 Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 17–29, New York, NY, USA, 2002. ACM.
[10] P. Marques, R. Fernando, E. Chen, P. Mohapatra, and H. Gredler,"Advertisement of the best external route in BGP, " draft-ietf-idr-best-external-05, 2017.
[11] C. H. Papadimitriou, and M. Yannakakis,"Optimization, approximation, and complexity classes," Journal of Computer System Sciences 43, pp. 425-440, 1991.
[12] J.H. Park. ,"Understanding the impact of internal BGP route reflection," PhD thesis, University of California, 2011.
[13] R. Raszuk, R. Fernando, K. Patel, D. McPherson, and K. Kumaki,"Distribution of diverse BGP paths," rfc6774, 2012.
[14] M. H. Raza, A. K. Kansara, and W. Robertson,"Effective iBGP operation without a full mesh topology," in International Refereed Journal of Engineering and Science (IRJES), vol. 5(8), pp. 16–23, 2016.
[15] E. Gutiérrez and D. Agriel and E. Saenz and E. Grampín,"RRLOC: A tool for iBGP RR topology planning and experimentation," in IEEE Network Operations and Management Symposium, NOMS 2014, Krakow, Poland, May 5-9, 2014, pp. 1–4. IEEE.
[16] Y. Rekhter and T. Li,"A Border Gateway Protocol 4 (BGP-4)," RFC1771 Obsoleted by RFC 4271, 1995.
[17] V. Solla, G. Jambrina and E. Grampín,"Route reflection topology planning in service provider networks," in 2017 IEEE URUCON, pp.1–4.
[18] S. Knight, H.X. Nguyen, N. Falkner, R. Bowden and M. Roughan,"The Internet Topology Zoo," Selected Areas in Communications, IEEE Journal, 2011, October, vol. 29(9), pp. 1765 -1775.
[19] S. Vissicchio, L. Cittadini, L. Vanbever, and O. Bonaventure,"iBGP deceptions: more sessions, fewer routes," in INFOCOM, IEEE, 2012.
[20] M. Vutukuru and P. Valiant and S. Kopparty and H. Balakrishnan,"How to construct a correct and scalable iBGP configuration," in Proceedings of the 25th IEEE International Conference on Computer Communications. INFOCOM, 2006, pp. 1-12.
[21] D. Walton, A. Retana, E. Chen, and J. Scudder,"Advertisement of multiple paths in BGP," RFC7911, July 2016.
[22] L. Xiao, J. Wang, and K. Nahrstedt,"Reliability-aware iBGP route reflection topology design," Proceedings - 11th IEEE International Conference on Network Protocols, ICNP, January 2003, pp.180–189.
[23] F. Zhao and X. Lu and P. Zhu and J. Zhao,"BGPSepD: an improved algorithm for constructing correct and scalable iBGP configurations based on vertexes degree," in Proceedings of the second International. Conference on High Performance Computing and Communications., Springer-Verlag, 2006, pp. 406-415.
[24] R. Zhang and M. Bartell,"BGP design and implementation," Cisco Press, Indianapolis, 2003, pp. 264–266.

Chapter 4

Designing an Optimal and Resilient iBGP Overlay with extended ORRTD

In this article a relaxation is introduced, allowing both internal and border routers to be eligible as RRs. The impact of Internet prefix categorization is considered in the proposed solution and show the resulting route reflectors and BGP sessions with different number of prefixes classes, comparing the performance of the extended ORRTD with the previous version of ORRTD.

Designing an Optimal and Resilient iBGP Overlay with extended ORRTD

Cristina Mayr[1 [0000−0002−6245−2999]], Claudio Risso[1,2 [0000−0003−0580−3083]], and Eduardo Grampín[1,3 [0000−0001−6046−0023]]

[1] Instituto de Computación, Universidad de la República, Montevideo, Uruguay
[2] mayr@fing.edu.uy
[3] crisso@fing.edu.uy
grampin@fing.edu.uy

Abstract. The Internet is composed of the interconnection of several thousands Autonomous Systems (ASes), which are networks under a single administrative domain such as corporations, service providers, universities, and content providers, among others. To ensure communication between users and applications it is necessary that the routers of the different Autonomous Systems have reachability towards the IP addresses of the endpoints of this extremely decentralized network. The Border Gateway Protocol (BGP) is the responsible of learning and distributing this reachability information among ASes. Unlike other routing protocols, BGP routers communicate over point-to-point BGP sessions over TCP, administratively set. BGP sessions are either external (eBGP, between routers of different ASes, a.k.a. Border Routers, or ASBRs) or internal (iBGP, between routers which belong to the same AS). eBGP is needed to exchange reachability information among ASes, while iBGP makes it possible for internal routers to learn this information in order to forward IP packets to the appropriate ASBRs. To make sure that the whole information is learnt and no traffic deflection occur, a full-mesh of iBGP sessions is required among routers within each AS, which causes scalability issues. Although Route Reflectors (RR) can be used to improve performance, designing a correct, reliable and consistent iBGP overlay of sessions whith RRs is a delicate, far from easy task for ASes engineers, even though several popular heuristics are common practice. In previous works we proposed combinatorial optimization models to design consistent and resilient BGP overlays, when only non-Border-Routers are eligible for RRs. The present work extends previous models to allow any router (including Border Routers) to be Route Reflectors.

Keywords: Network Overlay Design · Route Reflection, BGP · Internet Routing · Combinatorial Optimization · BGP resilience · Network Resilience · Internet Prefix Classes · Border Routers

1 Introduction

Autonomous Systems (ASes) are networks or sets of networks under a single and clearly defined external routing policy. The Internet is composed of the inter-

connection of several thousands ASes, which use the Border Gateway Protocol (BGP, [1]) to exchange network prefixes (aggregations of IP addresses) reachability advertisements. BGP advertisements (or updates) are sent over BGP sessions administratively set between pairs of routers. Those sessions have two variants: internal BGP (iBGP) is used between routers belonging to the same AS, and external BGP (eBGP) when the routers belong to different ASes. In the last case, BGP routers are called Autonomous System Border Routers (ASBRs), while those running only iBGP sessions are referred to as Internal Routers (IRs).

Global IP reachability information is acquired using BGP, but each AS also needs to deploy an Internal Gateway Protocol (IGP) intra-domain, so as to know the internal topology, and guarantee the delivery of IP packets within the domain. BGP is tightly tied to the IGP; indeed, when there are more than one next-hop option for a given IP prefix, the BGP decision process compares different attributes to break the tie, eventually reaching the IGP metric to the next-hop ("hot potato routing").

Traditional iBGP implementations require a full-mesh of sessions among routers of each AS. This is due to the *split horizon* rule, under which iBGP routers do not re-advertise routes learned via iBGP to other iBGP peers. As a result, a number of $\frac{n\times(n-1)}{2}$ iBGP sessions is needed for an AS with n routers. *Route Reflection* [2] is used as an alternative to reduce BGP sessions and gain efficiency in CPU and memory usage: one or more routers within the AS are designated as Route Reflectors (RRs) and they are allowed to re-advertise routes learned from an internal peer to other internal peers. The rest of the routers are *clients* of some RR. A *client* is an iBGP router that the RR will reflect routes to. Note that RRs re-advertise only best routes after running their own decision process, and also note that re-advertisements are biased by the placement of RRs within AS's topology, because as it was previously mentioned, prefixes selection considers IGP metric during the BGP path-selection. Routing is called *FM-optimal* whenever for the selected prefixes the gateways are those routers that would have been chosen under a full-mesh overlay. The *iBGP overlay design problem* consists in deciding which routers are to be route reflectors, and what sessions are to be established between clients and those RRs. Route Reflection has been extensively studied ([3], [4], [5]) with respect to reliability. There are also previous researches about how to locate the RRs ([6], [7], [8]).

Our approach for the optimization problem consist in minimizing the number of RRs in such a way that reliability is preserved and no sub-optimal route is chosen. We have studied the problem not only in the nominal case ([9]) but also the resilient case (see [10], [11]) when any single link or one router fails, and we have proposed a technique called Optimal Route Reflector Topology Design (ORRTD) to minimize the number of route reflectors that could be chosen among the IRs. In our previous models, Internal Routers were the only candidates to become Route Reflectors, which is a realistic constraint for many Internet Service Providers (ISP). However, in ASes whose goal is providing connectivity to other ASes (transit networks), many routers are actually Border Routers, turning previous premises impractical. This work expands previous models to allow

ASBRs to become RRs. The new technique is called Extended ORRTD, and we have obtained promising results in experimental environment: as networks become larger, the number of RRs obtained can be significantly low.

This document is organized as follows. Section 2 presents iBGP overlay based on RRs. Section 3 describes the impact of allowing border routers to be eligible as RRs. Section 4 presents experimental results over some network topologies, Section 5 summarizes our main conclusions and lines for further research.

2 The iBGP Overlay

2.1 Protocol concepts

ISPs and most large networks use some dynamic routing protocol intra-domain, called Internal Gateway Protocols (IGP). The most popular are Open Shortest Path First (OSPF) and Intermediate System - Intermediate System (IS-IS), which are efficient and safe, and fall in the link-state protocol category, since they build a complete network state database, using flooding of link state information among the routers in the domain; link costs may take into account actual links parameters, such as bandwidth. On the other hand, inter-domain routing (i.e., routing among ASes) is a job for BGP, which is a *path-vector* protocol, meaning that without administrative policies, the metric that determines the shortest path for a given prefix is the number of ASes that a routing announcement have crossed (the *AS_PATH*), used as a *hop* metric. Each BGP announcement contains a number of *attributes* (either mandatory or optional) which characterizes the routing information contained in the announcement (called Network Layer Reachability Information - NLRI, or simply "a route"). Some of the well-known, mandatory attributes are the aforementioned *AS_PATH*, and *NEXT_HOP*, which is the IP address of the router from a neighbour AS which sent a particular route announcement to the ASBR (the exit point of the current AS). BGP routers usually receive multiple route alternatives for a given destination prefix, and therefore need to run the BGP decision process to select the best path. As mentioned above, this decision process eventualy reach the point where the route with the lowest IGP metric (the IGP cost) towards the BGP next hop is chosen. When using RRs, a hierarchy among the iBGP speakers is created by clustering a subset of iBGP speakers with each RR. RRs must form a full mesh among themselves in order to make all announcements reachable, and each client peers with only its RR. To ensure reliability and loop-less [12, 13], it is important to carefully design this overlay.

2.2 Optimal Route Reflector Topology Design (ORRTD)

An optimization model can be formulated as an integer programming problem to minimize the number of RRs and BGP sessions under certain conditions: constraints are introduced to avoid problems described in [9], not only for steady/non-faulty state, but also in the case of each possible single node or link

failure [10], [11], and they have also been chosen to always select IGP optimal routes. The technique is called *Optimal Route Reflector Topology Design* (ORRTD) and is described in detail in [9, 10]. To formulate the problem we need the following definitions:

1. The *internal-to-border (IR-ASBR)* graph represents the preferred border router from each IR, based on the IGP costs (the one with the lowest cost path).

2. The *internal-to-internal (IR-IR)* graph represents the affinity among IRs: IRs that share a common preferred ASBR for a common *class* of prefixes, could serve as the reflector of each other for that class.

3. *Prefix classes* are aggregations of IP prefixes whose prefixes are indifferent at the time of choosing one ASBR over another, i.e. they take the same decisions in the previous steps of the BGP best path selection algorithm.

For this problem, we assume that the prefix classes are built and known in advance.

The basic idea is to construct an optimal IR-ASBR and an IR-IR graph based on the network topology represented by an undirected graph where the weight of each link is the IGP cost, and derive the resulting connections constraints. Then we introduce a set of control variables to help decide which routers are to be chosen as route reflectors and how to connect clients to those route reflectors. ORRTD also takes into account the *prefix classes* that arrive through the ASBRs. In Fig.1 we have an example of a graph transformation where there are two prefix classes A (received by routers A and AB) and B (received by routers B and AB). Nodes 1 through 6 represent the internal routers. The leftmost graph is the original weighted graph and at the right part we have the corresponding IR-ASBR (for prefix classes A and B, the ASBRs with the lowest cost path for each IR) and IR-IR graphs for those prefix classes. For example, the preferred border router from internal router 5 for prefix class A are both A and AB, as the path cost is 6. The preferred ASBR for prefix class B from IR 5 is AB. So in both IR-ASBR sub-graphs for prefix class A and B there is an edge from 5 to AB, and besides, for prefix class A there is also an edge from 5 to A.

Based on the information provided by the new graphs, we construct an optimization model where the adjacencies are represented in the constraints.

3 Border Routers and Route Reflectors

In the previous version (see [9, 10]), only IRs could be chosen as RRs. We represent a network topology as an undirected weighted graph. So an optimal internal-to-border router graph is built ([9, 10]) for each class of prefixes; IRs that share

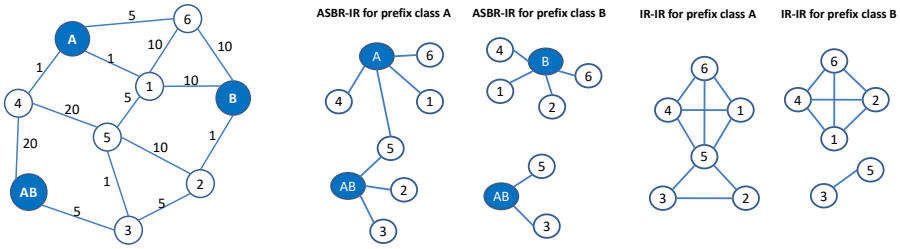


Fig. 1. Original graph and its IR-ASBR and IR-IR Adjacency graphs

a common ASBR for a common prefix class, could serve as the reflector of each other for that class. But the fact is that there are ASes that have most of the routers working as ASBRs, so introducing this relaxation has a practical interest.

A first option consists in applying the optimization model described in [10], ORRTD, to the graph representing the AS network but with the following adaptation: as border routers cannot be RRs, for each link from an ASBR to some other router, introduce a fictitious internal node in such a way that the cost from each ASBR to the fictitious node is 0, while the cost from the fictitious node to the original adjacent node is the real weight. To illustrate the situation, let's take the weighted graph shown in the left part of Fig. 2. Instead of the link from ASBR A to IR 4 with cost 1, we have a link from ASBR A to fictitious IR x_1 with cost 0, and another link from IR x_1 to IR 4 with cost 1. The dotted lines represent the fictitious links. Then connect all the fictitious nodes associated

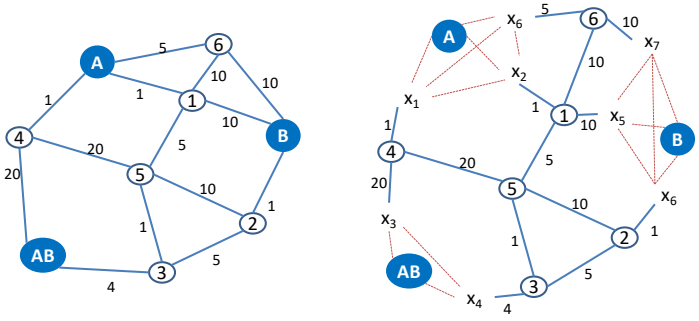


Fig. 2. Graph transformation with fictitious nodes and links

to the same ASBR in a full mesh. Once the new graph is obtained, we can find the IR-to-ASBR graph affinity by applying Dijkstra's algorithm, and continuing with the reasoning as described in [9, 10].

This simple transformation works fine for small networks, and a reduced number of ASBRs and prefix classes, but, as we have demonstrated in [10], this is a NP-hard problem, which means that when increasing the input as in

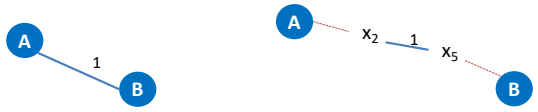


Fig. 3. Fictitious nodes and links for 2 adjacent ASBRs

the previous transformation, and when considering the resilient version of the problem where fictitious prefix classes are introduced for every alternative path to a different ASBR (even though failure of fictitious nodes is not considered, but failure of the links do apply), it becomes more difficult to solve. We can think of small improvements to the original implementation, like modifying Dijkstra's algorithm to get, among the shortest paths (by introducing the fictitious links with 0 cost there will be many of the same cost), the one with the lower quantity of edges, but anyway, the problem is still harder than the original one. So how many fictitious edges and nodes do we add? It seems reasonable to keep this quantity as low as possible. In Fig. 3 we show that we have one path with two fictitious routers and two fictitious links between the ASBRs. But when applying to the case shown in Fig.4, many fictitious nodes (the x_s) and links (the dotted lines) are needed, introducing complexity.

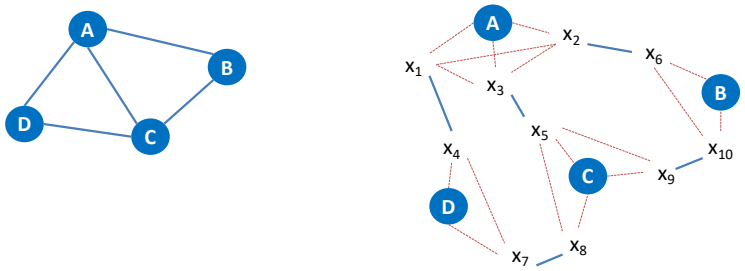


Fig. 4. Fictitious nodes and links for 4 adjacent ASBRs

3.1 Extended ORRTD

Given the previous considerations regarding the problem size growth, the most convenient alternative is to adapt the problem formulation to the new condition: border routers can also be designated as route reflectors, which leads to a new set of constraints shown in (1). In this research we assume that any border router (ASBR) or internal router (IR) can be a RR. The objective function in (1) pushes down to get the minimum number of RRs, where every router in the network could potentially be designated as route reflector. But this objective has several constraints, stated in equation groups (i) to (x). Equation groups (i) to (vii) ensure optimality, while those in groups $(viii)$ to (x) ensure resilience.

Equations (1) have the following **input sets**:

V : set of all routers, $V = \{IR \cup BR\}$
$\mathcal{C}$: set of prefixes classes
$\{S^k\}$: set of router-to-border BGP affinity matrices
$S^k_{ij} = 1$ if and only if $j \in$ ASBR-to-Router for prefix class k, with $k \in \mathcal{C}$, $i \in$ BR, $j \in$ IR $\cup$ BR
$\{T^k\}$: set of router-to-router BGP affinity matrices
FC : set of fictitious prefix classes
$\{P^l\}$: set of new BGP best path nodes from ASBR-to-Router affinity matrices
$\{Q^l\}$: set of new BGP best path Router-to-Router affinity matrices

Equations (1) have the following **parameters** to support resilience:

BR : set of all Autonomous System Border Routers
IR : set of all Internal Routers
S' : $\{S^k\} \cup \{FC^l\}$, set of router-to- border router affinity for every prefix class, including the fictitious prefix classes
C' : $C \cup FC$
T' : $\{T^k\} \cup \{Q^l\}$

and the following boolean **variables**:

x_i : 1 if router $i \in BR \cup IR$ is to be a RR and 0 otherwise;
y^k_{ij} : 1 if ASBR i is to be iBGP adjacent to router $j \in BR \cup IR$ for prefixes class k and 0 otherwise;
z^k_{ij} : 1 if router $i \in BR \cup IR$ is to be iBGP adjacent to router $j \in BR \cup IR$ for prefix class k and 0 otherwise;
w^l_{gh} : 1 if nodes $g, h \in P^l$, i.e., the alternative best path

$$
\begin{cases}
\min \sum_{i \in IR} x_i \\
\text{Subject to :} \\
\sum_{(ij) \in S'^k} y_{ij}^k \geq 1, & \forall i \in BR, k \in \mathcal{C}', S'^k \neq \emptyset \quad (i) \\
x_i + x_j - y_{ij}^k \geq 0, & \forall i \in BR, k \in \mathcal{C}', (ij) \in S'^k \quad (ii) \\
x_i + x_j - y_{ij}^k \leq 1, & \forall i \in BR, k \in \mathcal{C}', \quad (iii) \\
 & (ij) \in S'^k \\
x_j + \sum_{(ij) \in T'^k} z_{ij}^k \geq 1, & \forall j \in IR \cup BR, k \in \mathcal{C}' \quad (iv) \\
x_i + x_j - z_{ij}^k \geq 0, & \forall i \in IR \cup BR, k \in \mathcal{C}' \quad (v) \\
 & (ij) \in T'^k \\
x_i + x_j + z_{ij}^k \leq 2, & \forall i \in IR \cup BR, k \in \mathcal{C}' \quad (vi) \\
 & (ij) \in T'^k \\
\sum_{(jh) \in S'^k} y_{jh}^k - z_{ih}^k \geq 0, & \forall j \in IR, k \in \mathcal{C}' \quad (vii) \\
 & (ih) \in T'^k \\
\sum_{i \in IR \cup BR} x_i \geq 2, & \forall i \in IR \cup BR \quad (viii) \\
w_{gh}^l \geq y_{ij}^k, & \forall i \in BR, j \in IR \cup BR, \\
 & k \in \mathcal{C}', g, h \in FC^l \quad (ix) \\
\sum_{(ij) \in P^l} y_{ij}^l \geq 1, & \forall i \in BR, l \in \mathcal{FC} \quad (x) \\
x_i, y_{ij}^k, z_{ij}^k, w_{gh}^l \in \{0,1\}, \forall i, j \in V, k \in \mathcal{C}', l \in FC
\end{cases}
\tag{1}
$$

4 Experimental results

In this section we analyze the results of applying the proposed theoretical model to a selection of network topologies. Some of the topologies were taken from "The Internet Topology Zoo" repository ([16]) and slightly adapted to ensure 2-node connectivity (by introducing the minimum number of additional edges) to make finding a resilient topology design viable; other topologies are theoretical cases. They can be found in [17], where we show, for each topology, nodes, edges and the IGP costs. The models were solved with CPLEX Optimization Studio V12.6.3, running on an Intel Core I7 2.3GHz and 8 Gb RAM. For the purpose of this test we assume there are: four, ten, fifty and one hundred prefix classes, and that we know in advance which ASBRs advertise each prefix class.

Table 1 shows the results of applying the original model, ORRTD, with the addition of fictitious nodes and links, and the new model or extended ORRTD

(eORRTD for short), varying from four to one hundred prefix classes to the AB topology of Fig. 2. The leftmost columns of Table 1 show the network name, the number of border routers and internal routers and the number of prefix classes. The rest of the columns are the results obtained: number of RRs, number of BGP sessions, and number of equations both of the extended ORRTD technique and the original one (ORRTD) with the addition of fictitious nodes. The number of resulting RRs is the same (three route reflectors) in all cases and the number of BGP sessions is clearly better than in a full mesh (that would have fifty six sessions), but the quantity of equations grows faster when adding the fictitious nodes and links.

Table 1. Comparison of models

Network	**# BRs**	**#IRs**	**#PCs**	**#RRs**	**BGP sessions**	**Eqs. eORRTD**	**Eqs. ORRTD with fictitious nodes**
AB5	3	5	4	3	18	331	1324
AB5	3	5	10	3	18	666	2066
AB5	3	5	50	3	19	2742	8073
AB5	3	5	100	3	19	5404	16141

Table 2 shows the resulting number of route reflectors and BGP sessions for different network topologies, when having four, ten and fifty prefix classes, and allowing ASBRs to be RRs. It can be appreciated that the number of BGP sessions is significantly lower than the necessary BGP session in full mesh. The last column also shows the evolution of the number of equations in the model, which grows, depending on the size of the network and the quantity of prefix classes.

Table 3 shows that for the bigger networks studied, and supposing all ASBRs are eligible, the number of RRs can be reduced with eORRTD. The problem can be solved with any popular solver like GLPK or CPLEX. Fig. 5 shows an example of the execution time of the solver for different number of prefix classes, setting y-axis as log10 for better visualization. For more complex problems, with hundreds of classes, a heuristic approach should be used.

We also present results obtained in the emulation environment proposed by [18] which is based on Quagga[4], MiniNExT[5] and ExaBGP[6] for injecting BGP messages. After certain period of injection, we wait for the BGP network to become stable, i.e., best routes are selected after applying routing policies in each router, and then analyze the BGP tables. We study the content of the LOC_RIB table of each router in the network. LOC_RIB table contains the best route out of all those available for each distinct destination, and the NEXT_HOP attribute

[4] Quagga Routing Suite. Available at: `https://www.quagga.net/`. Accessed: 2018-09-01

[5] MiniNExT (Mininet ExTended). Available at: s`https://www.quagga.net/`. Accessed: 2019-03-01

[6] `https://github.com/Exa-Networks/exabgp`

Table 2. eORRTD: RRs, BGP sessions and equations - Different Topologies

Network	# BRs	#IRs	#PCs	#RRs	BGP Sessions	BGP Sessions FM	#Eqs.
AB5	3	5	4	3	18	28	331
AB10	3	10	4	2	23	78	1361
Abilene2	3	8	4	2	16	55	856
Cernet2	4	37	4	3	112	820	15765
Garr2	7	48	4	7	246	1485	11233
Jgn2Plus2	6	11	4	5	45	136	2598
UniC	3	24	4	3	68	351	8811
Uran	5	18	4	4	48	253	4448
WideJpn	11	19	4	5	77	435	5452
TtNew	4	96	4	6	532	4950	91499
AB5	3	5	10	3	18	28	666
AB10	3	10	10	2	24	78	2418
Abilene	3	8	10	3	25	55	1560
Cernet2	4	37	10	3	112	820	30784
Garr2	7	48	10	7	272	1485	26634
Jgn2Plus2	6	11	10	5	47	136	4635
UniC	3	24	10	3	69	351	13974
Uran	5	18	10	4	67	253	8700
WideJpn	11	19	10	5	89	435	10107
TtnNew	4	96	10	6	561	4950	173009
AB5	3	5	50	3	19	28	2742
AB10	3	10	50	2	24	78	9015
Abilene	3	8	50	3	26	55	5868
Cernet2	4	37	50	3	112	820	142748
Garr2	7	48	50	7	265	1485	103263
Jgn2Plus2	6	11	50	6	79	136	18041
UniC	3	24	50	3	68	351	46740
Uran	5	18	50	5	78	253	35354
WideJpn	11	19	50	5	113	435	36686
TtnNew	6	94	50	6	573	4950	789153

Table 3. ORRTD vs eORRTD

Network	# BRs	#IRs	#PCs	#RRs eORRTD	#RRs ORRTD
TtNew	6	94	10	6	18
SwitchL3	12	30	10	3	3
Garr2	7	48	10	7	28
Uran	5	18	10	4	5

(the preferred exit router for each prefix class). Whenever a routing prefix is received from a neighbor, BGP process decides if any of the neighbors new routes are preferred to routes already installed in the LOC_RIB and it replaces it as required. For the purpose of the experiment, we consider AB topology shown in Fig. 2 described in [10] and one hundred prefix classes, each one represented by one prefix, e.g. 195.66.4.0/24. For each router in that network and every prefix class, we compare the LOC_RIB table content in the emulation environment when applying the BGP overlay design resulting from the new model eORRTD, and ORRTD with fictitious nodes and links.

In Table 4 we show an extract of the LOC_RIB table, which is completely coinciding for both overlay designs.

Table 4. LOC_RIB table for topology AB

Prefix Class	**Next_Hop**	**Router**	**Id**
195.66.1.0/24	192.168.0.8	2	1
195.66.2.0/24	192.168.0.7	2	2
195.66.3.0/24	192.168.0.7	2	3
195.66.4.0/24	192.168.0.7	2	4
195.66.5.0/24	192.168.0.7	2	5
195.66.6.0/24	192.168.0.7	2	6
.....	.....	..	..
195.66.100.0/24	192.168.0.7	1	354
195.66.1.0/24	172.16.3.2	AB	382
195.66.2.0/24	172.16.3.2	AB	383
195.66.3.0/24	172.16.3.2	AB	384

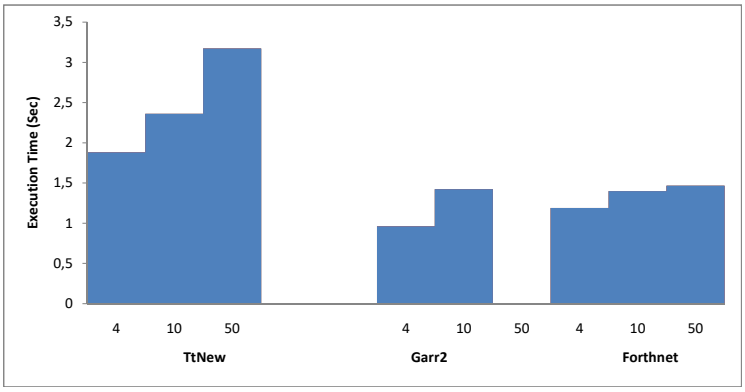


Fig. 5. Execution Time for 4, 10 and 50 Prefix Classes

5 Conclusion

In this article we focus on the efficient use of BGP, particularly in the intra-domain scope with Route Reflection. We describe an Integer Programming Problem to select the route reflectors, allowing both, internal and border routers to be eligible as RRs. We also consider the impact of Internet prefix categorization in the proposed solution and show experimental results with different number of prefix classes. The technique has been called *extended Optimal Route Reflector Topology Design*, or eORRTD for short. Among other advantages, with eORRTD there is no need to modify or augment existing BGP standards. Experimental results in emulation environments demonstrate the theoretical consistency of eORRTD, in the nominal case and in the event of fails over single nodes or links. In previous works ([9], [10]) we have shown that ORRTD outperforms other heuristic approaches, both in the number of route reflectors and the BGP sessions needed. In the present paper we show eORRTD outperforms full-mesh and ORRTD. Besides, we analyze the impact of increasing number of Internet prefix classes arriving to different border routers in the design of the best solution.

Finally, an important line of future research work is the construction of the prefix classes using the routing information, eventually considering the dynamic classification of prefixes. We are actually working on real data from an international provider; as a preliminary result, if we take the AS_PATH as in [14, 15], the number of prefix can be reduced in one order of magnitude. This may constitute a whole line of research, for example, using machine learning or other techniques to build the prefixes classes based on the BGP updates.

References

1. Rekhter, Y., Li, T.: A Border Gateway Protocol 4 (BGP-4), RFC1771 Obsoleted by RFC 4271 (1995).
2. Bates, T., Chen, E., and Chandra, R.: BGP route reflection: an alternative to full mesh internal bgp (iBGP), RFC4456 (Draft Standard) (2006).
3. Xiao, L., Wang, J., Nahrstedt, K.: Reliability-aware iBGP route reflection topology design, Proceedings - 11th IEEE International Conference on Network Protocols, ICNP, pp.180–189 (2003).
4. Park, J.H.: Understanding the impact of internal BGP route reflection, PhD thesis, University of California (2011).
5. Vissicchio, S., Cittadini, L., Vanbever, L., Bonaventure, O.: iBGP deceptions: more sessions, fewer routes, in INFOCOM, IEEE (2012).
6. Vutukuru, M., Valiant, P., Kopparty, S., Balakrishnan, H.: How to construct a correct and scalable iBGP configuration, in Proceedings of the 25th IEEE International Conference on Computer Communications. INFOCOM, pp. 1-12 (2006).
7. Zhao, F., Lu, X., Zhu, P., Zhao, J.:BGPSepD: An improved algorithm for constructing correct and scalable iBGP configurations based on vertexes degree, in Proceedings of the second International. Conference on High Performance Computing and Communications., Springer-Verlag, pp. 406-415 (2006).
8. Zhang, R., Bartell, M.: BGP design and implementation, Cisco Press, Indianapolis, pp. 264–266 (2003).

9. Mayr, C., Grampín, E., Risso, C.: Optimal route reflection topology design, in 10th Latinamerican Networking Conference (LANC '18). ACM, New York, NY, USA, pp. 65-72 (2018).
10. Mayr, C., Grampín, E., Risso, C.: A Combinatorial Optimization Framework for the Design of Resilient iBGP Overlays, in 15th International Conference on the Design of Reliable Communication Networks (DRCN'19) (2019). *To be published*
11. Mayr, C., Grampín: E., Risso, C.: A Combinatorial Optimization Framework for the Design of Resilient iBGP Overlays, *Technical Report*, `http://www.fing.edu.uy/~crisso/PID5740833.pdf` (2019).
12. Flavel, A.: BGP, not as easy As 1-2-3, Ph.D. Dissertation, University of Adelaide, Australia (2009).
13. Pelsser, C., Masuda, A., Shiomoto, K.: A novel internal BGP route distribution architecture, in IEICE General Conference (2010).
14. Broido, A. , kc claffy: Analysis of RouteViews BGP data: policy atoms, Cooperative Association for Internet Data Analysis - CAIDA, San Diego Supercomputer Center, University of California, San Diego. In Proceedings of NRDM workshop Santa Barbara (2001).
15. Afek, Y., Ben-Shalom, O., Bremler-Barr, A.: On the structure and application of BGP Policy Atoms, In Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment (IMW '02). ACM, New York, NY, USA, 209-214. DOI=http://dx.doi.org/10.1145/637201.637234 (2002).
16. Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., Roughan, M.: The Internet Topology Zoo, Selected Areas in Communications, IEEE Journal, October, vol. 29(9), pp. 1765 -1775 (2011).
17. Mayr, C., Grampín: E., Risso, C.: Examples of Internet Topologies, `https://www.fing.edu.uy/~crisso/Topologies.pdf`, 2019.
18. Solla, V., Jambrina, G., Grampín, E.: Route reflection topology planning in service provider networks, in 2017 IEEE URUCON, pp.1–4 (2017).

Part II

A combined BGP and IP/MPLS resilient transit backbone design

Chapter 5

A combined BGP and IP/MPLS resilient transit backbone design

This article focus upon the optimization of both control and forwarding planes in an Internet backbone network. A novel approach is proposed, which relies on optimization in two stages: the first one finds, after one node fails, the optimal and yet resilient choice for route reflectors and its BGP connections, and the second one applies traffic engineering to make the best possible use of the tunnels taking into account traffic demands and guarantying resilience for link failures. Additionally, the problem formulation allows to perform a sensitivity analysis to determine the impact of individual capacity changes in the maximum network capacity.

A combined BGP and IP/MPLS resilient transit backbone design

Claudio Risso
Instituto de Computación
Universidad de la República
Montevideo, Uruguay
crisso@fing.edu.uy

Cristina Mayr
Instituto de Computación
Universidad de la República
Montevideo, Uruguay
mayr@fing.edu.uy

Eduardo Grampín
Instituto de Computación
Universidad de la República
Montevideo, Uruguay
grampin@fing.edu.uy

***Abstract*—The Internet is a collection of interconnected Autonomous Systems (ASes) that use the Border Gateway Protocol (BGP) to exchange reachability information. The design of an optimal BGP overlay for an AS is a known NP-Hard problem this team tackled previously for IP networks, i.e, for the best effort paradigm. However, most Internet providers implement their backbones by combining IP routing with MPLS (Multiprotocol Label Switching) for QoS-aware traffic forwarding. MPLS forwarding incorporates traffic engineering and more efficient failover mechanisms. The present work introduces a coordinated design of both IP/MPLS substrates. Our contribution is on proposing an optimal and yet resilient topology design for an IP/MPLS Internet backbone, which takes advantage of traffic engineering features to optimize the demands, maintaining the aforementioned iBGP overlay optimality.**

***Index Terms*—Network Overlay Design, Route Reflection, BGP, Internet Routing, Combinatorial Optimization, BGP resilience, IP Prefixes, Internet Prefix Classes, MPLS, multi-protocol label switching, Traffic Engineering**

I. Introduction

The public Internet consist in a interconnection of Autonomous Systems (ASes), which are networks or sets of networks under a single and clearly defined external routing policy. While there are ASes which mostly give access to end users (local and regional Internet Service Providers - ISPs), others provide a transit service, which is the ability to route traffic from one AS to another AS, eventually reaching its final destination. Other major players in nowadays Internet are Content Providers, which partially utilize transit providers, but mostly rely on their own private resources to steer their internal traffic; another important architectural element are Internet Exchange Points (IXPs), where transit and content providers can establish peering agreements for traffic exchange using the Border Gateway Protocol (BGP, [1]), which is the standard protocol to exchange network prexes reachability advertisements among ASes. In his paper we are mostly concerned with traditional transit providers, but it worth mentioning that the advent of Content Providers and their ubiquitous presence in every corner of the Internet pose huge challenges for BGP.

BGP peering among Autonomous System Border Routers (ASBRs) in neighbour ASes is called External BGP (eBGP), while peering among routers inside the same AS (Internal Routers - IRs) is called Internal BGP (iBGP). In order to make sure that internal transport of BGP info is loop-free (control plane), and internal routing is coherent (loop-free data plane forwarding), the following iBGP advertisement rules must be observed: 1) prefixes learned from an eBGP neighbour can be re-advertised to an iBGP neighbour, and vice versa, and 2) prefixes learned from an iBGP neighbour cannot be re-advertised to another iBGP neighbour. While the 1st rule ensures that the complete routing information is disseminated, the 2nd rule prevent BGP announcements from looping, since iBGP cannot rely on the AS_PATH attribute to detect loops, because this attribute remains unchanged intra-domain. The practical implication of this rule is that a full mesh of iBGP sessions between each pair of routers in the AS is required, resulting in $\frac{n\times(n-1)}{2}$ iBGP sessions for a domain of n routers. Furthermore, the routing state (i.e. the size of iBGP Rib-In routing table) can be order n larger than the number of best routes (i.e., for T best routes, Rib-In size can reach up to $n \times T$ entries), imposing large CPU and memory requirements to every router in the AS. *Route Reflection* [2] is used as an alternative to reduce BGP sessions, and also seeking to gain efficiency in CPU and memory usage: one or more routers within the AS are designated as Route Reflectors (RRs) and they are allowed to re-advertise routes learned from an internal peer to other internal peers (contravening rule #2). Route Reflection can introduce a new set of problems, including routing, forwarding and dissemination correctness; these problems are deeply analyzed in: [3], [4], [5] and [6].

The *iBGP overlay design problem* consists in deciding which routers will be route reflectors, and how to assign their clients. Unfortunately, checking the correctness of an iBGP graph is NP-complete [7], but nevertheless, several algorithms have been proposed to locate RRs [8], [9], [10], including [11], which introduces the concept of *full-mesh optimality*.

We have studied the iBGP overlay design problem in pure IP networks, not only the nominal case [12], [13], but also the resilient case [14] (supporting a single link or one router failure), and we have proposed a combinatorial model to minimize the number of route reflectors that could be chosen among the IRs. We also introduced a relaxation where ASBRs can be eligible as RRs [15]. The methodology, called *Optimal Route Reflector Topology Design* (ORRTD), is based in the idea of *prefix classes*, which are aggregation of IP prefixes received by ASBRs.

So far, we tackled the *control plane* problem in a transit provider scenario, but in pure IP networks is hard to perform traffic engineering, and shortest path algorithms may poorly balance link occupation: some links may be congested while others are kept under-utilized; therefore, we must provide solutions for the *forwarding plane*. Multi Protocol Label Switching (MPLS) [16] is typically used to provide traffic engineering in IP backbones, and has been common practice for the last two decades. In this case, however, we have to solve the forwarding plane problem without degradation of the control plane solution. Therefore, a multi-layer approach is needed.

A typical IP backbone comprises several layers: i) the optical fiber network also referred to as the physical layer, ii) the IP data network or logical layer is compounded of routers and links among them (implemented over the physical layer), iii) the IP/MPLS tunnels (implemented over the logical layer), and iv) the set of traffic engineered paths assigned to each one of those tunnels (e.g. primary and secondary paths); in parallel we need to consider v) the iBGP overlay. Note that the behaviour of BGP is strongly related to the IGP used in the AS, given that one of the rules of the BGP process considers the IGP metric to decide the best exit point (*next hop*) to reach some Internet prefix. In the following chapter we will address these overlays in finer detail.

The rest of the paper is organized as follows: Section II presents iBGP and MPLS overlays. Section III presents the two stage mathematical model to optimize RRs and tunnel usage in IP/MPLS networks, Section IV presents the results when applying the model to an international provider network topology, Section V summarizes our main conclusions and lines for further research.

II. The BGP and MPLS Overlays

BGP and MPLS are, in principle, unrelated technologies: BGP is a *path-vector* policy routing protocol, while MPLS is a forwarding technique in between link and routing layer. BGP have been used for more than twenty-five years as the de-facto reachability protocol in the Internet, while MPLS is used since the early 2000's as the de-facto forwarding technique in IP service provider backbones. However, both technologies have been progressively augmented, and there are some applications where both are needed to implement a solution, notably the BGP/MPLS IP Virtual Private Networks (VPNs) [17]. Moreover, is common practice for transit providers to implement IP/MPLS traffic engineered tunnels among ASBRs. Thus, it is clear that a solution to the iBGP overlay design problem must take into account that the forwarding underlay is based on IP/MPLS, and therefore, it must be explicitly considered.

The BGP decision process is run when there is more than one next-hop option for a given prefix. For a prefix learnt from different ASBRs, and without explicit preferences from the network administrators (i.e. "local-pref" is not used), and disregarding the Multi Exit Discriminator (MED) attribute, if there is a tie in the AS_PATH lenght (this attribute is a list of traversed ASes, used as a metric in the path vector algorithm), then the IGP cost to the different next-hops is considered (i.e., the IGP cost from a given IR to the ASBRs which disseminate that route); we used this tie-break in our previous work, where we have studied BGP resilience over pure IP networks, also demonstrating that this is NP-hard [13]–[15].

So let's disscuss the IP/MPLS underlay; MPLS is a packet-forwarding technology which uses *labels* for data forwarding decisions. The packets are assigned to a FEC (Forwarding Equivalence Class) in the ingress LER (Label Edge Router), and for each FEC an unidirectional LSP (Label Switch Path) is built between ingress and egress routers; this LSPs are often called "tunnels". The LSPs are built either following the IGP shortest paths, or using link costs and other optimization criteria (e.g. QoS attributes such as bandwidth or available capacity [18]); once a path is computed by either the ingress LER or other external entity (e.g. the Path Computation Element - PCE [19]), the LSP is *signalled* by a label distribution protocol. LDP (Label Distribution Protocol) is used in the simplest case, where the LSPs follow the IGP shortest paths, while RSVP-TE (ReSerVation Protocol with Traffic Engineering extensions) is used to signal optimized LSPs; RSVP-TE also permits to implement Fast ReRoute (FRR) options. Note that in this case, the traffic demand matrix (the aggregated ingress to egress traffic flows across the network) must be known in advance. Therefore, when a packet is assigned to a FEC at the ingress point, the forwarding is entirely driven by the labels at the intermediate LSRs (Label Switch Routers).

MPLS networks with Traffic Engineering has been extensibly considered since the seminal article by Awduche et al. [21], including network management views such as [22] and design approaches, as for example [20], [23], [24], which integrates an overlay network design problem with the effective usage of trafficengineering features; this problem is NP-hard. There is a tight integration of MPLS traffic engineering with the IP routing protocols: MPLS-TE can automatically enhance the mesh of LSPs already established based on network topology discovered by IP routing protocols. As a consequence, the topology can be represented as a multi layer network.

Back to our problem, in the multi layer scenario we remark that the only type of failure to be considered in the BGP overlay are router failures. Link failures are not to be considered because link protection is encapsulated in the IP/MPLS overlay design. Nonetheless, IP/MPLS tunnels rely on the underlying level topology: if a physical link goes down, several tunnels may be affected, which implies that considering resilience in the IP/MPLS level is not independent from underlaying layers.

In summary, when using IP/MPLS, the BGP overlay is still in charge of ensuring router reliability for route dissemination, and for that purpose we use the aforementioned methodology ORRTD, which is formulated as an Integer Programming Problem to select the route reflectors and BGP sessions, considering that different Internet prefix classes arrive to each ASBR. ORRTD allows both, internal and border routers to be eligible as RRs, and it is resilient to a single node or link failure; as mentioned before, in the present work, ORRTD will consider only node failures. This means that when a router

fail, no routing updates are received from that router (i.e. it can not be considered net-hop for any prefix), while the links resilience problem is left to the IP/MPLS overlay design.

III. Designing the topology

In this paper we propose a two stage solution to build an optimal and resilient BGP overlay over an IP/MPLS backbone network. This backbone network uses LSP tunnels which allows traffic engineering, and they will be used to optimize the tunnels according to the traffic demand matrix. So in the first stage we apply the resilient version of ORRTD, but we leave link optimization to the second stage, were the specific characteristics of MPLS is taken into the model. For this problem, we assume that the Internet prefix classes that arrive to each border router are pre-built and known in advance.

A. First Stage - BGP overlay

BGP routers receive multiple paths to the same destination. The BGP best path algorithm decides which is the best path for each prefix class. In one of the steps the rule is to prefer the path with the lowest IGP metric towards the BGP next hop. So, if given two possible routes for a prefix, the preference is the same for all the steps prior to this one, then, the deciding factor on what the best path is, relies on the IGP cost.

$$
\begin{cases}
\min \sum_{i \in IR} x_i \\
\text{Subject to :} \\
\sum_{(ij) \in S'^k} y_{ij}^k \geq 1 & \forall i \in BR, k \in \mathcal{C}', S'^k \neq \emptyset \quad (i) \\
x_i + x_j - y_{ij}^k \geq 0 & \forall i \in BR, k \in \mathcal{C}', (ij) \in S'^k \quad (ii) \\
x_i + x_j - y_{ij}^k \leq 1 & \forall i \in BR, k \in \mathcal{C}', (ij) \in S'^k \quad (iii) \\
x_j + \sum_{(ij) \in T'^k} z_{ij}^k \geq 1 & \forall j \in IR \cup BR, k \in \mathcal{C}' \quad (iv) \\
x_i + x_j - z_{ij}^k \geq 0 & \forall i \in IR \cup BR, k \in \mathcal{C}' \quad (v) \\
 & (ij) \in T'^k \\
x_i + x_j + z_{ij}^k \leq 2 & \forall i \in IR \cup BR, k \in \mathcal{C}' \quad (vi) \\
 & (ij) \in T'^k \\
\sum_{(jh) \in S'^k} y_{jh}^k - z_{ih}^k \geq 0 & \forall j \in IR, k \in \mathcal{C}' \quad (vii) \\
 & (ih) \in T'^k \\
\sum_{i \in IR \cup BR} x_i \geq 2 & \forall i \in IR \cup BR \quad (viii) \\
w_{gh}^l \geq y_{ij}^k & \forall i \in BR, j \in IR \cup BR, \\
 & k \in \mathcal{C}', g, h \in FC^l \quad (ix) \\
\sum_{(ij) \in P^l} y_{ij}^l \geq 1 & \forall i \in BR, l \in \mathcal{FC} \quad (x) \\
x_i, y_{ij}^k, z_{ij}^k, w_{gh}^l \in \{0,1\} & \forall i, j \in V, k \in \mathcal{C}', l \in FC
\end{cases}
\tag{1}
$$

As we have seen, the iBGP overlay can be crafted up from the logical topology and a snapshot/forecast of the Internet prefixes information, so for practical purposes, it is an input for the traffic engineering stage. For this stage we use equations shown in (1). The variables x_i represent the routers, and $x_i = 1$ wether router i is an RR, and 0 otherwise. Variables y_{ij} represent the connection of ASBRs to other routers, variables z_{ij} the connection between any pair of routers, and variables w_{ij} are introduced to represent the nodes belonging to the best alternative path in case of failure. We also have a set of known Internet prefix classes. For any prefix class, each internal router (IR) has a preferred border router (ASBR), based on the IGP costs. The purpose is to minimize the number of RRs subject to constraints (i) to (x); constraints (i) to (vii) ensure correctness and optimality; constraints $(viii)$ to (x) ensure resilience.

B. Second Stage - Traffic Engineering

The first stage guarantees resilience against node failures (by dynamic routing protocols) as well as IGP optimality (by the optimization model used to craft it). An optimal iBGP topology provides fast recovery against node failures (IGP convergence times), at the cost of unavoidable fluctuations in traffic distribution among routers. Links failures were intentionally left aside of the iBGP overlay to be protected during the traffic engineering of the tunnels. So it is at this stage where we must provide resilience against links failures, but also guarantee no congestion of links in the logical network, while keeping good end-to-end delay to all users.

The data-set necessary to determine an instance for this problem comprises the following objects:

- The data layer $G = (V, E)$, where V represents the set of routers and E the set of connections among them (aka logical links).
- The lengths or delays for logical links, i.e., $L : E \to \mathbb{R}^+$.
- The capacity of logical links, i.e., $C : E \to \mathbb{R}^+$.
- The demands matrix after any node $v \in V$ failure, i.e., $D_v : V \times V \to \mathbb{R}^+$. For sake of simplicity we assume symmetry for demands, so $D_w(u, v) = D_w(v, u)$ for any u, v, w in V. Observe that for consistency: $D_v(u, v) = 0$ for any u, v in V. Let D be $D(u, v) = \max_{w \in V}\{d_w(u, v)\}$.
- The limit of delay for each tunnel: $MD : V \times V \to \mathbb{R}^+$, that is defined for those (u, v) such that $D(u, v) > 0$.
- The logical-to-physical dependence, which can be simply expressed by a boolean function $pd : E \times E \to \{0, 1\}$ that indicates whether or not any two logical links share a common physical one.

The Label Switched Paths or MPLS tunnels necessary to move traffic over the network are determined by the union of D_v matrices. If there is a $D_w(u, v) > 0$ for some w in V, that is: if $D(u, v) > 0$, then a tunnel must be provided between u and v so that traffic can traverse the network. Tunnels then are implicitly determined by the input data-set. The problem to be solved here is how to craft a scheme of primary and secondary path for those tunnels.

Consider a directed graph $G' = (V, E')$ equivalent to the undirected graph $G = (V, E)$, where all edges are duplicated

to include both directions. A possible set of control variables to model the traffic engineering problem consists of:

- Those variables that determine what path is going to be followed either by the primary or the secondary path over the logical network. Let $x_{ij}^{p,uv}$ be the boolean variable that indicates whether the logical link ij is going to be used as a hop within the primary path from u to v, while $x_{ij}^{s,uv}$ are the homologous for the secondary path;
- A set of auxiliary boolean variables $y_{ij,rs}^{uv}$ that indicate if the logical link ij is going to backup traffic from u to v after a failure in link rs. The previous happens as a consequence of using ij as a part of the secondary path for the tunnel uv and using rs in the primary path.

Three blocks of constraints are to be added to the problem to achieve consistency. The following block forces the construction of logically independent primary and secondary paths for each tunnel. The expression $E^+(u)$ in (2) alludes to the set of nodes v in V such that there is an edge uv in E. Conversely, $E^-(u)$ is the set of nodes v such that vu is in E.

$$\begin{cases} \sum_{j\in E^+(u)} x_{uj}^{p,uv}=1 & \forall u\in V, D(u,v)>0 \quad (i) \\ \sum_{j\in E^+(u)} x_{uj}^{s,uv}=1 & \forall u\in V, D(u,v)>0 \quad (ii) \\ \sum_{i\in E^-(j)} x_{ij}^{p,uv} - \sum_{k\in E^+(j)} x_{jk}^{p,uv}=0 & \forall j\neq u,v, D(u,v)>0 \quad (iii) \\ \sum_{i\in E^-(j)} x_{ij}^{s,uv} - \sum_{k\in E^+(j)} x_{jk}^{s,uv}=0 & \forall j\neq u,v, D(u,v)>0 \quad (iv) \\ x_{ij}^{p,uv}=x_{ji}^{p,uv} & \forall ij\in E', D(u,v)>0 \quad (v) \\ x_{ij}^{s,uv}=x_{ji}^{s,uv} & \forall ij\in E', D(u,v)>0 \quad (vi) \\ x_{ij}^{p,uv}+x_{ij}^{s,uv}\leq 1 & \forall ij\in E, D(u,v)>0 \quad (vii) \end{cases} \tag{2}$$

Equations (i) and (ii) in (2) guarantee that a unit of flow is injected through one outgoing link from u for respectively both: primary and secondary paths, of any tunnel. Variables $x_{iu}^{p,uv}$ and $x_{iu}^{s,uv}$ are dismissed so flow cannot drain backwards. Equations (iii) and (iv) are needed to preserve flow balance in any potentially intermediate node. (v) and (vi) impose both primary and secondary paths from u to v to follow the same path back and forth. Equations block (vi) seeks for logical links independence between primary and secondary paths for every tunnel. Up to this point we should get a topologically consistent pair of paths for each tunnel.

$$\begin{cases} x_{1,4}^{s,uv} + x_{1,2}^{p,uv} \leq 1, & \forall D(u,v)>0, \quad (i) \\ x_{1,4}^{s,uv} + x_{2,4}^{p,uv} \leq 1, & \forall D(u,v)>0, \quad (ii) \\ x_{1,4}^{p,uv} + x_{1,2}^{s,uv} \leq 1, & \forall D(u,v)>0, \quad (iii) \\ x_{1,4}^{p,uv} + x_{2,4}^{s,uv} \leq 1, & \forall D(u,v)>0, \quad (iv) \end{cases} \tag{3}$$

Resiliency is in general a direct consequence of logical independence for those physically-independent logical links, which are almost all of them, in this case of study (see Figure 1). Additional constraints are going to be added for extending resiliency upon exceptions, which in this example solely is `1-4` that is optically dependent of `1-2` and `2-4`; The formulation (see (3)) is quite simple for this instance because of the triangular logical-to-physical dependence mapping of Figure 1. Since resiliency is fully provided by combining (2) and (3), it remains to be seen how it is Quality of Service (QoS), which is introduced with (4).

$$\begin{cases} \sum_{ij\in E} L(ij)\cdot x_{ij}^{p,uv} \leq MD(u,v) & \forall D(u,v)>0 \quad (i) \\ \sum_{ij\in E} L(ij)\cdot x_{ij}^{s,uv} \leq MD(u,v) & \forall D(u,v)>0 \quad (ii) \\ \sum_{D(uv)>0} D(uv)(x_{ij}^{p,uv}+y_{ij,rs}^{uv}) \leq \beta\cdot C(ij) & \forall ij\neq rs\in E \quad (iii) \\ y_{ij,rs}^{uv} \geq x_{ij}^{s,uv} + x_{rs}^{p,uv} - 1 & \forall ij\neq rs\in E, D(u,v)>0 \quad (iv) \end{cases} \tag{4}$$

Those variables where $x_{ij}^{p,uv}=1$ define a path between u and v as a result of (2). Thus, equations blocks (i) and (ii) in (2) account for the total end-to-end delay for either the primary or the secondary path, which must comply with delay limits for respective tunnels. According on the definition of $x_{ij}^{p,uv}$ and $y_{ij,rs}^{uv}$ variables, the left hand side of (iii) merely adds up to the total traffic over ij from each tunnel uv under a rs failure scenario. The right hand side on (iii) sets an upper limit for that traffic that is proportional to ij link's capacity. That limit is bonded with the objective function to optimize in this problem, which is: $\min \beta$ with $\beta \geq 0$. After the optimization process, this last variable β attains the reduction ratio in link's capacity beyond which no feasible solution can be found. If that optimal $\bar{\beta}$ is greater than 1 it would mean that current traffic with those delay and resilience constraints cannot be fit in a network with current capacities. Conversely, a $\bar{\beta}$ value less or equal to 1 indicates the problem is feasible, while the inverse of $\bar{\beta}$ meassures how much greater that traffic could be before saturating the network. Finally, equations in (iv) in (4) force consistency between $x_{ij}^{p,uv}$ and $y_{ij,rs}^{uv}$ variables, since $y_{ij,rs}^{uv}$ must value to 1 when $x_{ij}^{s,uv}=1$ and $x_{rs}^{p,uv}=1$, which translates into: if ij is used by the secondary path of uv, rs by its primary and rs fails, then uv traffic will go through logical link ij.

IV. Results for the Studied Case

In this section we present the optimization results when applying the proposed model to a potential topology as shown in Figure 1. We assume there are three ASBRs which are nodes 3, 4 and 5, and the rest are internal routers. Distances and capacities are known. For example, from node 1 to node 2 the distance is 200km, and the arc capacity is 6 units.

For the case we are studying and for simplicity, we assume we have the following prefix classes:

- class A: arrives to border routers 4 and 5
- class B: arrives to border routers 3 and 5

In real world thousands of prefixes arrive from ASBRs, but for the purposes of the iBGP optimization and after the first

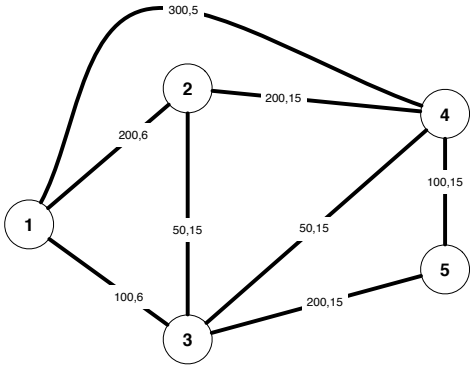


Fig. 1. Example nodes and links for the logical layer

steps of BGP path selection, the only information that counts is the combination of ASBRs those prefixes are advertised from, which in this case cannot be greater than $2^3 - 1 = 7$ classes. Note that to ensure resilience it is important that the prefixes or a superclass of them, arrive to at least two ASBRs. Then if traffic to an ASBR becomes interrupted, there is an alternative ASBR receiving those Internet prefixes.

A. *FirstStage*

The physical topology can be mapped to the IGP of the network. The resulting graph when applying ORRTD (the first stage model) as described in section III-A to the network of Figure 1 is shown in Figure 2. Even though the topology is very simple, for the purpose of resiliency two RRs are needed.

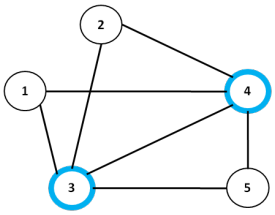


Fig. 2. Optimal RRs (in blue) and BGP sessions

B. *Second Stage*

We will present the results when optimizing the use of channels according to their capacity. We take once again the topology from Figure 1, where nodes 3, 4 and 5 are the ASBRs, and we have two prefix classes A and B, as described previously. We assume all nodes have a demand of 3 for prefix class A, and a demand of 2 for prefix class B. ASBRs have those same demands, although in some situations, they can solve the demands on their own. We take for each link the capacities shown in Figure 1. In Table I we present the demand matrix in the nominal case, i.e., when there is no failure.

Router 5 is no used by the rest of the routers, and it can manage to solve traffic to A and B. If router 3 adjacencies fail, the new demand matrix is shown in Table II: router 5 receives the traffic as it is the only way to reach B.

TABLE I
DEMAND MATRIX - NOMINAL CASE

	1	**2**	**3**	**4**	**5**
1	0	0	2	3	0
2	0	0	2	3	0
3	2	2	0	5	0
4	3	3	5	0	0
5	0	0	0	0	0

TABLE II
DEMAND MATRIX WHEN NODE 3 ADJACENCIES FAIL

	1	**2**	**3**	**4**	**5**
1	0	0	0	3	2
2	0	0	0	3	2
3	0	0	0	3	2
4	3	3	3	0	2
5	2	2	2	2	0

In Table III we show a similar case, when router 4 adjacencies fail. Finally, when router 5 adjacencies fail, traffic to 3 and 4 is generated as shown in Table IV. The resulting optimal primary and secondary tunnels are shown in Figure 3 and 4 respectively.

TABLE III
DEMAND MATRIX WHEN NODE 4 ADJACENCIES FAIL

	1	**2**	**3**	**4**	**5**
1	0	0	2	0	3
2	0	0	2	0	3
3	2	2	0	2	3
4	0	0	2	0	3
5	3	3	3	3	0

TABLE IV
DEMAND MATRIX WHEN NODE 5 ADJACENCIES FAIL

	1	**2**	**3**	**4**	**5**
1	0	0	2	3	0
2	0	0	2	3	0
3	2	2	0	5	2
4	3	3	5	0	3
5	0	0	2	3	0

This configuration satisfies all QoS constraints and resiliency. Besides, it allows the greatest demand growth, which is 20%. This number is given by link `1-3`, as traffic is reordered in the case that link `1-2` fails, as shown in Figure 5. In this case, traffic over link `1-3` goes through tunnels `1-3` and `1-4`, resulting in a value of 6. So it can be seen that with this solution no edge becomes congested, in any of the fail scenarios.

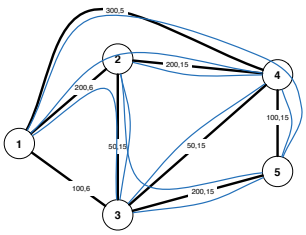


Fig. 3. Primary tunnels

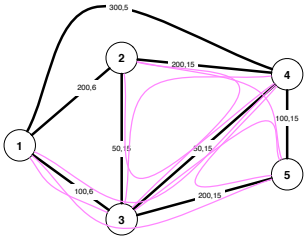


Fig. 4. Secondary tunnels

We can then perform a numerical sensitivity analysis. Suppose we then want to re-optimize after obtaining the initial solution, increasing in 1 unit each link capacity separately, e.g., first increase link 1-2 capacity in 1 unit and apply the optimization, then increase link 1-3 capacity in 1 unit, etc. Then a change in the optimal solution may indicate an improvement in the global network capacity. In the case we are studying, the only worthwhile increment is that of link `1-3`, which allows a growth of 25 % in the demand, instead of the previous obtained value of 20%. Note that the combined capacity of the network is 96; increasing link `1-3` capacity in 1 unit represents approximately 1% of the total capacity. But

this results in a better performance, as the obtained increase for the maximum network capacity is 5%.

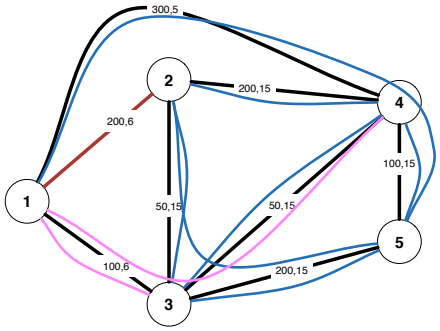


Fig. 5. Primary and secondary tunnels after link 1-2 fails

V. Conclusion

In this article we focus on optimization of both control and forwarding plane in an Internet backbone network. We propose a novel approach which relies on optimization in two stages: the first one finds the optimal and yet resilient choice after on node failure for route reflectors and its BGP connections, and the second one applies traffic engineering to make the best possible use of the tunnels taking into account traffic demands and guarantying resilience for link failures. The proposed model for the tunnels optimization is always feasible, as it decreases the demands until fitting the tunnels. At the same time the problem formulation easily allows to perform a sensitivity analysis by making input data perturbations or by checking values for dual variables of constraints in (4) (ii), allowing to analyze the impact of individual capacity changes in the maximum network capacity. Complementarily, active constraints in equation group (4) (ii) identify those link failures that stress capacities the most over the surviving network. All those features combined constitute a valuable tool for backbone operators, both to assign resources in an optimal way as well as to determine their growth strategy.

This team is currently working upon applying this technique for a real-world ISP of South America.

References

[1] Y. Rekhter and T. Li,"A Border Gateway Protocol 4 (BGP-4)," RFC1771 Obsoleted by RFC 4271, 1995.

[2] T. Bates, E. Chen, and R. Chandra, "BGP route reflection: an alternative to full mesh internal bgp (iBGP)," RFC4456 (Draft Standard), 2006.

[3] A. Flavel,"BGP, not as easy As 1-2-3," Ph.D. Dissertation, 2009, University of Adelaide, Australia.

[4] L. Xiao, J. Wang, and K. Nahrstedt,"Reliability-aware iBGP route reflection topology design," Proceedings - 11th IEEE International Conference on Network Protocols, ICNP, January 2003, pp.180–189.

[5] J.H. Park. ,"Understanding the impact of internal BGP route reflection," PhD thesis, University of California, 2011.

[6] S. Vissicchio, L. Cittadini, L. Vanbever, and O. Bonaventure,"iBGP deceptions: more sessions, fewer routes," in INFOCOM, IEEE, 2012.

[7] T. G. Griffin and G. Wilfong,"On the correctness of iBGP configuration," in SIGCOMM '02 Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 17–29, New York, NY, USA, 2002. ACM.

[8] M. Vutukuru, P. Valiant and S. Kopparty and H. Balakrishnan,"How to construct a correct and scalable iBGP configuration," in Proceedings of the 25th IEEE International Conference on Computer Communications. INFOCOM, 2006, pp. 1-12.

[9] F. Zhao, X. Lu, P. Zhu and J. Zhao,"BGPSepD: an improved algorithm for constructing correct and scalable iBGP configurations based on vertexes degree," in Proceedings of the second International. Conference on High Performance Computing and Communications., Springer-Verlag, 2006, pp. 406-415.

[10] R. Zhang and M. Bartell,"BGP design and implementation," Cisco Press, Indianapolis, 2003, pp. 264–266.

[11] M. O. Buob, S. Uhlig and M. Meulle, "Designing Optimal iBGP Route-Reflection Topologies," in Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet, pp. 542–553, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] C. Mayr, E. Grampín, and C. Risso,"Optimal route reflection topology design," in 10th Latinamerican Networking Conference (LANC '18). ACM, New York, NY, USA, pp. 65-72, 2018.

[13] C. Mayr, E. Grampín, and C. Risso,"Optimal route reflection topology design," *Technical Report*, https://www.fing.edu.uy/~crisso/Optimal_Route_Reflection_Topology_Design.pdf, 2018.

[14] C. Mayr, E. Grampín, and C. Risso,"A Combinatorial Optimization Framework for the Design of Resilient iBGP Overlays" , in 15th International Conference on the Design of Reliable Communication Networks (DRCN'19), pp. 6–10, 2019. DOI=http://dx.doi.org/10.1109/DRCN.2019.8713744

[15] C. Mayr, C. Risso and E. Grampín,"Designing an Optimal and Resilient iBGP Overlay with extended ORRTD" , *Technical Report*, https://www.fing.edu.uy/~crisso/LOD2019paper76.pdf, 2019.

[16] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC3031 (Draft Standard), 2001.

[17] D.O. Awduche, "BGP/MPLS IP Virtual Private Networks (VPNs)," rfc4364 (Proposed Standard), Updated by RFCs 4577, 4684, 5462, https://www.rfc-editor.org/rfc/rfc4364.txt, 2006.

[18] D. Awduche, J. Malcolmm J. Agogbua, M. O'Dell and J. McManus, "Requirements for Traffic Engineering Over MPLS," RFC2702 (Informational Standard), 1999.

[19] A. Farrel and J.-P. Vasseur and J. Ash," A Path Computation Element (PCE)-Based Architecture", rfc4655 (Informational Standard), https://www.rfc-editor.org/rfc/rfc4655.txt, 2006.

[20] Risso, C., Nesmachnow, S. and Robledo, F," Metaheuristic approaches for IP/MPLS network design", in International Transactions in Operational Research, 25(2), 599-625. 2018 doi:10.1111/itor.12418

[21] D. O. Awduche, "MPLS and Traffic Engineering in IP Networks," in Comm. Mag., 37(12), pp 42–47, 1999. doi:10.1109/35.809383

[22] E. Grampin and J. Serrat," Cooperation of control and management plane for provisioning in MPLS networks", in 9th IFIP/IEEE International Symposium on Integrated Network Management, pp 281-294, 2005. doi:10.1109/INM.2005.1440798

[23] A. Mereu and D. Cherubini and A. Fanni and A. Frangioni," Primary and backup paths optimal design for traffic engineering in hybrid IGP/MPLS networks", in 7th International Workshop on Design of Reliable Communication Networks, pp 273-280, 2009. doi:10.1109/DRCN.2009.5339995

[24] F. Skivee and S. Balon and G. Leduc," A scalable heuristic for hybrid IGP/MPLS traffic engineering - Case study on an operational network", in 14th IEEE International Conference on Networks, pp 1-6, 2006. doi:10.1109/ICON.2006.302621

Chapter 6

Scalable iBGP and IP/MPLS combined resilient transit backbone design

In this chapter the theoretical foundation for studying the BGP overlay is presented, a model is proposed for optimizing it even in the case of BGP adjacencies failure, and the concept of Internet *prefixes classes* is developed. Those prefixes classes are calculated for a real world transit Internet Provider, showing how the complexity of millions of BGP prefix announcements can be reduced to just less than thirty prefixes classes for the ISP case considered. Besides, a two stage model is applied to that provider network, achieving a resilient and yet optimal design. The results of a coordinated BGP and IP/MPLS scenario are also compared to those obtained by calculating LDP, showing that Traffic Engineering with routing allows 40% slack vs capacity deficits of 110% in LDP.

Highlights

Scalable iBGP and IP/MPLS combined resilient transit backbone design

Cristina Mayr, Claudio Risso, Eduardo Grampín

- We group 800 thousand prefixes advertised by ASBRs to iBGP into 27 prefix classes
- We built a correct and scalable iBGP overlay for an international ISP transit network
- We perform a complete multi-layer optimization process for this network
- Traffic Engineering turn capacity deficits up to 110% into a 40% slack scenario

Scalable iBGP and IP/MPLS combined resilient transit backbone design

Cristina Mayr, Claudio Risso, Eduardo Grampín

Abstract

The design of ISP backbones with Quality of Service (QoS) restrictions is an established discipline which has been applied for decades using the technologies available at each time. The use of IP/MPLS has been the dominant forwarding technology for the last two decades, and although new proposals for Traffic Engineering have emerged based on the separation of the control plane and the forwarding plane, such as Software Defined Networking (SDN), together with the rise of Network Function Virtualization (NFV), IP/MPLS technology is still widely used. On the other hand, a transit ISP uses BGP as a fundamental tool to determine the routing of IP traffic, both for the exchange of reachability information with other Autonomous Systems (external BGP - eBGP), and to disseminate the information into the AS (internal BGP - iBGP), allowing internal AS routers to choose the best ASBR for each IP prefix. Both technologies coexist on the backbone of the ISP, built on fiber optic links, often trans-continental, which adds significant dimensions to the costs of the IGP and in terms of delay. In this work we rely on our ORRTD model to obtain an optimal iBGP overlay, and we combine this problem with the optimization of IP/MPLS Traffic Engineering, both for nominal cases and for failure cases, using BGP updates, traffic and real topology data from a transit ISP with Points of Presence (POPs) in four american countries. Likewise, we managed to scale the problem based on a significant reduction of several orders of magnitude of the problem of BGP updates, using the concept of prefix classes. A review of previous works and a detailed study of the multi-layer optimization problems necessary to arrive at a joint optimization is presented. Moreover, the sensitivity of the solutions

Email addresses: mayr@fing.edu.uy (Cristina Mayr), crisso@fing.edu.uy (Claudio Risso), grampin@fing.edu.uy (Eduardo Grampín)

Preprint submitted to Computer Networks *October 2, 2019*

is studied, while proposing small corrections to the network topology that can achieve significant improvements in the quality of service.

Keywords: Internet Routing, BGP, Route Reflection, Network Design, Combinatorial Optimization, BGP resilience, MPLS, Demand Matrix, MPLS tunnels, Network Resilience, Internet Prefix, Prefixes Classes

---

1. Introduction

IP/MPLS backbone Traffic Engineering (TE) is a well established subject among networking academia and practitioners [1, 2]. MPLS adds TE capabilities to pure IP, both in a dynamic, decentralized way, using Constrained Shortest Path First (CSPF) algorithms and LDP signalling, and in a logically centralized manner, using some external TE engine (for example, the Path Computation Element Architecture -PCE), and RSVP-TE signalling, which also provides Fast Rerouting (FRR) features. The emergence of Control Plane programmability under the Software Defined Networking (SDN) paradigm, Data Plane programmability (using for example the P4 programming language) and Network Function Virtualization (NFV) which permits to build service chains (i.e. concatenate a load balancer with firewalling services), in conjunction with traffic monitoring techniques, usually based on Machine Learning approaches, offer a diversified toolset to tackle network TE challenges. Also, the re-emergence of source routing ideas (now re-branded as Segment Routing), adds a powerful tool for TE, possibly replacing or combined with MPLS [3]. Nevertheless, the underlying optimization problem is yet to be solved for every particular backbone design, taking into account the Service Provider requirements, traffic demands and resiliency. Classical TE and resilience problems fall into the category of hard-to-solve network problems. Moreover, TE and QoS planning in an IP/MPLS network depends on how traffic comes-in or gets-out from that network, which is in turn conditioned by BGP protocol. One of the backbone design steps of our approach directly deals with this problem.

The Internet is a loosely hierarchical interconnection of Autonomous Systems, which rely on BGP for disseminating IP prefixes reachability information; BGP is a policy-routing protocol, meaning that each AS may filter announcements and modify attributes in order to implement local policies. Typical policies include prefix aggregation and de-aggregation for coarse grain TE, and advertisement filtering to comply with valley-free rules (mean-

ing that no network will carry other's traffic for free) [4]. BGP among ASes is usually referred as External BGP (eBGP), which covergence properties and scalability issues have been extensively researched, regarding both the size of the Default Free Zone (DFZ) routing table and the protocol churn (the frequency of update messages) [5].

Regarding the intra-domain, a typical Service Provider run an Interior Gateway Protocol (IGP) for reachability of internal prefixes, combined with instances of Internal BGP (iBGP) in order to disseminate external reachability information inside the AS, giving internal routers the ability to correctly choose the Autonomous System Border Router (ASBR) for any given prefix. In order to make sure that internal transport of BGP prefixes info is loop-free (control plane), and internal routing is coherent (loop-free and deflection-free data plane forwarding), the following iBGP advertisement rules must be observed: 1) prefixes learnt from an eBGP neighbor should be re-advertised to iBGP neighbors, and vice versa; and 2) prefixes learnt from an iBGP neighbor cannot be re-advertised to another iBGP neighbor. Whatever the rule applied, BGP routers must pre-process prefixes attributes prior to relaying them, what biases attributes with particulars of its own placement within the network; nevertheless, in the intra-domain scope attributes are seldom changed [6].

A mechanism to comply with the aforementioned advertisement rules, assuring the correct dissemination of reachability information, is to implement a full-mesh of BGP sessions among all routers within the AS; this solution introduces severe scalability issues: on the one hand, a quadratic number of BGP sessions must be maintained, and on the other hand, the routing state (i.e. the size of iBGP Rib-In routing table) can be order n larger than the number of best routes (i.e., for T best routes, Rib-In size can reach up to $n \times T$ entries), imposing large CPU and memory requirements to every router in the AS. Also, another aspect of iBGP scalability that shall be considered is the number of BGP messages generated intra-domain by external BGP updates (churn). The widely adopted alternative to full-mesh are Route Reflectors (RRs), which explicitly infringe rule #2, admitting the relying of internally learnt prefixes to other internal BGP peers, named clients. Choosing RRs and BGP sessions is referred to as the *iBGP overlay design problem*, which is discussed in next section. Correct and scalable solutions to this problem depends not only on network topology, but also on the prefix amount and diversity. Another design step tackled into the article aims upon this second problem.

From an abstract point-of-view a modern telecommunications network is comprised of a stack of physical resources, technological infrastructure and network protocols. A typical IP/MPLS transit backbone comprises these layers: i) the optical fiber network also referred to as the physical layer; ii) the IP data network or logical layer is compounded of routers and links among them (implemented over the physical layer); iii) the IP/MPLS tunnels (implemented over the logical layer) for traffic forwarding; iv) the set of traffic engineered paths assigned to each one of those tunnels across the data network (e.g. primary and secondary paths); moreover, we need to consider v) the iBGP overlay, i.e., the set of BGP sessions among routers and their role (RR or client), whose topology, combined with the prefixes particulars of the whole set of eBGP messages, affects how traffic traverses the AS. An *overlay network* is that where connections between nodes are in fact supported as services of an underlying network. Therefore, the network of a modern AS is in fact an entwined stack of overlays, whose global design is intractable as a whole.

Our main contribution, building over previous work, undertakes the design problem for a real-world based case studio. Our approach follows a novel sequence of overlay optimization sub-problems, starting from the processing of several millions of eBGP updates to curtail them to some dozens sets that capture all the important routing information as families of *routing classes*. These classes, combined with the logical network topology are subsequently used as supplies for designing a minimal and consistent iBGP overlay, using our framework Optimal Route Reflector Topology Design (ORRTD) [7, 8, 9, 10]. This work elaborates about a variant of the framework that grants consistency between a pure IP routing protocol and the existence of MPLS forwarding. Details of this variant and results for the real-world scenario are part of the contribution of this article, which shows that popular heuristic approaches to tackle de iBGP overlay are not optimal, or directly unfeasible, when applied to this experimental instance. Another contribution of the work comes from applying a novel traffic-engineering model to optimize an overlay of MPLS paths in accordance with IP routing demand particulars, links capacities, resiliency requirements and the existence of limits for propagation delays between zones of the network. It is also shown how results can be used to determine where capacities should be expanded to get the most significative returns in the overall performance, which is our last remarkable contribution.

The remaining of this article is organized as follows. Section 2 intro-

duces the details for the most important technologies elaborated in this work (i.e. BGP and MPLS); It also presents a summary of the related literature and state-of-the-art. Section 3 presents the specifics for the ORRTD variant and formulation of an optimal and resilient iBGP overlay that is to be coordinated with another overlay of MPLS tunnels. The section also documents the algorithm used to generate the classes of prefixes and it finally elaborates about how to design a BGP consistent traffic-engineering based overlay of MPLS tunnels, which complements the previous resilience while seamlessly serves its forwarding purposes. Section 4 introduces the details of the particular application case and presents the main results obtained for it. Finally, Section 5 presents the main conclusions regarding the problem and its experimental results, and presents lines of future work.

2. Background Technologies

This work deals with a transit ISP backbone design, using as underlying technologies IP/MPLS over optical transport, and an iBGP overlay for traffic steering.

2.1. iBGP Overlay

BGP [11] is the de-facto standard to exchange reachability information among neighbor ASes in the Internet. Unlike IGPs such as RIP, OSPF, and IS-IS, which automatically determine network adjacencies with other routers directly connected, and share routing information thereafter, BGP adjacencies are administratively set point-to-point, using BGP sessions over TCP, which rely over basic IP connectivity. BGP sessions between routers from different ASes are usually established over direct links, but in the intra-domain scope the underlying IP connectivity is sustained by an IGP protocol. Therefore, we refer to the set of BGP routers and their adjacencies as an overlay over the IP network, which topologies seldom match.

BGP is a *path-vector* policy routing protocol that ensures loop-free paths throughout the network, using the AS_PATH attribute, which records the route taken by a BGP update. Loops can be easily avoided: routers which find its own AS number in the AS_PATH sequence of incoming BGP updates must discard them.

AS_PATH is one of several BGP attributes, which can either be: i) mandatory (must be attached to every prefix entry), ii) discretionary (must be recognized by every router but not necessarily appears in every entry)

or iii) optional (not necessarily supported by all implementations). Another classification dimension includes iv) transitive (must be propagated to other ASes), or v) non-transitive attributes (could be propagated, but only within each AS). Attributes are used for consistency (e.g. the AS_PATH) or to determine the most suitable route to a destination when multiple paths are available. Some of these attributes are:

- LOCAL_PREF: Local Preference is a well-known discretionary attribute used to administratively indicate paths preference. When there are several paths to a destination network, and the path to be chosen by default is other than that desired by network administrators, they can manually set a higher local-pref to force other routers to use that path. The preferred path is the one having the highest local-pref. LOCAL_PREF is non-transitive since it only has significance within an administrative domain (an AS), but it is internally relayed by iBGP.

- MED: Multi Exit Discriminator is a discretionary non-transitive attribute, also manually set, but in this case by administrators of a neighbor AS to suggest the preferred entry path to its network, when more than one entry point is available for a certain prefix. Although is received from an external AS, MED is non-transitive, because it is only intended to be processed by the receiver.

- AS_PATH: Autonomous System Path is a mandatory attribute that describes the path (list of ASes) traversed by that update in its way towards the receiver. When a route (i.e. a prefix) is advertised through an AS, that AS number is added to the sequenced list or AS path. Its purpose is to avoid loops of ASes, as it was previously explained.

- NEXT_HOP is a mandatory attribute that specifies the IP address that must be taken as a reference (i.e. the effective gateway) to reach the destination (next AS). A subtle but cardinal characteristic of BGP is that it relies upon the existence of an operative intra-AS-network routing protocol. So, by means of BGP a router can get to know directions towards every IP address in the world, except for those belonging to its own AS. A typical example for the usefulness of this attribute is that where an ASBR relays an eBGP update to other routers in its AS. Prior to do that, the ASBR replaces the original NEXT_HOP address (the one of its peer) with the IP address of any of its own interfaces,

which are part of the IGP, and therefore they can be effectively taken as gateways by the remaining nodes.

- WEIGHT is an optional (Cisco proprietary) path attribute used to influence a local router's path-selection for some outbound routes. The active path with the highest weight value always wins. It is a way to forcing a specific router to use certain BGP update whenever it is available. It resembles a static route, with the difference that in this case, the route can be dynamically withdrawn from the *routing table* as a consequence of a dynamic routing protocol.

- ORIGIN: this mandatory attribute informs all ASes about the mechanisms by which the prefix was introduced into BGP, i.e., by EGP, IGP or incomplete redistributed.

Most ISPs deploy link-state IGPs, such as OSPF and IS-IS, which rely on a topological database, dynamicaly updated by the exchange of Link State Advertisements (LSAs) among routers, to determine the shortest path between destination pairs. The total cost of a shortest active path between two nodes in an AS is referred to as the *IGP metric* between those nodes. Meanwhile, finding routes for external prefixes relies upon BGP, which usually must consider multiple alternative paths, and therefore needs a mechanism to determine which one is going to be selected. The following hierarchical tie-break steps synthesize the so-called *BGP best path selection* process:

---

i. Prefer highest weight (local to router)
ii. Prefer highest local preference (global within the AS)
iii. Prefer route originated by the local router
iv. Prefer shortest AS path
v. Prefer lowest origin code (IGP < EGP < incomplete)
vi. Prefer lowest MED (from other AS)
vii. Prefer eBGP path over iBGP path
viii. Prefer the path through the closest/lower cost IGP neighbor
ix. Prefer oldest route for eBGP paths
x. Prefer the path with the lowest neighbor BGP router ID

Keep in mind that it is not uncommon that many prefixes keep tied after

rule *vii*, and consequently rule *viii* must be applied, settling a dependence between BGP and the IGP. Indeed, the IGP metric will lead to choose *next-hop-best* specific and likely different among routers. Steps *ix* and *x* are seldom reached, as usually the steps up to *viii* are tie-breakers.

Remember that two rules must be observed to make sure that internal transport of BGP prefixes info is loop-free (control plane), and internal routing is coherent (loop-free and deflection-free data plane forwarding): 1) prefixes learnt from an eBGP neighbor should be re-advertised to iBGP neighbors, and vice versa; and 2) prefixes learnt from an iBGP neighbor cannot be re-advertised to another iBGP neighbor, and a simple mechanism to comply with such rules is to implement a full-mesh of BGP sessions among all routers within the AS. Having considered the scalability problems introduced by this practice, a widely accepted alternative is implementing Route Reflectors (RRs) [12]. In this case, one or more routers within the AS are designated as RRs and they are allowed to re-announce routes learned from an internal router to other internal peers, while the rest of them act as RRs clients. An advantage of route reflection is that the number of iBGP sessions scales linearly with the number of routers. It is important to remark that RRs re-advertise only best routes after running their own decision process, as any BGP router. As a consequence, re-advertisements are biased by the placement of RRs within the AS's topology, as prefixes selection considers IGP metric during the BGP path-selection. Main drawbacks of reflection are reliability and biasing, since the outcome after step *viii* for a route reflector would probably be different from the one a client router would have chosen if it had had the whole information to make the decision by its own.

Forwarding consistency in a pure IP network relies upon the routing optimality. In such networks, forwarding actions are taken hop-by-hop, and they involve all routers along the path. Suppose some router `A` sends a packet with destination to some router within the AS, namely `D`. Up from its IGP topology database, `A` finds out that the shortest path to `D` is `A-B-C-D`, so it sends the packet to `B`. Once that packet is in `B`'s hands, the last one must relay the packet to `C`, or towards another router with the same cost to the destination `D`. The *optimality principle* guarantees that the optimal cost between `A` and `D` cannot be different from the sum of the optimal costs from `A` to `B` and that from `B` to `D`. This simple fact prevents from looping.

When the destination is an IP address of another AS, however, that loop-free guarantee might not hold for an AS that implements its iBGP overlay with Route Reflectors. To avoid a single point of failure, redundant route

reflectors are used as a rule. Consequently, an iBGP overlay should have at least two RRs, namely `RR1` and `RR2`. Suppose some internal router `A` needs to send an IP packet to a foreign IP destination, and suppose that `A` is connected to `RR1`, for whom, the optimal border for that destination is `BR1`, so that is the optimal for `A` as well. Then, `A` uses its IGP to determine that `B` is in the optimal path towards `BR1`, and then it sends the packet to `B`. The router `B` on the other hand is a client of `RR2`, and unlike the other, this reflector finds border `BR2` more suitable and teaches that to `B`. If `A` happens to be in the shortest path from `B` to `RR2`, we would be in the presence of an IP loop.

The previous situation would not have happened for an overlay with a full-mesh of sessions, because of the optimality principle. An iBGP overlay is *fm-optimal*, when every router in that network ends up choosing the same route it would have chosen under a full-mesh of sessions [13]. Such optimality not only guarantees a better quality in routing decisions but also in forwarding consistency. Deciding what routers will be RRs and their adjacencies is a complex and delicate task, as it has been discovered that persistent BGP route oscillations and stable routing loops can occur when BGP route reflection is deployed ([14, 15, 16]).

The *iBGP overlay design problem* consists in deciding which routers are to be route reflectors, and which sessions are to be established between remaining routers (i.e. clients) and those RRs, in order to obtain a *fm-optimal* iBGP overlay. Moreover, the most important question is: what is the fm-optimal iBGP overlay with the minimum number of RRs and sessions? The problem is of notorious academic and industrial relevance.

Certainly, faced to the iBGP overlay design problem, an operator would not only take into account the correctness/scalability trade-off, but principally needs practical guidelines. Many algorithms have been presented in the literature, mainly focused in the reduction of iBGP peering sessions. A first family of algorithms are based in the construction of a Graph Separator for a given IGP topology, in order to realise that for every router P and every egress router E in the topology, either P and E have an iBGP session, or P is the client of a route reflector on the shortest path between P and E. The algorithm recursively build a hierarchy or Route Reflectors. The original proposal, BGPSep [17], has been improved by it variants BGPSep_D [18] and BGPSep_S [19].

Buob et al. [20], following the fm-optimality idea described before, built an algorithm where the route chosen for every egress point is the best route that should have been chosen by a full mesh configuration. The algorithm

builds an iBGP topology robust to IGP link failures and router maintenance, claiming that even after a single link failure or the removal of a router, the topology remains fm-optimal. As mentioned before, the scalability objective is maintained: the iBGP topology comprise as few iBGP sessions as possible.

In [21], Flavel and Roughan explore routing algebras and propose an alteration to the BGP decision process that evaluates the length of the *cluster-list* before comparing IGP weights. The cluster-list attribute in iBGP contains the identifiers of the clusters the route has traversed (similar to the AS-path attribute in eBGP). It is primarily used to avoid loops but it implicitly contains the number of iBGP hops, and therefore it can be used as a new decision step. Such a variant of iBGP is proved to always converge, but requires changes to the BGP implementation on routers, which may prove difficult to achieve.

From long ago, BGP practitioners have been using heuristics to establish hierarchical route reflection. In [12], Bates et al. state some recommendations for iBGP route reflection topologies. They advise to configure one or multiple RRs per Point of Presence (PoP) in the network, where all the routers in a PoP are clients of the RRs in this PoP. In addition, the authors require a full-mesh of iBGP sessions between the RRs, and they also recommend the configuration of a full-mesh of iBGP sessions between all the routers in a PoP. Large Service Provider networks may set up hierarchical, multi-level route reflection topologies. Routers that are clients of RRs at the top-level may on the other hand be RRs for routers at lower levels. In [22], Zhang and Bartell provide recommendations for the design of such hierarchical iBGP topologies, advising that the RRs at the top-level must be fully meshed, but, on the contrary, this is not required for RRs at lower levels.

The IETF has been concerned in practical iBGP operational matters, such as increasing the path diversity in iBGP, aiming to reduce the convergence time. Raszuk et al. [23] propose to increase path diversity within an AS by modifying the best path selection in RRs so that different RRs will advertise different paths to client routers. Another proposal is adding a best external option in BGP [24], by which a border BGP router can propagate more than one best external path to iBGP neighbors inside an AS. This can increase the number of paths observed by iBGP routers and decrease the number of hidden paths. Yet another proposal by Walton et al. [25] advise to allow any BGP router to propagate more than a single best path to increase the overall path diversity. The costs and benefits of BGP add-paths have been carefully analysed in [26]. An alternative already implemented by

major network vendors is BGP Optimal Route Reflection (BGP-ORR) [27], which claims to deploy a fm-optimal solution, based on the centralized approaches originaly presented by Feamster el al.[28] and Oprescu et.al. [29].

In our previous work we undertook the optimal iBGP overlay problem for an IP network [7, 8, 9]. The first approach [7] introduced a basic combinatorial optimization formulation of the problem, where only Internal Routers (IR) could be electable as RRs. That formulation did not guarantee fm-optimality after some failure in a node or link. Later on, the model was extended [8] to craft optimal and yet resilient iBGP overlays, capable of preserving fm-optimality even after any single node or link failure. In that work, we also proved that the basic problem is $\mathcal{NP}$-Hard. Resiliency apart, the second version, however, still limited RRs election to IRs. A third variant [9] of the same basic framework relaxed that constraint, allowing any router (IRs and ASBRs) to be designated as a reflector. Because of the flexibility of the basic model to be adapted to all those variants, we decided to name it Optimal Route Reflector Topology Design (ORRTD).

The scope of the aforementioned antecedents is a pure IP network. Some potential issues which may appear in the presence of failures can be tackled when MPLS forwarding is available, as we detail in the following section. Nevertheless, the basic goal of realizing the best possible IP routing holds, and therefore, the fourth variant of the ORRTD problem tackles the case where routing decisions are those of an IP network, while the forwarding process within an AS relies upon MPLS forwarding. A blueprint model to solve that problem is published in [10]. In the present article, we elaborate about details of the formulation and show how the model was applied to solve a real-world application case of a South-american Internet Service Provider.

2.2. MPLS

MPLS has been adopted by ISPs as the de-facto forwarding technology in the last couple of decades, and continues to be predominant, despite the emergence of the Software Defined Networking (SDN) paradigm, which combined with other technologies may ease the Traffic Engineering in large network backbones; indeed, since MPLS basic forwarding operations are supported by OpenFlow [30], MPLS/SDN solutions may be deployed [31]. MPLS offers a scalable, protocol agnostic data-carrying mechanism, which transfers packets by assigning *labels* across the network through virtual circuits, what resembles legacy Frame Relay or ATM technologies. So, the underlying idea

is that forwarding decisions are based on *labels* rather than on destination IP addresses. In fact, except for the communication endpoints, MPLS switches are agnostic about the packets payloads, which could be either: IP packets, Ethernet frames, ATM cells, or even other MPLS packets, since stacks of MPLS labels usually appear, for instance, to emulate Virtual Private Networks or for implementing Fast Reroute. The standard application of this technology is entwined with the legacy IP protocol stack. Actually, many extensions to classical protocols were introduced to improve the matching between both technologies capabilities. Examples of that are OSPF-TE [32] and ISIS-TE [33], but there are many more. For example, BGP itself was extended as a mean to distribute MPLS information. These technologies are so entangled, that they are now inseparable, and the whole stack is referred to as IP/MPLS. The scheme is as follows.

Label Edge Routers (LERs) are placed at the edge of an MPLS network, acting as gateways between local and external networks, i.e. the Internet for an ISP case. Packets coming into each LER are classified (usually following some QoS criteria administratively set) and assigned to a FEC (Forwarding Equivalence Class). Each FEC is assigned a network path between origin and destination, signalled by an unidirectional LSP (Label Switch Path); LSPs are also referred to as *tunnels* or virtual cirtcuits. When there is more than one FEC between two nodes, stack of MPLS labels are used to separate packets at endpoints.

There are many alternatives for signaling LSPs. An LSP could be computed and its signaling triggered by the ingress LER, or by a consensual distributed protocol among switches, or by other external entity (e.g. the Path Computation Element - PCE [34]). Most popular mechanisms derive from dynamic protocols and rely upon classic IP connectivity. For instance, LSPs might be dynamically built and updated either following the IGP shortest paths, or other optimization criteria (e.g. link cost, bandwidth or available capacity [35]). As we see in this document, the overall performance of a centralized design far exceeds that of popular distributed signaling strategies.

Among distributed variants, LDP (Label Distribution Protocol) is the most popular [36]. LDP signals tunnels among every pair of LERs in the network, which automatically replicate the underlying IGP topology, and therefore reproduce the IGP metrics. Consequently, the path that an MPLS packet would follow is exactly the same that an IP packet would have used. Nevertheless, it is worth mentioning that an LDP based IP/MPLS network has augmented capabilities with respect to the underlying IP network, such

as the potential of seamlessly transporting any protocol (not only IP), native support to IP Virtual Private Networks, support for specialization between border and core routers (e.g., only the border routers shall deal with traffic classification - FECs), and avoiding IP addresses lookups for packet forwarding at intermediate nodes, among others. Label-based forwarding not only may favour efficient lookup implementation, but in our case, it may guarantee loop-free integrity in opposition to the hazards of a suboptimal iBGP overlay, as previously described.

An example of a non-consensual mechanism for signaling LSPs is RSVP-TE (ReSerVation Protocol with Traffic Engineering extensions) [37]. In this case, the ingress node initiates the LSP signalling towards the egress endpoint. While LDP replicates the IGP topology, RSVP-TE might be used for signaling optimized LSPs, at the cost of maintaining the state at intermediate nodes. RSVP-TE also permits implementing Fast Re-Route (FRR) options, which provide fast traffic recovery upon link or router failures for mission critical services. RSVP-TE signaling teams with IGP-TE protocols (e.g. OSPF-TE or ISIS-TE), in the sense that after getting labels for a path and reserving bandwidth along it, remaining nodes are updated about these changes by means of the IGP database.

The set of requirements for Traffic Engineering over Multiprotocol Label Switching (MPLS) was first presented in [1], and has been deeply studied afterwords, both from the management point of view as in [38] and design approaches [39, 40, 41]. A well-known automatic mechanism to accomplish TE is the Constrained Shortest Path First (CSPF) algorithm. CSPF is an efficient variant of the Dijkstra's algorithm that computes the minimal cost path between two points, whose links satisfy some particular conditions. Examples of such constraints could be such as: links having at least some capacity available; links labeled with certain codes (administrative groups); or on the contrary, links not labeled as certain groups. Observe that constraints in previous examples are always local constraints. The traffic demand matrix (the aggregated ingress to egress traffic that flows across the network) must be known in advance to be able to craft congestion free traffic-engineered tunnels.

It is worth mentioning that CSPF cannot solve global constraints, such as: find the lowest cost path whose end-to-end delay is not greater than *some milliseconds*. Precisely, the previous example, i.e., the end-to-end delay, is becoming a critical value for final users, and it should be a major concern for network designers. A step in the sequence of optimized overlay design

problems solved in this work combines: resilience, delay, congestion and costs as elements of the design of paths for the MPLS tunnels.

Before going into the problem formulation, we have to mention some details about how BGP routing and MPLS tunnels integration. Recall that step *viii* in the BGP decision process (Section 2.1) uses the IGP metric to select some routes over others. That makes sense in a pure IP network, since the IGP shortest path is the one that packets should follow. Even in an LDP signaled MPLS network, the process remains compatible, since LDP tunnels match IGP shortest paths; but that condition generally does not hold for traffic-engineered paths. The solution to that problem is quite simple indeed and consists in presenting tunnels as interfaces within the IGP table. So, as LDP tunnels signaling advances, more and more virtual interfaces appear in the IGP table eventually till the point where all routers are neighbors.

Just as any regular interface, tunnels have a cost attribute. Without manual intervention, that *virtual cost* is automatically set to the IGP cost for that path. The cost could also be administratively set, what provides a highly adjustable tuning between the BGP overlay (the IP routing) and the MPLS tunnels (traffic forwarding) interoperation. For instance, consider a tunnel between nodes `A` and `B` which is assigned with two paths, primary and secondary paths, physically independent, so the secondary path spares the primary path. Suppose that router `A` determines that `B` is the best *next-hop* for some prefixes, because of the cost of the primary path. Afterwards, a failure in some link tears down that primary path so the second path activates. Primary and secondary paths costs are most likely different, and there is a certain chance for the BGP path selection process at node `A` to switch the next-hop to a router different from `B`.

That effect is undesirable in general. First of all because the failover does not actually work and that deteriorates the traffic-engineering goals, but also because for the sake of the global stability of the Internet. A minor fault within an AS should not affect how traffic is interchanged with others, especially in the case when both paths' costs are similar among them and also regarding the optimal IGP distance.

As we see later on, our design supposes that tunnels are administratively (statically) assigned with the IGP cost of a faultless network, while constraints are added to guarantee that paths actual costs are as close as possible to that value. In other words, we are designing both layers coordinately, in such a way that IP routing is immune to internal link failures.

3. Problem Formulation - The Backbone Topology Design

A typical ISP backbone comprises several layers:

- the optical fiber network also referred to as the physical layer;
- the IP data network or logical layer that is compounded of routers and links among them (implemented over the physical layer);
- the IP/MPLS tunnels (implemented over the logical layer);
- the set of traffic engineered paths assigned to each one of those tunnels (e.g. primary and secondary paths);
- the iBGP overlay.

The design process followed in this work goes in the opposite direction to this list. The first stage is the design of an optimal and resilient iBGP overlay (Section 3.1) for all prefixes, which are grouped in equivalence classes. This premise is critical for model scalability, since a typical AS receives millions of prefixes updates from peering ASes. We consider the routes that maintain a tie after executing up to step *vii* of the BGP selection process, so prefixes selection only relies upon the IGP metric. Details about how those prefixes classes are assembled are documented in Section 3.2, so it is the process followed to determine how IP traffic is expected to be rearranged after losses of adjacencies with other ASes, which are the only kind of faults against which we are explicitly protecting the iBGP overlay (since faults in the IP and lower layers are covered by MPLS). Those demands determine what IP/MPLS tunnels are to be set. Finally, a traffic engineering optimization stage (Section 3.3) provides complementary resilience against any internal link failure. Both overlays (BGP and IP/MPLS tunnels) take the data network topology as an input. The outcome of the second model, however, not only crafts paths but it also provides valuable information to asses, how and where additional capacity should be installed. To be effective, the design of the IP/MPLS tunnels must consider logical-to-physical layers dependencies, i.e., what data-links share a common optical fiber.

3.1. iBGP Overlay - Optimal Route Reflector Topology Design (ORRTD)

In our previous work we studied the iBGP overlay design problem in pure IP networks, not only the nominal case [7], but also in the case where the

overlay must preserve optimality after any single link or router fails [8]. Furthermore, we proposed a combinatorial model to minimize the number of route reflectors that could be chosen among the IRs, and a relaxation where ASBRs can be eligible as RRs [9]. The methodology, called *Optimal Route Reflector Topology Design* (ORRTD), considers that different prefixes classes (groups of Internet prefixes) are received by ASBRs.

In order to explain how ORRTD works, consider the network as a graph, which could be that shown in Fig. 1, where the white nodes are internal routers and the rest are ASBRs.

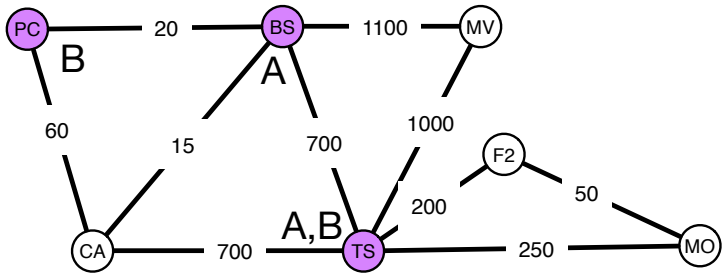


Figure 1: A logical network example

This graph is undirected, and the weight of each link is the IGP cost; in this work (since the object of study is an international backbone), this cost corresponds to the length of the link. Border router `BS` receives updates for prefixes class A, which it would relay to every other router under a full-mesh of iBGP sessions. `PC` announces a different class B, while `TS` announces both classes. The objective is to find the minimum number of RRs and the corresponding iBGP adjacencies, so that reliability is preserved, and no sub-optimal route is chosen. As we previously explained, we consider the BGP path selection process after step *vii*, and therefore ASBR selection only depends on the IGP metric. With information as in Fig. 1, an optimal router to border-router graph can be build ([7, 8, 9]) for each class of prefixes.

We follow the example in Fig. 1 focusing now in prefixes classes A. Disregarding class B in Fig. 1, the only effective border routers are `BS` and `TS`, since they are the sole advertising these prefixes to the others. As a consequence, regarding class A, `PC` is just another internal router. Being `BS` and `TS` the only candidates for next-hop, remains to be seen which of them is to be chosen by the other nodes. That can be easily determined up from the shortest

path towards both options. The result is sketched in Fig. 2 by means of links indicating the desirability between internal and border routers. Fig. 3 on the other hand sketches affinities between nodes preferences. For instance, the affinity between `MO`, `MV` and `F2` for class A indicates that if any of them would be a reflector connected to `TS`, as its clients, the other two would have full-mesh optimal routing information.

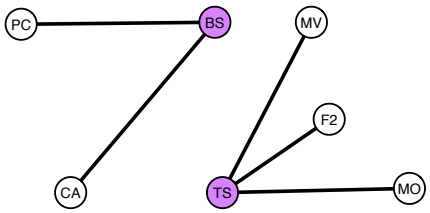


Figure 2: Router to ASBR preferences (class A)

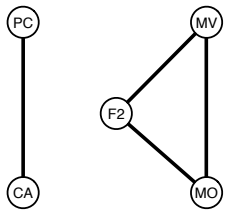


Figure 3: Router to router affinities (class A)

An equivalent example is the homologous for class B, whose result is sketched in Fig. 4 and Fig. 5.

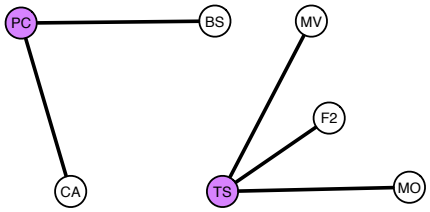


Figure 4: Router to ASBR preferences (class B)

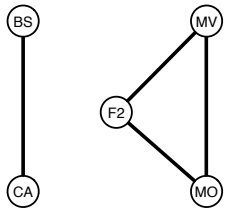


Figure 5: Router to router affinities (class B)

Variants of the previous idea allows modeling other situations. Consider now a loss of eBGP adjacencies at node `TS` that banishes in fact its condition of border router. That issue leaves `BS` as the only possible gateway for class A, and it raises `PC` to the same condition regarding class B., The simplicity of the example results in the border preference graphs being star graphs, with centers respectively at `BS` and `PC`. Affinity graphs, on the other hand, are cliques with all nodes except `BS` or `PC`. It is worth noticing that these graphs capture preference and affinities for the same classes under a contingency condition. A consistent design must consider all of them simultaneously while determining route reflectors and adjacencies, because the iBGP

overlay is unique. In other words, each class defines a preference graph and an affinity graph, and that would be enough for the nominal (non-faulty) scenario [7]. Dealing with losses of eBGP adjacencies can be easily introduced into the same model by generating *virtual classes* [8], which simply result from eliminating borders (one-by-one) to check if new preference or affinity graphs are to be introduced. With the information provided by these auxiliary graphs we can craft a combinatorial optimization model to tackle the problem. Its formulation is that presented in Equations 1. Prior to going into the whole problem, we enumerate variables and those parameters that define an instance.

Input data-sets for problem formulation in Equations 1

BR	:	set of all Autonomous System Border Routers
IR	:	set of all Internal Routers
V	:	routers $V = \{IR \cup BR\}$. It holds that $BR \cap IR = \emptyset$
E	:	set of logical links connecting those nodes V
W	:	IGP cost function $W : E \to \mathbb{R}^+$
$\mathcal{C}$	:	set of nominal prefixes classes
$\mathcal{F}$	:	classes to borders correspondence $\mathcal{F} : \mathcal{C} \to 2^{BR}$.

As we see in Section 3.2, function $\mathcal{F}$ must be injective, since classes are in fact defined up from those prefixes that are announced from the same set of border routers. Proceeding as in the example of Fig. 1, we derive other intermediate objects for the model.

Intermediate objects for the problem in Equations 1

$\{BR^k\}$	:	$BR^k \subseteq BR$ is compounded of those BR that update the class k
$\{IR^k\}$	:	$IR^k \subseteq V$ is compounded of those routers that do not update the class k, which could either be native IR of BR of other classes.
$\{S^k\}$	:	sets of router-to-border BGP preference edges. $S^k_{ij} = 1$ if and only if ij is in the ASBR-to-Router preference graph for prefix class k, with $i \in BR$, $j \in V$ and $k \in \mathcal{C}$. As in Fig. 2 for class A or as in Fig. 4 for class B.
$\{T^k\}$	:	sets of router-to-router BGP affinity edges. $T^k_{ij} = 1$ if and only if ij is in the Router-to-Router affinity graph for prefix class k, with $i, j \in V$ and $k \in \mathcal{C}$. Like the one shown in Fig. 3 for class A or in Fig. 5 for class B.
$\mathcal{FC}$	:	set of contingency fictitious classes (arise from borders adjacencies losses).
$\{P^l\}$	:	fictitious router-to-border BGP preference edges. They are like $\{S^k\}$ but for fictitious prefixes classes $l \in \mathcal{FC}$.
$\{Q^l\}$	:	fictitious router-to-router BGP affinity edges. They are like $\{T^k\}$ but for fictitious prefixes classes $l \in \mathcal{FC}$.

After all that preprocessing we can get to the definite set of parameters for Equations 1. Starting from the original graph $G = (BR \cup IR, E, W)$, and classes information $\mathcal{F} : \mathcal{C} \rightarrow 2^{BR}$, we derive: the full-set of classes (either real or virtual) $\mathcal{C}' = \mathcal{C} \cup \mathcal{FC}$; the full-set of router-to-border preferences $S' = \{S^k\} \cup \{P^l\}$; as well as the full-set of router-to-router affinities $T' = \{T^k\} \cup \{Q^l\}$. The problem formulation is then:

$$
\begin{cases}
\min \sum_{i \in V} x_i & & \\
\text{Subject to :} & & \\
\sum_{i \in V} x_i \geq 2 & \forall i \in V & (i) \\
x_i + \sum_{ij \in S'_k} y^k_{ij} \geq 1 & \forall k \in \mathcal{C}', S'_k \neq \emptyset, i \in BR^k & (ii) \\
x_i + x_j - y^k_{ij} \geq 0 & \forall k \in \mathcal{C}', j \in IR^k, ij \in S'_k & (iii) \\
x_i + x_j + y^k_{ij} \leq 2 & \forall k \in \mathcal{C}', j \in IR^k, ij \in S'_k & (iv) \\
x_j + \sum_{jh \in T'_k} z^k_{jh} + \sum_{ij \in S'_k} y^k_{ij} \geq 1 & \forall k \in \mathcal{C}', \forall j \in IR^k & (v) \\
x_i + x_j - z^k_{jh} + y^k_{ih} \geq 0 & \forall k \in \mathcal{C}', jh \in T'_k, ih \in S'_k & (vi) \\
x_i + x_j - z^k_{ij} \geq 0 & \forall ij \in T'_k, k \in \mathcal{C}' & (vii) \\
x_i + x_j + z^k_{ij} \leq 2 & \forall ij \in T'_k, k \in \mathcal{C}' & (viii) \\
x_i,\ y^k_{ij},\ z^k_{ij} \in \{0, 1\} & \forall i, j \in V, k \in \mathcal{C}' &
\end{cases}
\tag{1}
$$

Binary variables x_i correspond to the reflector condition of routers, i.e., router i is to be a reflector whether $x_i = 1$ and not otherwise. Variables y^k_{ij} indicate a necessary full-mesh optimality adjacency for a class k between $i \in BR^k$ and $j \in IR^k$; variables z^k_{ij} indicate necessary adjacencies between pairs of internal routers for the class k. Objective function in Equations 1 simply express the fact that our goal is to have the minimum possible number of route reflectors. Equations (i) guarantee that number to be at least two, since a resilient overlay cannot have a single point of failure. Equations in group (ii) force every non-RR border router, full-mesh optimal for some other router and that prefixes class, to be adjacent to at least one internal router with preference for that class, which is forced then to be a reflector because of the following equations. Equations in group (iii) impose either

the internal router j or the border router i to be a reflector when $y_{ij}^k = 1$. Equations in group (iv) forbid $y_{ij}^k = 1$ when both i and j are reflectors. This is because in our model, reflector-to-reflector adjacencies are implicit, as they are in fact in the standard implementation of an iBGP overlay with route reflectors. Equations in group (v) guarantee that each internal router IR of a class k is either a route reflector; or it is a client of its preferred RR border for that class of prefixes; or, for that class, IR is adjacent to another affine internal router. In that last case, the peer has to be a preferred border for that class, which is provided by constraints (vi). Equations (vi) and (vii) combined force the number of reflectors in an IR-to-IR adjacency to be 1. This is also because of the implicit reflector-to-reflector adjacencies.

The formulation might look confuse at first glance. We revise now Equations 1 from a different point-of-view. First of all, recall that our target underlying design is that of a standard iBGP overlay with reflectors, and the application case is that of a transit backbone. Therefore, it is not sufficient that each internal router gets to know optimal external routes. That must be also accomplished for border routers, since they must relay optimal paths to neighbor ASes through eBGP. Recall that classes are processed up from a set of prefixes that have been filtered by the BGP selection process up to step vii), which states *Prefer eBGP path over iBGP path*. As a consequence, a border router would never get from other source a prefix it knows by its own. For instance, in Fig. 1 and for a prefix in classes A or B, `TS` will not get a different next-hop from that it has learned (via eBGP) from a peer AS. The exception is in the contingency where `TS` losses the eBGP session from which it has learned A or B. Therefore, iBGP optimality must be crafted for each class, either real or fictitious (created as a consequence of an adjacency loss). Moreover, the border or internal condition of an ASBR is also bond to each class. Despite that, the overlay is unique, and all the constraints should be satisfied simultaneously to have a feasible solution.

Given any class, a first design goal is: each prefix, optimal for some internal router of that class, must reach all RRs. The simplest alternative is to assign the reflector role to that border router. Complementary, when that border router is not a RR and since there is a full-mesh of implicit sessions among reflectors, suffices that the corresponding update reach some reflector R with preference for the originating border. Because of that preference, R will relay the update towards every other internal router, and also towards every client of it. Observe that the previous is accomplished by equations groups (ii) and (iii) in Equations 1. Since reflectors get the whole of the

optimal updates, remains now to be seen how iBGP clients get to them. The second design goal aims then at solving that problem.

If an internal router j is not a reflector (i.e. it is a client), then it could be a client of a preferred border router i or it should be adjacent to some h, an affine internal router for that class. This is basically what equations group (v) guarantee. Then, router h is to be a reflector, because equations (vii) and $(viii)$ force one and only one router to be a reflector, whenever there is an adjacency between them. There is a particular case to consider: the case where border i is not a reflector but it is the border that triggered the affinity between j and h. To be sure the update from non-reflector i is going to get to the reflector h, there must be an adjacency between them. That is precisely why equations (vi) are included. Observe that logically speaking, each equation in (vi) could be read as: if $x_i = 0$ and $x_j = 0$ and $z_{jh} = 1$, then $y_{ih} = 1$. This set of equations corrects those presented in [10] in the case that an IR is client of an ASBR chosen as RR.

Equations 1 formulation aims at minimizing the number of reflectors, which is of primordial importance. Usually, there is more than one combination of feasible solutions with the same number of reflectors, and some of them may allow a lower number of sessions, which is another goal, of secondary importance. The problem can be solved as a whole by adding y_{ij} and z_{hl} boolean variables, for respectively every possible border-to-internal or internal-to-internal adjacency. We also need to add constraints $|\mathcal{C}'|y_{ij} \geq \sum y_{ij}^k$ and $|\mathcal{C}'|z_{hl} \geq \sum z_{hl}^k$, which activate y_{ij} or z_{hl} whenever an adjacency is required for at least one class. Finally, the objective function could be rewritten to: $|V| \cdot \sum x_i + \sum y_{ij} + \sum z_{hl}$, in order to optimize both objectives simultaneously.

As mentioned in previous Section 2.1, there are a number of proposals for the iBGP overlay design problem. Many of them emphasize reliability aspects, but in general they do not present explicit analysis regarding node or link failure. In addition, we introduce explicitly the IP/MPLS transport layer, which is mandatory for ISP scenarios, as well as considering the interaction of the forwarding and control plane. Finally, a distinctive aspect of the ORRTD framework in comparison with previous work is the introduction of the concept of *prefix classes*, which group the route updates received at the ASBRs in order to simplify the selection of the optimal RRs in an iBGP topology, which is further described in the next section.

3.2. Classifying Internet Prefixes into Classes

The IP address space has been originally sub-divided into network classes (namely A, B, C) using 8, 16, and 24 netmask bits. Since the emergence of Classless Inter Domain Routing (CIDR) [42], crafted to alleviate the problem of IP address shortage and the fast growth of routing tables, netmasks can be arbitrary long, and since then a network representation is called a prefix; note that CIDR supports both IPV4 and IPV6. The idea is that multiple IP addresses must share the same leftmost bits in order to summarize them. The notation consist in specifying an IP address and a mask, and the smaller the number representing the mask, the wider the range of IP addresses covered. In the example shown in table 1, 1.0.128.0/24 is an address prefix with 24 netmask bits, which contains a range of 254 possible hosts from 1.0.128.1 to 1.0.128.254 (by convention .0 represents the network, and .255 is the broadcast subnet address), while prefix 1.0.128.0/19 host addressable range goes from 1.0.128.1 to 1.0.159.254 (8190 hosts). Note that the second IP prefix contains the first one in the example, which is a particular case of the prefix hierarchy. This addressing hierarchy influences network routing; on the one hand, we can perform *subnetting*, i.e. sub-dividing a prefix into smaller ones, while on the other hand, we can do prefix aggregation or *supernetting*, where many prefixes can be grouped into a bigger one. With subnetting the number of routes grow, while with supernetting (or aggregation), the number of routes decrease.

Table 1: Example of IP Prefixes Hierarchy

IP Prefix	**From**	**To**
1.0.128.0/24	1.0.128.0	1.0.128.255
1.0.128.0/19	1.0.128.0	1.0.159.255

Internet prefixes can be represented in a general n-ary tree. In Fig. 6 a partial extract of a prefix tree is shown; at the root there is a prefix that contains all other (the supernet), while the leaves represent the most specific IP prefixes (note that in this particular example the hierarchy is not complete). Observe that the range of IP addresses covered by 1.0.132.0/22 is also covered by the set of its children (subnetting is complete), and this means that every IP address in the range is represented by the children, and therefore the father prefix does not provide new information. On the other hand, if we look at prefix 1.0.128.0/19 the situation is different, as the set of its children do not cover its whole IP address range.

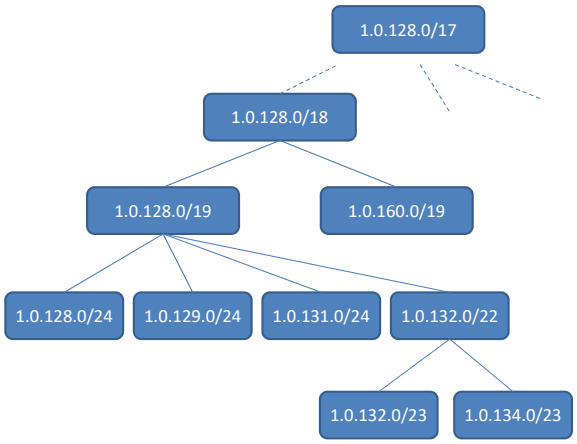


Figure 6: IP Prefixes Hierarchy Tree

When a network prefix gets announced or withdrawn, this information propagates through the Internet, impacting BGP traffic. Daily BGP update messages count has raised up to more than 100,000 in the last couple of years, as measured by Geoff Huston at APNIC[1], stressing BGP routers which need to maintain a local database with all prefixes announced by each routing peer, running the BGP decision process to select the best option among multiple route alternatives for a given destination prefix. Hence, a grouping strategy may achieve savings in both router resources and BGP traffic; BGP *policy atoms* are a possible aggregate of the IP space ([43],[44]). This concept was introduced to reduce complexity and has been also studied to predict failures ([45]); as defined by the authors, a policy atom is a maximal group of prefixes that have the same AS path to them from any major router (i.e. a router with default-free BGP table) in the Internet. At the time of that writing, the Atom count was about one sixth of the total prefixes in the DFZ, and roughly similar to the number of active ASes. Note that the longest prefix accepted by convention in IPv4 is that of a /24 prefix advertisement, which will propagate across the entire IPv4 default-free zone.

With these ideas in mind, an initial approach regarding prefix classes was presented in our previous work [7, 8]. We optimized the number of route reflectors taking into account the grouping of prefix classes at the border routers, obtaining early experimental results for two prefix classes. In [9] we

[1]BGP in 2018 Part2: BGP Churn: `https://labs.apnic.net/?p=1200`

extended the model by allowing ASBRs to play the RR role, and additionally showed the results for an increased quantity of prefix classes. In those works we assumed that prefixes were previously grouped somehow.

We now elaborate further in the concept of *Prefix Classes*, taking into account the fact that prefixes are hierarchically organized, and considering only those prefixes which remain in a tie after running rule *vii* of the BGP decision process. In the considered ISP case, this corresponds to maintaining prefixes with the same AS_PATH length, since administrative attributes and MED are not used. Anyway, the technique does not depend on how the attributes are processed and it works in any AS. Therefore, for the classification we basically consider only which ASBRs the prefix announcement are advertised from, and consequently the number of combinations cannot be greater than $2^n - 1$ classes, where n is the number of ASBRs. For example, if prefix p is advertised by ASBRs B_1 and B_2 and prefix q is advertised by B_1 and B_3, they belong to different prefix classes.

In the following paragraphs we formally explain the concept of prefix classes. Given a network represented as an undirected graph $G = (V, E)$, let BR be the set of border routers of the network. $BR \subseteq V$. Let n be the quantity of border routers, $n = |BR|$.
There is a set of Internet prefixes P, that survived to the BGP filtering algorithm up to step *vii*.

Definition 1. *Let $\mathcal{P}$ be the set of all possible prefixes/masks in the Internet, and BR a set of indices (ASRBs of an AS in our case). The updates function $FU : \mathcal{P} \to 2^{BR}$ is that which assigns to each prefix $p \in \mathcal{P}$ the subset of border routers that announce it in that AS. By 2^{BR} we refer to the "power set", i.e., the set of all possible subsets of BR.*

Definition 2. *Relation prefix-border (RPB): two prefixes p_1 and p_2 are related by RPB: $RPB(p_1, p_2)$, if and only if its images through FU coincide.*

Property 1. *The relation prefix-border RPB is an equivalence relation.*

Proof. To demonstrate that it is an equivalence relation, we have to show that this binary relation is reflexive, symmetric and transitive.

(a) Reflexive: $RPB(p_1, p_1)$ because $FU(p_1) = FU(p_1)$ as in any function.

(b) Symmetric: if $RPB(p_1, p_2)$ then $RPB(p_2, p_1)$. It is because of the symmetry in the equality relation and the functional definition of RPB. if $RPB(p_1, p_2)$ then $FU(p_1) = FU(p_2)$, then $FU(p_2) = FU(p_1)$, finally, it holds that $RPB(p_2, p_1)$.

(c) Transitive: it is the analogous argument that the prior. if $RPB(p_1, p_2)$ then $FU(p_1) = FU(p_2)$. On the other hand, if $RPB(p_2, p_3)$ then $FU(p_2) = FU(p_3)$. Thus, $FU(p_1) = FU(p_3)$ and $RPB(p_1, p_3)$. □

Observe that as with any equivalence relation, the previous property naturally splits Internet prefixes $\mathcal{P}$ into a set of equivalence classes (proper of each AS). These classes are in last term defined by the set of BGP attributes administratively managed by that AS, as well as from the particular set of border router announcing those prefixes.

Property 2. *Given any set of Internet prefixes $P \subseteq \mathcal{P}$ that survived to the BGP filtering algorithm up to step vii for any AS, whose number of border routers is n, it must hold that remaining prefixes capture all the routing information in at most in $2^n - 1$ equivalence classes of the RPB relation.*

Proof. After running steps i to vii of the BGP path selection algorithm we get to $P \subseteq \mathcal{P}$, a set of prefixes/masks that capture all the information necessary for any router in the AS to get to its optimal gateway. The last decision relies in the IGP shortest path towards an ASBR, which only depends on the internal placement of each router and the set of ASBRs updating about that prefix, what is in turn captured by the RPB relation. The RPB relation is defined up from a function whose co-domain has cardinality 2^n, so there cannot be more classes than that. Besides, from a practical point-of-view, the $\emptyset$ (empty-set) corresponds to those prefixes unannounced from any border, which are disregarded for practical purposes. □

Taking as an input the list of prefixes that are advertised from every ASBR after running rule *vii*, we build a hierarchical tree which maintains the prefix provenience (i.e., which ASBR advertised it), which is needed to determine if the prefix, and potentially its successors, are covered. The procedure is shown below in Algorithm 1.

- We consider as the root vertex the default prefix 0.0.0.0/0
- Each node has an attribute L: list of border routers which receive this prefix.
- $T = (V, E)$ is a general n-ary tree.
- *pos* is the position where p should be inserted or the position where it is found.

Algorithm 1 IP Prefix Tree Construction Algorithm

```
1: procedure BuildTree(T)
2:     for all prefix p do
3:         ReadPrefix(p)
4:         pos=SearchTree(T,p)
5:         if not found then
6:             InsertPrefix(T, pos, p, B)
7:         else if
8:             thenL=AddBorderRouter(T, pos, p, B)
9:         end if
10:    end for
11: end procedure
```

To consider resiliency, we take advantage of the prefixes hierarchy, and we define prefix protection in the following way:

- if prefix p arrives at more than one ASBR, it is protected
- if prefix p arrives at only one ASBR, find the nearest ancestor q that protects it (because any given prefix should have at least two different candidate ASBRs)

Besides, if a prefix p which covers IP address range r is announced, and its children cover the whole range r, then we can discard p, unless it is necessary to ensure protection as described in the previous paragraph: to ensure resilience it is important that at least two ASBRs receive the prefixes or a

super-class of them.

With algorithm 2 two sets of list are created: the first for the protected prefixes classes, which we call *Covered*, and the second one for unprotected prefix classes. The construction is achieved using a post-order traversal, i.e., from the leaves where the more specific prefixes are placed, to the root.

Covered: list of prefix classes
NotCovered: list of prefix classes

Each $l \in Covered$ or $l \in NotCovered$ has the following attributes:

- list of border routers L_b
- list of prefixes L_p

Algorithm 2 Classes Construction Algorithm

```
1: procedure BuildClasses(T)
2:     for all prefix p do
3:         if cant_ASBRS(p) > 1 then
4:             Add_to_List(p,L)
5:         else if p is not NULL then
6:             BuildClasses(Covered, NotCovered, p→child)
7:             if p→child is NULL then
8:                 if cantASBRs(p)>1 then
9:                     AddList(Covered, p)
10:                end if
11:                if Covered(p) then
12:                    AddList(Covered, p)
13:                else
14:                    AddList(NotCovered, p)
15:                end if
16:            end if
17:        end if
18:    end for
19: end procedure
```

In order to get an efficient traversal, we built in fact a collection of trees, instead of just one tree. In this way we got a faster look-up for the specific position of a certain prefix.

As we will further comment in Section 4, after applying rule *vii* of the BGP selection algorithm, the ISP we considered got about 800 thousand prefixes left, and therefore an upper bound of $2^n - 1$ classes, where n is the number of ASBRs, which may be around 10 in our case, is a very interesting result.

3.3. IP/MPLS and Traffic Engineering

Up to this point, link failures were intentionally left aside of the iBGP overlay, because they were to be protected during the tunnels traffic engineering. It is precisely in this stage where we must provide that resilience, while at the same time, we must prevent from congestion of links in the logical network and keeping end-to-end delays as comparable as possible to IGP metrics, for coherence with the iBGP overlay.

The data-set necessary to determine an instance for this problem comprises the following objects:

- The data layer $G = (V, E)$, where V represents the set of routers and E the set of connections among them (aka logical links).
- The lengths (or delays) for logical links, i.e., $L : E \to \mathbb{R}^+$. For consistency between iBGP and MPLS overlays, these lengths must match those costs used in Section 3.1.
- The capacity of logical links, i.e., $C : E \to \mathbb{R}^+$.
- The demands matrix between nodes, i.e., $D : V \times V \to \mathbb{R}^+$. For sake of simplicity we assume symmetry for demands, so $D(u, v) = D(v, u)$ for any u, v in V.
- The limit of delay for each tunnel: $MD : V \times V \to \mathbb{R}^+$, that is defined for those (u, v) such that $D(u, v) > 0$.
- The logical-to-physical dependence, which can be simply expressed by a boolean function $pd : E \times E \to \{0, 1\}$ that indicates whether or not any two logical links share a common physical one.

The Label Switched Paths or MPLS tunnels necessary to move traffic over the network are determined by those pairs (u, v) such that $D(u, v) > 0$. So, tunnels are implicitly determined by the input data-set, with the demand matrix as one of the components in that set. As a matter of fact, that demand matrix depends of the iBGP overlay and the scheme of connections between our transit AS and the neighbor ones. We elaborate about those details during the real-world application case (Section 4). The problem to be solved here is how to craft a scheme of primary and secondary paths for those tunnels, given the whole information.

Consider a directed graph $G' = (V, E')$ equivalent to the undirected graph $G = (V, E)$, where all edges are duplicated to include both directions. A possible set of control variables to model the traffic engineering problem consists of:

- Those variables that determine what path is going to be followed either by the primary or the secondary path over the logical network. Let $x_{ij}^{p,uv}$ be the boolean variable that indicates whether the logical link ij is going to be used as a hop within the primary path from u to v, while $x_{ij}^{s,uv}$ are the homologous for the secondary path;

- A set of auxiliary boolean variables $y_{ij,rs}^{uv}$ that indicate if the logical link ij is going to backup traffic from u to v after a failure in link rs. The previous happens as a consequence of using ij as a part of the secondary path for the tunnel uv and using rs in the primary path.

Three blocks of constraints are to be added to the problem to achieve consistency. The following block forces the construction of logically independent primary and secondary paths for each tunnel. The expression $E^+(u)$ in Equations 2 alludes to the set of nodes v in V such that there is an edge uv in E. Conversely, $E^-(u)$ is the set of nodes v such that vu is in E.

$$
\begin{cases}
\sum\limits_{j\in E^{+}(u)} x_{uj}^{p,uv}=1 & \forall u\in V, D(u,v)>0 \quad (i)\\
\sum\limits_{j\in E^{+}(u)} x_{uj}^{s,uv}=1 & \forall u\in V, D(u,v)>0 \quad (ii)\\
\sum\limits_{i\in E^{-}(j)} x_{ij}^{p,uv} - \sum\limits_{k\in E^{+}(j)} x_{jk}^{p,uv}=0 & \forall j\neq u,v, D(u,v)>0 \quad (iii)\\
\sum\limits_{i\in E^{-}(j)} x_{ij}^{s,uv} - \sum\limits_{k\in E^{+}(j)} x_{jk}^{s,uv}=0 & \forall j\neq u,v, D(u,v)>0 \quad (iv)\\
x_{ij}^{p,uv}=x_{ji}^{p,uv} & \forall ij\in E', D(u,v)>0 \quad (v)\\
x_{ij}^{s,uv}=x_{ji}^{s,uv} & \forall ij\in E', D(u,v)>0 \quad (vi)\\
x_{ij}^{p,uv}+x_{ij}^{s,uv}\leq 1 & \forall ij\in E, D(u,v)>0 \quad (vii)
\end{cases}
\tag{2}
$$

Equations (i) and (ii) in Equations 2 guarantee that a unit of flow is injected through one outgoing link from u for respectively both: primary and secondary paths, of any tunnel. Variables $x_{iu}^{p,uv}$ and $x_{iu}^{s,uv}$ are dismissed, so flow cannot drain backwards. Equations (iii) and (iv) are needed to preserve flow balance in any potentially intermediate node. (v) and (vi) impose both primary and secondary paths from u to v to follow the same path back and forth. Equations block (vii) seeks for logical links independence between primary and secondary paths for every tunnel. So far, we should have completed the design of a topologically consistent pair of logically independent paths for each tunnel.

It is worth mentioning that in this work, we are only concerned with link failures. Those referred to as "nodes" along this document, are actually Points-of-Presence (PoPs) in the real-world network. Furthermore, these PoPs are all Tier-4 Data-Centers, counting more than one router each, and routers are carrier-class routers (redundant control, forward, and multiple connections). As a consequence, nodes reliability is so much higher than links', that we are only aiming at protecting the later ones.

Relying on logical links independence is not usually a good idea to get to a resilient design. This is due to the fact that two or more logical links might share a common physical resource, an optical cable in our example. A usual pattern where that happens is a triangular-to-linear mapping. Suppose two optical fiber cables `A`-to-`B` and `B`-to-`C` are connecting sites `A`, `B` and `C`. With an

appropriate configuration of the add-drop optical multiplexers, three kinds of point-to-point optical links among these sites could be established. Those links would be: `A-B`, `B-C` and `A-C`. It is clear that the optical circuit `A-C` relies upon the operational condition of both optical cables, what means that a physical failure in `A-B` or in `B-C` translates into a failure in `A-C`. In such cases, logical independence is not enough for physical resiliency.

$$\begin{cases} x_{A,C}^{s,uv} + x_{A,B}^{p,uv} \leq 1, & \forall D(u,v) > 0, \quad (i) \\ x_{A,C}^{s,uv} + x_{B,C}^{p,uv} \leq 1, & \forall D(u,v) > 0, \quad (ii) \\ x_{A,C}^{p,uv} + x_{A,B}^{s,uv} \leq 1, & \forall D(u,v) > 0, \quad (iii) \\ x_{A,C}^{p,uv} + x_{B,C}^{s,uv} \leq 1, & \forall D(u,v) > 0, \quad (iv) \end{cases} \tag{3}$$

Sets of constraints as is Equations 3 prevents from primary and secondary logical paths to fall into a physical single point of failure, for the triangular-to-linear mapping. Our real-world application case only counts a couple of such exceptions, so a pair of constraints sets of that kind are going to be added for extending resiliency upon those exceptions. Since resilience is fully provided by combining Equations 2 and Equations 3, it remains to be seen how it is Quality of Service (QoS) guaranteed, which is introduced with Equations 4.

Variables where $x_{ij}^{p,uv} = 1$ define a path between u and v, as a result of Equations 2. Thus, equations blocks (i) and (ii) in Equations 4 account for the total end-to-end delay for either the primary or the secondary path, which must comply with delay limits for respective tunnels. According to the definition of $x_{ij}^{p,uv}$ and $y_{ij,rs}^{uv}$ variables, the left-hand side of (iii) merely adds up to the total traffic over ij for each tunnel uv under an rs failure scenario.

$$\begin{cases} \sum\limits_{ij \in E} L(ij) \cdot x_{ij}^{p,uv} \leq MD(u,v) & \forall D(u,v) > 0 \quad (i) \\ \sum\limits_{ij \in E} L(ij) \cdot x_{ij}^{s,uv} \leq MD(u,v) & \forall D(u,v) > 0 \quad (ii) \\ \sum\limits_{D(uv) > 0} D(uv)(x_{ij}^{p,uv} + y_{ij,rs}^{uv}) \leq \beta \cdot C(ij) & \forall ij \neq rs \in E \quad (iii) \\ y_{ij,rs}^{uv} \geq x_{ij}^{s,uv} + x_{rs}^{p,uv} - 1 & \forall ij \neq rs \in E, \; D(u,v) > 0 \quad (iv) \end{cases} \tag{4}$$

The right-hand side on (iii) sets an upper limit for that traffic that is proportional to ij link's capacity. That limit is bonded with the objective

function to optimize in this problem, which is: $\min \beta$, with $\beta \geq 0$. After the optimization process, this last variable β attains the reduction ratio in link's capacity beyond which no feasible solution can be found. If that optimal $\bar{\beta}$ is greater than 1, it would mean that current traffic with those delays and resilience constraints cannot be fit in a network with current capacities. Conversely, a $\bar{\beta}$ value less or equal to 1 indicates the actual instance is feasible, while the inverse of $\bar{\beta}$ measures how much greater that traffic could be before saturating the network. Finally, equations in (iv) in Equations 4 force consistency between $x_{ij}^{p,uv}$ and $y_{ij,rs}^{uv}$ variables, since $y_{ij,rs}^{uv}$ must be 1 when $x_{ij}^{s,uv} = 1$ and $x_{rs}^{p,uv} = 1$, which translates into: if ij is used by the secondary path of uv, rs is used by its primary path and rs fails, then uv traffic will go through logical link ij. A very simple but highly detailed example about how these equations allow to optimally solve such traffic-engineering problem can be found at [10].

Theorem 1. *The sub-problem of fitting independent primary/secondary paths within a network with capacities, turns NP-Hard the traffic-engineering problem of this section, regardless of delays limits.*

Proof. The proof lies under a polynomial reduction of the NPP (Number Partitioning Problem) to this one. It is in fact based upon the Theorem-7 [46], so here, we only summarize the main idea and changes. Refer to the original proof to complete technical details.

First of all we disregard Equations 3, that is, we assume that physical and logical links match one-to-one, which is a particular subfamily within this problem, which results from combining Equations 2 and Equations 4. Consider an instance of the NPP, that is, a list of positive integers: $a_1, a_2, \ldots, a_N$, for which we seek a partition $\mathcal{A} \subseteq \{1, 2, \ldots, N\}$ such that discrepancy:

$$E(\mathcal{A}) = |\sum_{i \in \mathcal{A}} a_i - \sum_{i \notin \mathcal{A}} a_i|,$$

finds its minimum value within the set $\{0, 1\}$, i.e., it has almost the same sum. It could happen that $\sum_{i=1}^{N} a_i$ is not even. NPP is a very well known $\mathcal{NP}$-Complete problem. Consider now the data network as in Fig. 7, where all links are physically independent.

Nodes v_1 to v_N in Fig. 7 are associated to numbers a_1 to a_N. Remaining nodes are always the same for any NPP instance. Let all lengths $L(i, j)$ be equal to 1, and take $MD(u, v) = 5 + 2N$, what to all effects deactivates

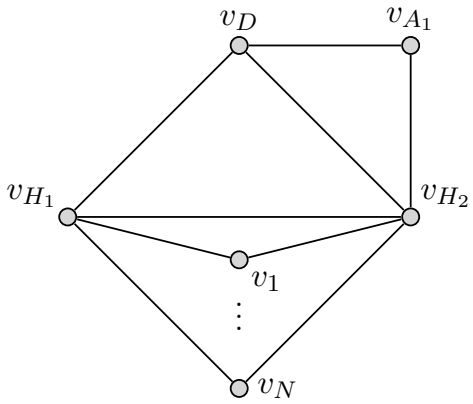


Figure 7: Network topology for NPP reduction

constraints (i) and (ii) (i.e. delay limits) in Equations 4. Assume that only demands are $D(v_i, v_D) = a_i$ and that the only capacity is $M = \lceil \frac{1}{2} \sum_{1 \leq i \leq N} a_i \rceil$.

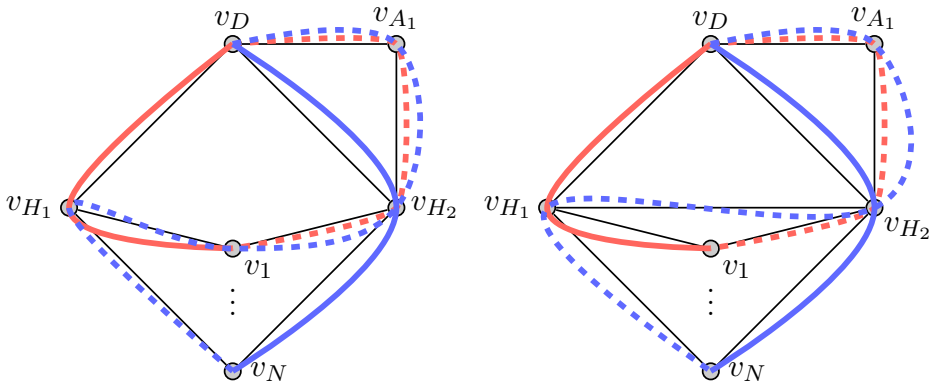


Figure 8: Primary and secondary paths of demands

It can be proven (refer to [46]) that when $\beta = 1$, our traffic engineering paths is feasible if and only if any of the solutions in Fig. 8 fits with capacities. Therefore, we can assert that the original NPP instance has a partition with almost the same sum, if and only if, the optimal solution to the problem in this section has $\hat{\beta} \leq 1$. The previous completes the reduction, which is obviously of polynomial complexity. □

4. The ISP case: multi layer planning

Fig. 9 summarizes the main Points-Of-Presence (POPs), connections (capacities and lengths included) and the main eBGP adjacencies of our referential case-of-study, which corresponds to a real-world transit backbone network with presence at South and North-America, concretely at: United States (US), Brazil (BR), Argentina (AR) and Uruguay (UY). The instance counts fourteen Points-of-Presence (PoPs) and other four virtual nodes. Actual nodes are: `AS` and `MI` at US; `PA`, `RI`, `SS`, `S1` and `S2` at BR; `BS`, `CA`, `PC` and `TS` at AR; and `MO`, `MV` and `PD` at UY. Since our models are not multigraph aware (i.e. there could be at most one link between each pair of nodes), fictitious nodes were introduced to emulate the existence of more than one physically independent link between some pairs of nodes. These fictitious nodes are `F1`, `F2`, `F3` and `F4`. Those eighteen nodes and their connections compound our reference AS network. As we mentioned, real nodes are actually PoPs with an arrangement of redundant carrier class nodes, so we only consider link and adjacency failures. The scheme in Fig. 9 also sketches eBGP adjacencies with others ASes, which are identified by clouds with their name and AS number. Yellow clouds correspond to Content Delivery Networks (CDNs) of some major content providers, namely: Google, Netflix, Facebook, Microsoft, Telecom and Internexa. White clouds are ASes of other transit ISPs, either regional or international. Green clouds are Internet eXchange Points, which provide point-to-point connections that multiplex peerings with dozens of other content or transit providers. Thus, the application case actually counts around forty eBGP peers. Whatever the flavor of the eBGP adjacency, they are represented with red stroke in Fig. 9. Black stroke lines in Fig. 9 correspond to logical links physically independent. Physically, each link can be ground or submarine supported. Each one has a distance, which is to be considered as the cost for the BGP overlay, and a bandwidth, used for the congestions calculation. Finally, dashed lines of other colors are logical links that are physically dependent of those with the same color and solid lines. They are `AS-SS`, which depends of `AS-RI` and `RI-SS` (triangular-to-linear mapping as in Equations 3, Section 3.3). An analogous case is `MI-MO` regarding `MI-SS` and `SS-MO`. The ASBRs are marked with lavender.

Links failures are going to be protected by the traffic-engineering of paths at the MPLS overlay. Adjacency failures in eBGP are protected by the IP routing level, i.e., by the optimal iBGP overlay. For the last case, the iBGP design protects the entire loss of adjacencies of each ASBR, one at the time,

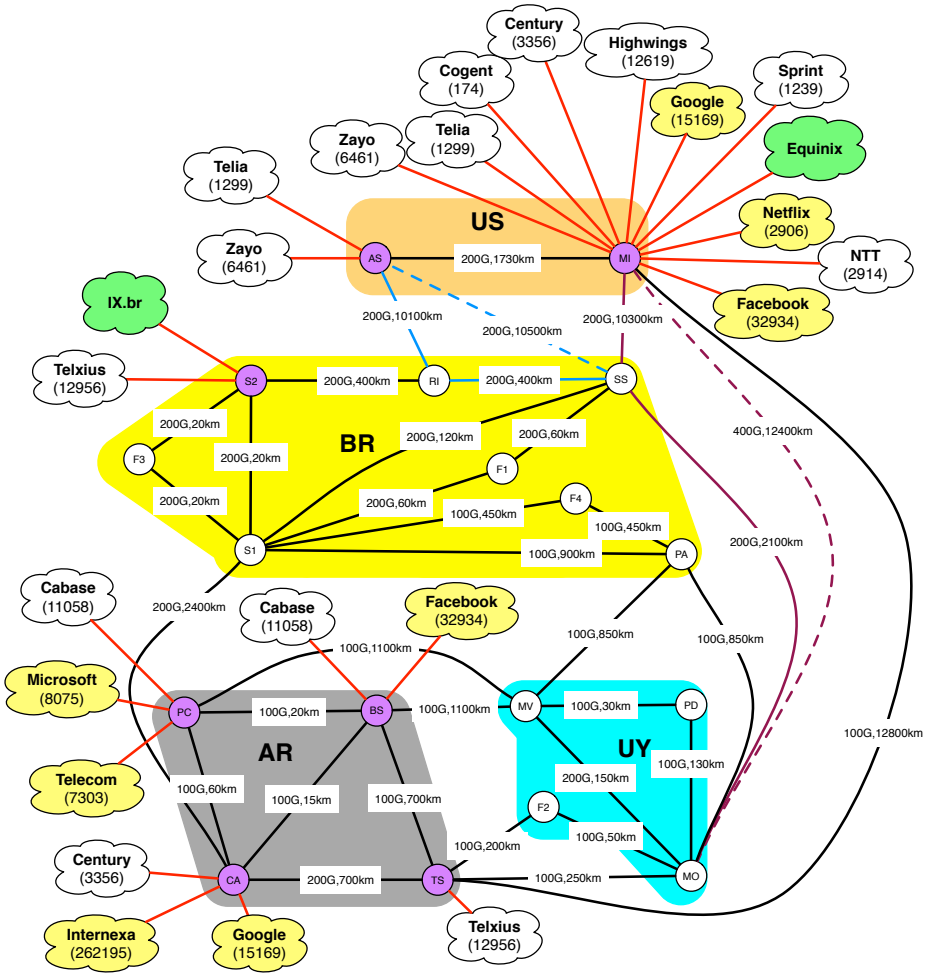


Figure 9: Latin American AS Network - the abstraction

preserving routing optimality. It actually happens that some adjacency fails from time to time, but what happens much more often is for content providers to move their traffic from a point of peering towards another, without previous notice. For instance, Google could move substantial portions of its traffic from `MI` to `CA` in a matter of minutes, or Facebook might do the same from `BS` to `MI`. The combined of Google and Facebook represents around 30% of the total traffic traversing the network of Fig. 9. The figure raises to almost 50% after adding up the other CDNs' traffic. Those percentages show how CDNs have changed traffic paradigms. Some years ago, it was usual to estimate traffic demands based on statistics analysis, and most ASes worldwide relied upon similar premises and homogeneous technologies to deal with the problem. Nowadays, CDNs, whose algorithms for traffic injection are part of dynamical proprietary software platforms, cannot merely be stated as statistical because they're not. They might be unpredictable and uncontrollable for ISPs, but they are far from being random. Instead, we must combine classic random failures with substantial changes in the absolute and relative weights of demands between points. Such uncertainty in the input data-set regards with a robust design rather than with resilience. A robust transport backbone must be prepared to such changes in traffic matrices. Those estimations are part of the iBGP (IP level) post-processing. A resilient transport backbone on the other hand must be tolerant to disruptions in optical cables, which should past unnoticed to end users. This second goal is strictly solved at the MPLS overlay, taking the previously computed robust demands matrix as an input. That is the strategy through which we seek for being tolerant against combinations of both situations.

The actual backbone topology as well as the logical-to-physical dependencies are part of the basic premises. Given the IP prefixes and the information about their traffic and the ASBRs that received it, the dsign process steps are the following:

i. Apply the BGP decision process to every ASBR (knowing their Rib-In) to get the preferred option for each prefix (in the emulated environment)
ii. Build the prefixes tree
iii. Group the prefixes into classes
iv. Determine the optimal iBGP overlay, by applying the ORRTD model
v. Calculate the demand matrix for the nominal case
vi. Calculate the demand matrix for each eBGP adjacency failure scenario, and up from it, compute the *worst-case* demand matrix.
vii. Determine the optimal MPLS overlay (primary and secondary tunnels)
viii. Perform a sensitivity analysis of the results of the previous stage to determine bottlenecks in the current design.
ix. Whenever result are not up the expectations of designers, explore expanding the topology, incrementally, starting by those links with the lowest investment-to-capacity return ratio.

4.1. BGP modeling and data processing

The experimental setup is composed of several tools, both real and emulated. On the one hand, the ISP has monitoring tools that capture both BGP and traffic data, including:

- Snapshot of ASBRs Rib-In databases
- On demand time-lapse captures of BGP updates arriving at the ASBRs
- Netflow and SNMP counters information for every ASBR

On the other hand, we model different scenarios using an emulation environment [47] based on Quagga[2], MiniNExT[3] (an extension layer to build complex networks in Mininet [48]) and ExaBGP[4] for injecting BGP messages.

[2]Quagga Routing Suite. Available at: `https://www.quagga.net/`. Accessed: 2019-08-20

[3]MiniNExT (Mininet ExTended). Available at: `https://www.quagga.net/`. Accessed: 2019-08-20

[4]`https://github.com/Exa-Networks/exabgp`

In order to pre-process the prefixes classes, we run the BGP decision process for every ASBR, using as input the Rib-In databases. From an original set of over 9 million eBGP updates, the BGP decision process produces around 800 thousand prefixes left to consider by the classification algorithms of Section 3.2, which in turn decreases to around 750 thousand by discarding prefixes of low specificity. This is because when a range of IP addresses is spanned by more than one prefix/mask entry, the router always chooses the most specific as its gateway. Those 750 thousand are in turn reduced to some dozens by grouping them into prefixes classes (also refer to Section 3.2).

Another important task performed in the emulation environment is the comparison of solutions to the iBGP overlay design problem, including ORRTD, BGPSep [17], BGPSep_D [18], BGPSep_S [19] and Zhang [22]. The input to the emulation environment are the network topology (including the neighbour ASes ASBRs), the RRs and BGP sessions (determined by the corresponding algorithm), and the BGP updates, injected with ExaBGP. Every algorithm is compared against the full-mesh solution, to verify its correctness.

The emulation environment also permits to test what-if cases, for example neighbor AS adjacencies down, to verify that backup ASBRs work correctly for every prefixes class. In a nutshell, several standard open-source tools were combined to either: filter eBGP updates to the path selection level, to check the correctness of solutions, and to compare with solutions found with other heuristic approaches.

4.2. Building the iBGP overlay

After building the prefixes tree and applying the technique explained in Section 3.2, we get to a set of 743033 effective prefixes, i.e., that could either be used as an active default route, or as a backup route after losing a more specific one that is not being published by more than one border router. So, all the prefixes are properly covered in the sense that an alternative advertising ASBR exists for each one.

Moreover, after grouping those prefixes into prefixes classes, we got just to 27 classes, with the 6 most extensive ones gathering 80% of the total prefixes, and the following 3 in importance raising that percentage over 90%. The whole information is in Table 2. Before going any further we briefly pause to reconsider those numbers. The fact that a transit backbone spanning the Americas along, which receives many millions of eBGP updates from over forty peers, could crush over 90% of that diversity to less than ten classes,

Table 2: Prefixes Classes

Class Id Identifier	ASBRs codes	Prefixes quantity	Cumulative Percentage
1	TS CA S2 MI	239183	32.2%
2	TS CA S2 AS MI	205359	59.8%
3	MI	53072	67.0%
4	CA	38908	72.2%
5	AS MI	36029	77.1%
6	CA MI	32652	81.4%
7	TS S2 MI	29510	85.4%
8	CA AS MI	26426	89.0%
9	TS S2 AS MI	21228	91.8%
10	TS CA S2	20504	94.6%
11	TS S2	18347	97.1%
12	TS	7914	98.1%
13	PC	4841	98.8%
14	CA S2 MI	3369	99.2%
15	CA S2 AS MI	3083	99.6%
16	S2	462	99.7%
17	AS	427	99.8%
18	CA S2	411	99.8%
19	S2 AS MI	387	99.9%
20	S2 MI	316	99.9%
21	PC CA	258	100.0%
22	TS CA	248	100.0%
23	TS CA S2 AS	48	100.0%
24	BS MI	34	100.0%
25	CA AS	11	100.0%
26	TS PC S2	8	100.0%
27	TS PC CA S2	6	100.0%

it is a major practical result for itself. This may indicate that with small adjustments, either by policies administration or negotiations with partners, prefixes classes might be reduced even more.

For the network shown in Fig. 9, and when minimizing both the number of RRs and the number of BGP sessions we get 6 RRs with the BGP adjacencies shown in Fig. 10. The nodes with red borders are the RRs, which must be connected with each other in a full mesh. The total BGP sessions found (37) are less than half the sessions in the case of a full mesh design, which translates into less administration effort. Observe that some of the RRs are ASBRs and the rest are internal routers. We also remark that the original number of prefix classes was 27, but 64 fictitious classes are created to ensure resilience.

Table 3 compares the number of RRs and BGP sessions obtained with different approaches, using one prefix family, i.e., all the prefixes arrive to all the ASBRs. In all the cases we obtain a reduced number of RRs and BGP sessions.

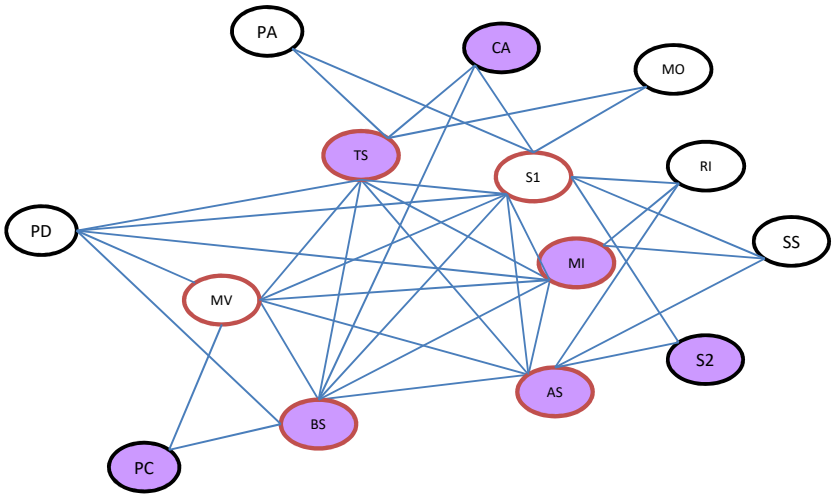


Figure 10: BGP Overlay

Table 3: Comparison of algorithms- 1 prefixes class

	ORRTD	**BGPSep**	**BGPSep_D**	**BGPSep_S**	**Zhang**
#RRs	3	6	6	7	4
# sessions	23	76	67	68	35

4.3. Traffic data processing

As it was mentioned in Section 4.1, Netflow and SNMP counters information for every ASBR that is available from ISP measurements. We use this information to determine the amount of traffic for each external prefix announced into the iBGP overlay. We have two different sources of information for every interface of every ASBR: i) 5 minutes samples Netflow records [49], and ii) SNMP traffic counters in RRD[5] format. While Netflow gives us the proportion of traffic originated for every external IP prefix, we still need to adjust the data in order to match the total traffic seen by the SNMP counters; basically we need to multiply every component of the traffic by the relationship among the total SNMP and Netflow data for each interface. This correction is applied for every 5 minutes interval under consideration.

A first outcome of the previous process is a *nominal* traffic matrix, in the sense that it is a snapshot of that matrix for that particular sampling interval, and for a fully operational network, with no link or adjacency failures.

A second important result, is that the process allows to attach the traffic by source as an attribute of leaves in the prefixes tree of Fig. 6. Observe that by going upwards in that tree while adding up traffic of children nodes, we can easily estimate by-source traffic for any node (prefix/mask) in the actual structure of prefixes with RIBs in our network.

The optimality of the iBGP overlay, and moreover, the particular topology of adjacencies in accordance with the preference and affinity trees for each loss of an eBGP adjacency, allows us to simulate how that nominal traffic matrix would change after losing each adjacency, one-by-one. The process gives actually a set of many traffic matrices, one per eBGP adjacency loss. From that set we can easily craft a so called *worst-case* traffic matrix, where each cell has the maximum value for that cell among all matrices. Being able to fulfill that worst-case traffic matrix captures the essence of the IP layer robustness we referred to previously.

The per-country summary of that information is detailed in Table 4. The total traffic in the worst-case (495Gbps) is 43% higher than in the nominal case (346Gbps).

Since solving an optimal traffic-engineering problem is of high computational intrinsic complexity (see final results in Section 3.3), we agreed with our counterpart ISP to manage a reduced number of MPLS tunnels. This

[5]Round Robin Database tools: `https://oss.oetiker.ch/rrdtool/`

Table 4: Nominal [leftmost] and worst-case [rightmost] demand matrices between countries for the real-world application case

	US	***BR***	***AR***	***UY***
US	0	24	24	93
BR	24	0	15	32
AR	24	15	0	158
UY	93	32	158	0

	US	***BR***	***AR***	***UY***
US	0	36	36	143
BR	36	0	21	46
AR	36	21	0	213
UY	143	46	213	0

is because the size of the problem is proportional to that number. Despite that, the agreed structure of tunnels precisely capture traffic between countries and span all nodes but SS, which is basically an optical articulation point between the Americas. So, the structure is quite representative of the scenarios. The whole set of tunnels –which is near to a full-mesh of among all nodes– is stated as future work, and its solution relies upon the development of a heuristic approach. Note that internal traffic is not considered, since we focus on a transit ISP.

Table 5: Nominal demand scenario for the reference set of tunnels

	AS	MI	PA	RI	SS	S1	S2	BS	CA	PC	TS	MO	MV	PD
AS	-	-	3	0	0	0	0	3	0	0	0	0	10	0
MI	-	-	0	10	0	11	0	0	0	11	10	0	43	40
PA	3	0	-	-	-	-	-	0	0	0	0	0	0	0
RI	0	10	-	-	-	-	-	0	0	0	0	0	0	0
SS	0	0	-	-	-	-	-	0	0	0	0	0	0	0
S1	0	11	-	-	-	-	-	0	0	0	0	0	0	0
S2	0	0	-	-	-	-	-	0	15	0	0	32	0	0
BS	3	0	0	0	0	0	0	-	-	-	-	30	0	0
CA	0	0	0	0	0	0	15	-	-	-	-	0	73	0
PC	0	11	0	0	0	0	0	-	-	-	-	35	0	0
TS	0	10	0	0	0	0	0	-	-	-	-	0	0	20
MO	0	0	0	0	0	0	32	30	0	35	0	-	-	-
MV	10	43	0	0	0	0	0	0	73	0	0	-	-	-
PD	0	40	0	0	0	0	0	0	0	0	20	-	-	-

Table 5 shows the demands for those selected tunnels in the nominal routing scenario. Table 6 shows the analogous information for the worst-case scenario.

Table 6: Worst-case demand scenario for the reference set of tunnels

	AS	MI	PA	RI	SS	S1	S2	BS	CA	PC	TS	MO	MV	PD
AS	-	-	6	0	0	0	0	6	0	0	0	0	23	0
MI	-	-	0	15	0	15	0	0	0	15	15	0	60	60
PA	6	0	-	-	-	-	-	0	0	0	0	0	0	0
RI	0	15	-	-	-	-	-	0	0	0	0	0	0	0
SS	0	0	-	-	-	-	-	0	0	0	0	0	0	0
S1	0	15	-	-	-	-	-	0	0	0	0	0	0	0
S2	0	0	-	-	-	-	-	0	21	0	0	46	0	0
BS	6	0	0	0	0	0	0	-	-	-	-	35	0	0
CA	0	0	0	0	0	0	21	-	-	-	-	0	88	0
PC	0	15	0	0	0	0	0	-	-	-	-	35	0	0
TS	0	15	0	0	0	0	0	-	-	-	-	0	0	55
MO	0	0	0	0	0	0	46	35	0	35	0	-	-	-
MV	23	60	0	0	0	0	0	0	88	0	0	-	-	-
PD	0	60	0	0	0	0	0	0	0	0	55	-	-	-

4.4. Traffic Engineering

Before entering into the MPLS overlay optimization, let us recall what data do we need to complete an instance. They are: i) the data layer $G = (V, E)$, with lengths and capacities, all of which are detailed in Fig. 9; ii) the demands matrix between nodes, for which in this application we have two scenarios: nominal and worst-case, whose details count in Section 4.3 (Table 5 and Table 6); iii) the logical-to-physical dependence that also is detailed in Fig. 9; and iv) the limit of delay for each tunnel.

Apart from the goals, demands matrices were derived from design premises: robustness and traffic statistics in our case. When it comes to delay limits, however, there is more than one criterion to set them. In fact, in this work we explored four alternatives before setting the definite values.

We aimed upon simple/abstract targets. As a general approach we seek for setting upper bounds for delays between countries rather than between nodes. For the first case we set the maximum delays for any tunnel between a pair of countries, as the largest "shortest path" between pairs of nodes of those countries. These values are detailed over the leftmost of Table 7. For example, the delay between Argentina and USA is set to the propagation time of a 12955km path, because that is the shortest distance between the two most distant nodes, which are AS and PC. In addition, that path is: AS,

RI, S2, S1, CA, BS and PC. We refer to delays and distances indistinctly, because over a so vast and non-congested high-speed network, propagation times are the most important by far.

Table 7: Distances for pairs of most-apart nodes between countries [leftmost]. The rightmost are maximum for those values after emulating every single physical failure.

	US	***BR***	***AR***	***UY***
US	1730	11420	12955	12300
BR	11420	1320	2855	2200
AR	12955	2855	720	1100
UY	12300	2200	1100	150

	US	***BR***	***AR***	***UY***
US	-	13150	14585	13930
BR	13150	-	3140	2320
AR	14585	3140	-	1130
UY	13930	2320	1130	-

Observe that inter-country distances are much higher than internal ones. Besides, since we ar considering a transit ISP application, we disregard intranational distances. Recall that our MPLS overlay must provide resilience against physical failures, so, there is certain chance that limits in the leftmost of Table 7 could not be attained when some links are unavailable. Distances were recomputed simulating the loss of every single physical link. Maximum values among all shortest paths for each failure scenario are those at the rightmost of that table. Most differences are negligible in relative terms. Those two values that proportionally increase the most are *BR-UY* (21%) and *AR-BR* (15%). In absolute terms, however, both figures are around 400km, what increases delay in a few milliseconds, unnoticeable for applications and users.

Limits in Table 7 would be referential values whether all nodes are to communicate among them, but, as it counts in Table 5 and Table 6, we are only considering a subset of those tunnels. Observe that by setting to 71435km the values for $MD(u,v)$ in equations groups (i) and (ii) of Equations 4, i.e., by using the cumulative length of edges in Fig. 9, we are in fact deactivating those equations. Instead, we can replace the objective function by $\sum_{D(u,v)>0} \sum_{ij \in E} L(ij) \cdot (x_{ij}^{p,uv} + x_{ij}^{s,uv})$ to minimize the total length of paths. By lowering demand (to ignore congestion issues) and after solving that problem variant, we get inter-country limits in between those of Table 7. After introducing real values for demands, some of those values had to be increased to be able to use other paths slightly larger, since some logical links were never used. Delay values finally used are those in Table 8.

Observe that distances between *AR-BR* and *BR-UY* can bet set below values on the leftmost of Table 8. Conversely, distance limit between *US* and *UY* must be above that value for the rightmost of Table 8; otherwise

Table 8: Definite delay limits used in the problem.

	US	***BR***	***AR***	***UY***
US	1730	13050	14565	14500
BR	13050	1320	2740	1940
AR	14565	2740	720	1115
UY	14500	1940	1115	200

some submarine cables from `MI` to `MO` or `TS` cannot be used to balance traffic among connections.

The detail that delay limits are very close to values in Table 7 is very important, due to the fact that those values are basically IGP metrics under different scenarios. Recall that in our real-world application, we are setting shortest path lengths between nodes as costs for tunnel interfaces. This is because we seek for the BGP decision process to use our traffic-engineered paths to deliver traffic. After finding solutions we are going to check that hypothesis against results.

We relied upon a standard optimization tool to find solutions to the combinatorial optimization model depicted in Section 3.3 for the network topology sketched in Fig. 9, delay limits as in Table 8, and tunnels demands of Table 5 and Table 6. Optimal solutions respectively have the following optimal values for $\bar{\beta}$: 0.85 and 1.25. It is worth mentioning that the number of variables for both instances is 34248, all of them boolean except for one (i.e. β). The number of constraints on the other hand is 19122. It took less than ten seconds to tackle the problem by using `IBM ILOG CPLEX(R) Interactive Optimizer` version 12.6.3 as the optimization software. The hardware was an `HP ProLiant DL385 G7` server, with 24 `AMD Opteron` processor 6172@2.1GHz and 64GB of RAM.

We will not elaborate about paths specifics now, because they change from solution to solution; instead, we analyze the overall performance. The first figure for $\bar{\beta} = 0.85$ might be read as: *the network has enough capacity to support these nominal demands and delay limits, even after any single physical failure.* Furthermore, there is a slack of 15% in links capacity, or, in other words, traffic should be 17.6% higher for some logical link to reach congestion under some failure scenario. Conversely, the other figure $\bar{\beta} = 1.25$ indicates that that network cannot comply with demands in the worst-case scenario. In fact, the best possible arrangement of paths for tunnels exceeds in 25% the capacity for some link at least in one failure scenario.

Table 9 summarizes links information and their worst congestion condition for the worst-case demand scenario.

Table 9: Links data and awaited load for worst-case demand scenario.

node Id1	**node Id2**	**length (km)**	**capacity (Gbps)**	**slack of bandwidth**
AS	MI	1730	200	50
AS	RI	10100	200	21
AS	SS	10500	200	194
MI	SS	10300	200	126
MI	TS	12800	100	25
MI	MO	12400	400	42
PA	S1	900	100	11
PA	MO	850	100	33
PA	MV	850	100	17
PA	F4	450	100	33
RI	SS	400	200	170
RI	S2	400	200	111
SS	S1	120	200	179
SS	F1	60	200	170
SS	MO	2100	200	194
S1	S2	20	200	65
S1	F1	60	200	170
S1	CA	2400	200	158
S1	F3	20	200	133
S1	F4	450	100	33
S2	F3	20	200	133
BS	CA	15	100	-23
BS	PC	20	100	44
BS	TS	700	100	15
BS	MV	1100	100	12
CA	PC	60	100	50
CA	TS	700	200	112
PC	MV	1100	100	94
TS	MO	250	100	-25
TS	F2	200	100	-25
MO	MV	150	200	-9
MO	PD	130	100	-15
MO	F2	50	100	-25
MV	PD	30	100	-15

Most data in Table 9 are directly taken from Fig. 9. The column *slack of bandwidth* results from taking the minimum for each link in equations group (*iii*) of Equations 4, after resetting β to 1. Observe that in fact, there are seven links that might fall into congestion (negative slack of bandwidth),

they are: BS-CA, TS-MO, TS-F2, MO-MV, MO-PD, MO-F2 and MV-PD.

Equations (*iii*) in Equations 4 give us a straightforward way to effectively and efficiently expand our network. The procedure simply goes by increasing the capacity of those links congested. The basic unit to increase bandwidths is 100Gbps. In fact, the speed of all actual data links is 100Gbps. To achieve higher bandwidth, Link Aggregation Groups (LAGs) are needed, i.e., bundle of links grouped as virtual interfaces, which load-balance their traffic in order to add up their capacities. After repeating the previous procedure twice, we get to the result in Table 10. Links whose capacities were updated are remarked in boldface. The total number of links is 12 out of 34, which might look substantial. Using the kilometers of 100Gbps links as a metric for the infrastructure cost, we conclude that the baseline network is 147.715km, while the second adds up to 163.215km. The second figure represents an increment of only 10% over the first, and except for MI-SS, all links to be increased are regional links. Therefore, the associated investments are affordable, and we may say that both designs are economically neutral.

When instead of investments we aim upon performance, differences are remarkable. First of all, observe that the links more compromised in terms of slack of capacity are: PA-S1, PA-MV and BS-TS, and for all of them the gap *capacity minus highest traffic* is 40% in the worst-case demand scenario, so demands could proportionally rise up to 67% without congesting any link, even after any physical failure and any combination of losses in adjacencies. In fact, the average *worst relative slack of bandwidth* is 53% for the worst-case demand scenario, 77% is the maximum, while the standard deviation is 10%. These figures show not only that after an additional 10% of investments, the network passed from having a lack of bandwidth of 25% to a surplus of 40%. They also show how balanced the use of links capacities is for that configuration of paths. As expected, analogous figures for the nominal demand scenario are even better, with a surplus of almost 50% and an average slack of capacity of 67%.

Details for optimal tunnels paths of the last network configuration are in Table 11. Primary or secondary paths are hop-by-hop specified in the first columns of both tables. The leftmost table corresponds to the primary, while the rightmost is for secondary path. Attributes common to both paths only appear in the leftmost table, they are: *IGP* and *Limit*. The column *IGP* corresponds to the shortest path when all links are operative, whose values might not be attainable after some physical failure. Column named

Table 10: Links data and expected load for both demand scenarios after updating capacities.

node Id1	**node Id2**	**length (km)**	**capacity (Gbps)**	**slack of bandwidth [nominal]**		**[worst-case]**	
AS	MI	1730	200	136	68%	110	55%
AS	RI	10100	200	123	62%	81	41%
AS	SS	10500	200	154	77%	134	67%
MI	SS	10300	**300**	203	68%	151	50%
MI	TS	12800	100	68	68%	55	55%
MI	MO	12400	400	320	80%	280	70%
PA	S1	900	**200**	114	57%	80	40%
PA	MO	850	**200**	153	77%	133	67%
PA	MV	850	100	57	57%	40	40%
PA	F4	450	100	65	65%	48	48%
RI	SS	400	200	124	62%	89	45%
RI	S2	400	200	133	67%	96	48%
SS	S1	120	200	121	61%	83	42%
SS	F1	60	200	174	87%	154	77%
SS	MO	2100	200	144	72%	119	60%
S1	S2	20	200	125	63%	94	47%
S1	F1	60	200	160	80%	124	62%
S1	CA	2400	200	133	67%	96	48%
S1	F3	20	200	143	72%	110	55%
S1	F4	450	100	65	65%	48	48%
S2	F3	20	200	143	72%	110	55%
BS	CA	15	**300**	197	66%	177	59%
BS	PC	20	100	65	65%	65	65%
BS	TS	700	**200**	95	48%	80	40%
BS	MV	1100	**200**	114	57%	91	46%
CA	PC	60	100	55	55%	42	42%
CA	TS	700	200	115	58%	100	50%
PC	MV	1100	100	79	79%	62	62%
TS	MO	250	**300**	196	65%	135	45%
TS	F2	200	**300**	162	54%	142	47%
MO	MV	150	**400**	260	65%	216	54%
MO	PD	130	**300**	230	77%	162	54%
MO	F2	50	**300**	162	54%	142	47%
MV	PD	30	**300**	240	80%	185	62%

Table 11: Primary and secondary paths for tunnels

Primary path	*IGP*	*Length*	*Spread*	*Limit*	*Secondary path*	*Length*	*Spread*
AS RI SS S1 F4 PA	11420	11520	0.9%	13050	AS MI SS F1 S1 PA	13050	14.3%
AS SS MO MV BS	12935	13850	7.1%	14565	AS MI SS S1 CA BS	14565	12.6%
AS RI S2 F3 S1 CA PC MV	12270	14100	14.9%	14500	AS MI MO PD MV	14290	16.5%
MI SS RI	10700	10700	0%	13050	MI AS RI	11830	10.6%
MI SS S1	10420	10420	0%	13050	MI AS RI S2 F3 S1	12270	17.8%
MI TS MO MV PC	12855	14300	11.2%	14565	MI SS S1 CA PC	12880	0.2%
MI SS MO MV BS TS	12420	14350	15.5%	14565	MI TS	12800	3.1%
MI SS MO MV	12170	12550	3.1%	14500	MI AS RI S2 S1 PA MV	14000	15.0%
MI MO PD	12300	12530	1.9%	14500	MI AS RI S2 S1 PA MV PD	14030	14.1%
S2 F3 S1 CA	2420	2440	0.8%	2740	S2 S1 PA MO TS CA	2720	12.4%
S2 S1 F4 PA MO	1770	1770	0%	1940	S2 F3 S1 PA MV MO	1940	9.6%
BS TS MO	950	950	0%	1115	BS CA TS F2 MO	965	1.6%
CA BS MV	1100	1115	1.4%	1115	CA TS F2 MO MV	1100	0%
PC BS TS F2 MO	970	970	0%	1115	PC CA TS MO	1010	4.1%
TS MO PD	380	380	0%	1115	TS F2 MO MV PD	430	13.2%

Length show actual length values for paths. Despite the fact that *lengths* for resilient paths should be usually greater than *IGP* values, they actually match, or its difference is bellow 1% for 10 out of the 30 paths. Paths lengths and IGP costs are actually pretty close in general, being 17.8% the worst relative deviation, while the average is 6.7%. The column *Limit* corresponds to the values in Table 8, which were mostly set to be near to the largest among shortest paths after single failures. *Lengths* are lower or equal to corresponding *limit* values, because it is a design premise (constraints (*i*) and (*ii*) of Equations 4). Moreover, for 17 out of the 30 paths, the actual length is below the average between the *IGP* and *limit* lengths. In general terms, we can assert that delays for traffic-engineered paths and homologous values dynamically computed by the IGP after link failures are quite similar. These results confirm the coherence between BGP and MPLS overlays.

As a final element of this analysis, we are going to compare the overall performance of the previous solution against that of an LDP based MPLS overlay. Recall that in this case, there is no need to coordinate costs of both overlays, since LDP always uses the shortest available path between any two nodes to move their traffic across the network. To compare performance, we craft a new table, equivalent to Table 10 but for this alternative implementation.

Differences between results in Table 10 and Table 12 are many and notorious. For the nominal matrix of demands, there are 4 links that would congest

Table 12: Expected load for both demand scenarios when using LDP signaling.

node Id1	**node Id2**	**length (km)**	**capacity (Gbps)**	**slack of bandwidth [nominal]**		**[worst-case]**	
AS	MI	1730	200	179	90%	165	83%
AS	RI	10100	200	163	82%	135	68%
AS	SS	10500	200	184	92%	165	83%
MI	SS	10300	300	159	53%	85	28%
MI	TS	12800	100	-4	-4%	-50	-50%
MI	MO	12400	400	400	100%	400	100%
PA	S1	900	200	33	17%	-52	-26%
PA	MO	850	200	65	33%	-4	-2%
PA	MV	850	100	-35	-35%	-104	-104%
PA	F4	450	100	-38	-38%	-110	-110%
RI	SS	400	200	190	95%	185	93%
RI	S2	400	200	173	87%	150	75%
SS	S1	120	200	69	35%	0	0%
SS	F1	60	200	85	43%	35	18%
SS	MO	2100	200	200	100%	200	100%
S1	S2	20	200	126	63%	83	42%
S1	F1	60	200	85	43%	35	18%
S1	CA	2400	200	171	86%	158	79%
S1	F3	20	200	137	69%	98	49%
S1	F4	450	100	-38	-38%	-110	-110%
S2	F3	20	200	137	69%	98	49%
BS	CA	15	300	213	71%	191	64%
BS	PC	20	100	54	54%	50	50%
BS	TS	700	200	121	61%	109	55%
BS	MV	1100	200	127	64%	112	56%
CA	PC	60	100	54	54%	50	50%
CA	TS	700	200	62	31%	42	21%
PC	MV	1100	100	100	100%	100	100%
TS	MO	250	300	59	20%	-33	-11%
TS	F2	200	300	132	44%	72	24%
MO	MV	150	400	274	69%	229	57%
MO	PD	130	300	207	69%	157	52%
MO	F2	50	300	132	44%	72	24%
MV	PD	30	300	187	62%	152	51%

after some physical failure. They are: `MI-TS`, `PA-MV`, `PA-F4` and `S1-F4`. For the last three of them, traffic exceeds links capacity by over 35%. When we look at the worst-case demand scenario, the list of congested links appends other 3 members: `PA-S1`, `PA-MO` and `TS-MO`. Congestion not only causes packet loss, but also increase point-to-point delays because of the queuing delays, that distort the IGP best path selection, which is only based in propagation times. The figures for this scenario are appalling. Three of those links would be receiving over twice the traffic they can handle, which translates in over 50% of packet loss. That figure would leave most applications at off-line status. Recall this is an international backbone, with high end-to-end delays, so standard traffic control mechanisms (as the TCP congestion window) cannot receive rapid feedbacks to adapt themselves.

Another remarkable fact is that there are 3 links that are never used: `MI-MO`, `SS-MO` and `PC-MV`. The combined capacity of these links adds up to almost 10% of the total. It makes no sense wasting resources in such way, while in parallel, several links are under extreme congestion. This kind of issue has been extensively documented in the bibliography. It is known as the *fish problem in routing* [50]. Our application case merely constitutes another example.

5. Conclusions and Future Work

In the present work we expand Internet prefix classes concept, that allows us to optimize a BGP overlay based on route reflection. We show how to classify and group Internet prefixes into classes, for the purpose of optimizing a network topology design, by building a prefix tree. In our experimental studies, even though theoretically we could have up to $2^n - 1$ prefix classes, where n is the number of ASBRs, we got a reduced number of classes - less than thirty -, and found that 20% of those classes represented 80% of the total number of prefixes, which suggests that, with small adjustments, maybe by partner agreements or just configuration, that number could be even smaller. Besides, we also propose a method to verify that a prefix class is *covered* in the senses that there is a less specific prefix arriving at a different ASBR, that includes each Internet prefix. This is very important to ensure complete Internet visibility.

Then an end-to-end application to a real world ISP provider topology design is shown, where both the control plane and forwarding plane over IP/MPLS is in place. A resilient and good QoS network is crafted, mixing

delay and capacity constraints assuming an optimal and resilient BGP overlay exists. For the purpose of the experimental case, real Internet traffic data captured with Netflow and SNMP for every ASBR that is available from ISP measurements where used. This information was used to obtain the Internet prefix classes, necessary to design the BGP overlay, and to study the network capacity and the tunnels congestion limits, both for the nominal and for each loss of an eBGP adjacency scenario. This process allows to attach the traffic by source in the prefix tree proposed, for each Internet prefix, and calculate the total demand.

Regarding the study of the MPLS tunnels it is important to remark that with the same network (the same traffic, topology and capacity), traffic engineering coordinated with routing gives us a slack scenario before failures exceeding 40%, compared to capacity deficits of up to 110% in LDP. It also tells us about the superiority of the coordinated IP/MPLS versus pure IP routing. Besides, the technique shows how balanced the use of links capacities is for that configuration of paths.

The techniques presented are based on heuristics and meta-heuristics applied to a combination of various optimization problems, which in previous research we demonstrated they are individually, $\mathcal{NP}$-hard. We also demonstrate that, regardless of delays limits, fitting independent primary/secondary paths within a network with capacities, turns $\mathcal{NP}$-Hard the traffic-engineering problem.

Taking into account the computational effort, future research could explore into how to improve the prefix classes, in order to get a simpler and more manageable topology. Regarding the integration of MPLS tunnels, the whole set of tunnels is stated as future work, and its solution relies upon the development of a heuristic approach.

It is important to remark that our proposal is from the point of design view, and we take a static photo of the BGP announcements, either withdrawal or insertions update messages, to deduce the prefix classes. Another possible research line is related to crafting those prefix classes in a dynamic way.

References

[1] D. O. Awduche, MPLS and Traffic Engineering in IP Networks, IEEE Communications Magazine 37 (12) (1999) 42–47. doi:10.1109/35.

809383.
URL https://doi.org/10.1109/35.809383

[2] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, F. True, Deriving traffic demands for operational IP networks: methodology and experience, IEEE/ACM Transactions on Networking 9 (3) (2001) 265–279. doi:10.1109/90.929850.

[3] T. Schuller, N. Aschenbruck, M. Chimani, M. Horneffer, S. Schnitter, Traffic engineering using segment routing and considering requirements of a carrier IP network, IEEE/ACM Trans. Netw. 26 (4) (2018) 1851–1864. doi:10.1109/TNET.2018.2854610.
URL https://doi.org/10.1109/TNET.2018.2854610

[4] Lixin Gao, On inferring autonomous system relationships in the internet, IEEE/ACM Transactions on Networking 9 (6) (2001) 733–745. doi:10.1109/90.974527.

[5] A. Elmokashfi, A. Kvalbein, C. Dovrolis, BGP churn evolution: A perspective from the core, IEEE/ACM Transactions on Networking 20 (2) (2012) 571–584. doi:10.1109/TNET.2011.2168610.

[6] L. Cittadini, S. Vissicchio, G. Di Battista, Doing don'ts: Modifying BGP attributes within an autonomous system, in: 2010 IEEE Network Operations and Management Symposium - NOMS 2010, 2010, pp. 293–300. doi:10.1109/NOMS.2010.5488479.

[7] C. Mayr, E. Grampín, C. Risso, Optimal Route Reflection Topology Design, in: Proceedings of the 10th Latin America Networking Conference, LANC '18, ACM, New York, NY, USA, 2018, pp. 65–72. doi:10.1145/3277103.3277124.
URL http://doi.acm.org/10.1145/3277103.3277124

[8] C. Mayr, C. Risso, E. Grampín, A Combinatorial Optimization Framework for the Design of resilient iBGP Overlays, in: 2019 15th International Conference on the Design of Reliable Communication Networks (DRCN), 2019, pp. 6–10. doi:10.1109/DRCN.2019.8713744.

[9] C. Mayr, C. E. Risso, E. Grampín, Designing an Optimal and Resilient iBGP Overlay with extended ORRTD, accepted for publication at the

Fifth International Conference on Machine Learning, Optimization, and Data Science (LOD) (2019).
URL https://www.fing.edu.uy/~crisso/LOD2019paper76.pdf

[10] C. Risso, C. Mayr, E. Grampín, A combined iBGP and IP/MPLS resilient transit backbone design, Accepted for publication at 11th International Workshop on Resilient Networks Design and Modeling (2019).
URL https://www.fing.edu.uy/~crisso/RNDM2019.pdf

[11] Y. Rekhter, T. Li, A Border Gateway Protocol 4 (BGP-4), RFC 1771 (Draft Standard), Obsoleted by RFC 4271 (Mar. 1995). doi:10.17487/RFC1771.
URL https://www.rfc-editor.org/rfc/rfc1771.txt

[12] T. Bates, E. Chen, R. Chandra, BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP, RFC4456 (Apr. 2006).

[13] M. Buob, M. Meulle, S. Uhlig, Checking for optimal egress points in ibgp routing, in: 2007 6th International Workshop on Design and Reliable Communication Networks, 2007, pp. 1–8. doi:10.1109/DRCN.2007.4762263.

[14] D. McPherson, V. Gill, D. Walton, A. Retana, Border Gateway Protocol (BGP) Persistent Route Oscillation Condition. RFC 3345 (2002).

[15] A. Basu, L. O. Chih-Hao, A. Rasala, B. Shepherd, G. Wilfong, Route Oscillation in iBGP with Route Reflection, in: Proceedings of SIGCOMM 2002 Conference in Pittsburgh, ACM, New York, NY, USA, 2002, pp. 235–247.

[16] S. Vissicchio, L. Cittadini, L. Vanbever, O. Bonaventure, iBGP Deceptions: More Sessions, Fewer Routes, in: INFOCOM, 2012, IEEE, 2012.

[17] M. Vutukuru, P. Valiant, S. Kopparty, H. Balakrishnan, How to Construct a Correct and Scalable iBGP Configuration, in: Proceedings of the 25th IEEE International Conference on Computer Communications., INFOCOM 2006, 2006, pp. 1–12. doi:10.1109/INFOCOM.2006.122.

[18] F. Zhao, X. Lu, P. Zhu, J. Zhao, BGPSepD: An Improved Algorithm for Constructing Correct and Scalable IBGP Configurations Based on

Vertexes Degree, in: Proceedings of the Second International Conference on High Performance Computing and Communications, HPCC'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 406–415. doi:10.1007/11847366_42.

[19] F. Zhao, X. Lu, P. Zhu, J. Zhao, Bgpsep_s: An algorithm for constructing IBGP configurations with complete visibility, in: Distributed Computing and Networking, 8th International Conference, ICDCN 2006, Guwahati, India, December 27-30, 2006., 2006, pp. 379–384. doi:10.1007/11947950_42.
URL https://doi.org/10.1007/11947950_42

[20] M.-O. Buob, S. Uhlig, M. Meulle, Designing optimal iBGP route-reflection topologies, in: Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet, NETWORKING'08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 542–553.

[21] A. Flavel, M. Roughan, Stable and flexible iBGP, in: Proceedings of the ACM SIGCOMM 2009 conference on Data communication, SIGCOMM '09, ACM, New York, NY, USA, 2009, pp. 183–194. doi:10.1145/1592568.1592591.
URL http://doi.acm.org/10.1145/1592568.1592591

[22] R. Zhang, M. Bartell, BGP Design and Implementation, Cisco Press, Indianapolis, 2003.

[23] R. Raszuk, R. Fernando, K. Patel, D. McPherson, K. Kumaki, Distribution of Diverse BGP Paths, RFC6774 (Nov. 2012).

[24] P. Marques, R. Fernando, E. Chen, P. Mohapatra, H. Gredler, Advertisement of the best external route in BGP, IETF Internet Draft – work in progress 05, IETF (July 2012).
URL http://tools.ietf.org/html/draft-ietf-idr-best-external-05

[25] D. Walton, A. Retana, E. Chen, J. Scudder, Advertisement of Multiple Paths in BGP, IETF Internet Draft – work in progress 08, IETF (June 2013).

URL http://tools.ietf.org/id/draft-ietf-idr-add-paths-08.txt

[26] V. Van den Schrieck, P. Francois, O. Bonaventure, Bgp add-paths: The scaling/performance tradeoffs, Selected Areas in Communications, IEEE Journal on 28 (8) (2010) 1299 –1307. doi:10.1109/JSAC.2010.101007.

[27] R. Raszuk, C. Cassar, E. Aman, B. Decraene, K. Wang, BGP Optimal Route Reflection (BGP-ORR), expires January, 2020 (2019).
URL https://tools.ietf.org/html/draft-ietf-idr-bgp-optimal-route-reflection-19

[28] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, J. van der Merwe, The case for separating routing from routers, in: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture, FDNA '04, ACM, New York, NY, USA, 2004, pp. 5–12. doi:10.1145/1016707.1016709.
URL http://doi.acm.org/10.1145/1016707.1016709

[29] I. Oprescu, M. Meulle, S. Uhlig, C. Pelsser, O. Maennel, P. Owezarski, oBGP: an overlay for a scalable iBGP control plane, in: Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part I, NETWORKING'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 420–431.
URL http://dl.acm.org/citation.cfm?id=2008780.2008822

[30] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, Openflow: Enabling innovation in campus networks, SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74. doi:10.1145/1355734.1355746.
URL http://doi.acm.org/10.1145/1355734.1355746

[31] S. Vidal, J. R. Amaro, E. Viotti, M. Giachino, E. Grampin, Rauflow: Building virtual private networks with mpls and openflow, in: Proceedings of the 2016 Workshop on Fostering Latin-American Research in Data Communication Networks, LANCOMM '16, ACM, New York, NY, USA, 2016, pp. 25–27. doi:10.1145/2940116.2940133.
URL http://doi.acm.org/10.1145/2940116.2940133

[32] H. L. et al., OSPF Traffic Engineering (OSPF-TE) Link Availability Extension for Links with Variable Discrete Bandwidth, RFC 8330 (Standards Track) (Feb. 2019).
URL https://tools.ietf.org/html/rfc8330.txt

[33] T. Li, H. Smit, IS-IS Extensions for Traffic Engineering, RFC 5305 (PROPOSED STANDARD) (Oct. 2008).
URL https://tools.ietf.org/html/rfc5305.txt

[34] A. Farrel, J.-P. Vasseur, J. Ash, A Path Computation Element (PCE)-Based Architecture, RFC4655 (Informational Standard) (2006).

[35] D. Awduche, J. Malcolmm, J. Agogbua, M. O'Dell, J. McManus, Requirements for Traffic Engineering Over MPLS, RFC2702 (Informational Standard) (1999).

[36] L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas, LDP Specification, RFC 3036 (Proposed Standard), obsoleted by RFC 5036 (Jan. 2001). doi:10.17487/RFC3036.
URL https://www.rfc-editor.org/rfc/rfc3036.txt

[37] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: Extensions to RSVP for LSP Tunnels, RFC 3209 (Proposed Standard), updated by RFCs 3936, 4420, 4874, 5151, 5420, 5711, 6780, 6790, 7274 (Dec. 2001). doi:10.17487/RFC3209.
URL https://www.rfc-editor.org/rfc/rfc3209.txt

[38] E. Grampin, J. Serrat, Cooperation of control and management plane for provisioning in MPLS networks, in: 9th IFIP/IEEE International Symposium on Integrated Network Management, 2005, pp. 281–294. doi:10.1109/INM.2005.1440798.

[39] F. Mereu, A., Cherubini, D. , Frangioni, A. , Primary and backup paths optimal design for traffic engineering in hybrid IGP/MPLS networks, in: 7th International Workshop on Design of Reliable Communication Networks, 2009, pp. 273–280. doi:10.1109/DRCN.2009.5339995.

[40] S. B. F. Skivee, G. Leduc, A scalable heuristic for hybrid IGP/MPLS traffic engineering - Case study on an operational network, in: 14th IEEE International Conference on Networks, 2006, pp. 1–6. doi:10.1109/ICON.2006.302621.

[41] C. Risso, S. Nesmachnow, F. Robledo, Metaheuristic approaches for IP/MPLS network design, International Transactions in Operational Research 25 (2) (2018) 599–625. doi:10.1111/itor.12418.

[42] V. Fuller, T. Li, J. Yu, K. Varadhan, Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy, RFC1519 (Sep 1993).

[43] k. Broido, A. , Analysis of routeviews bgp data: policy atoms, cooperative association for internet data analysis, in: NRDM workshop Santa Barbara, CAIDA, San Diego Supercomputer Center, University of California, San Diego, 2001.
URL https://www.caida.org/publications/papers/2001/NdrmBgp/NdrmBgp.pdf

[44] Y. Afek, O. Ben-Shalom, A. Bremler-Barr, On the structure and application of bgp policy atoms, in: Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment, IMW '02, ACM, New York, NY, USA, 2002, pp. 209–214. doi:10.1145/637201.637234.
URL http://doi.acm.org/10.1145/637201.637234

[45] R. Khosla, S. Fahmy, Y. C. Hu, Bgp molecules: Understanding and predicting prefix failures, in: 2011 Proceedings IEEE INFOCOM, 2011, pp. 146–150. doi:10.1109/INFCOM.2011.5934935.

[46] C. Risso, Using GRASP and GA to design resilient and cost-effective IP/MPLS networks, Ph.D. thesis, UdelaR/INRIA (2014).
URL https://www.fing.edu.uy/~crisso/Thesis.pdf

[47] V. Solla, G. Jambrina, E. Grampín, Route reflection topology planning in service provider networks., in: 2017 IEEE URUCON, URUCON 2017, 2017th Edition, Vol. 2017-December, 2017, pp. 1–4.

[48] B. Lantz, B. Heller, N. McKeown, A Network in a Laptop: Rapid Prototyping for Software-defined Networks, in: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets-IX, ACM, New York, NY, USA, 2010, pp. 19:1–19:6. doi:10.1145/1868447.1868466.
URL http://doi.acm.org/10.1145/1868447.1868466

[49] B. Claise (Ed.), Cisco Systems NetFlow Services Export Version 9, RFC 3954 (Informational) (Oct. 2004). doi:10.17487/RFC3954.
URL https://www.rfc-editor.org/rfc/rfc3954.txt

[50] E. Osborne, A. Simha, Traffic Engineering with MPLS, Pearson Education, 2002.

Part III

Conclusions and Future Work

Conclusions and Future Work

This thesis focuses upon the efficient use of BGP, particularly in the intra-domain scope with Route Reflection. An Integer Programming Problem formulation is developed to optimally select the route reflectors and its clients (i.e. the BGP sessions), introducing several variations: first the nominal case in pure IP networks when only internal routers can be selected as route reflectors; then single node or link failures are considered; after that, a relaxation is proposed when border routers are eligible as RRs; and finally, the integration with an IP/MPLS forwarding network is analyzed. The problem is proven $\mathcal{NP}$-Hard in general.

In addition, an innovative concept of *prefixes classes* is described, that allows to classify and group Internet prefixes into equivalence classes, prior to optimizing the network topology design, by building a prefixes tree. Even though theoretically, the number of classes could be as large as $2^n - 1$ (being n is the number of ASBRs), our real-world based experimental study has shown a much lower number, which in fact allowed us to tackle instances with exact methods. Besides, a method is proposed, to verify that a prefixes class is *covered*, in the sense that there is a less specific prefix arriving at a different ASBR, that includes each Internet prefix. This is very important to ensure complete Internet visibility.

An end-to-end application case is shown, which is based on a real-world ISP provider topology design, where both the control and forwarding planes over IP/MPS are to be coordinated. As a result, a resilient and QoS aware network is crafted by combining delay limits, capacity constraints and BGP optimality. Actual updates and traffic information were used to obtain Internet prefixes classes by building a prefixes tree. This tree is necessary for both: designing the BGP overlay and determining the configurations of demands the network should be prepared to handle. According to our experimental evaluation, such effort to optimally coordinate BGP routing and MPLS traffic engineering is thoroughly repaid in terms of quality and efficiency.

Future research could explore into how to coordinate external-BGP updates to improve the prefixes classes even more. Even to the point to make humans being able to intuitively capture interconnection essentials. Regarding the scalability of MPLS tunnels, tackling a whole set of tunnels is stated as future work, and its solution relies upon the development of a heuristic approach.

It is important to remark that this proposal is based upon a snapshot of the BGP announcements, so it must be regularly applied to keep consistency along time. Another possible research line is related to crafting those prefixes classes in a dynamic way to build the classes.