ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2016-ENST-0071

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité “Informatique et Réseaux”

présentée et soutenue publiquement par

Antoine FRESSANCOURT

le 25 novembre 2016

Conception et mise en œuvre d’overlays réseau dynamiques pour la résilience du Cloud : Vers une flexibilité et une résilience accrue du Cloud Computing

Improved resiliency for inter-datacenter network connections

Directeur de thèse: **Maurice GAGNAIRE**
Co-encadrement de la thèse: **Luigi IANNONE**

Jury

M. Olivier BONAVENTURE, Professeur, Université catholique de Louvain	Rapporteur
M. Philippe OWEZARSKI, Directeur de Recherche, LAAS - CNRS	Rapporteur
M. Josué KURI, Ingénieur réseau en chef, Facebook	Évaluateur
M. Jérémie LEGUAY, Ingénieur de recherche principal, Huawei Technologies	Évaluateur
Mme. Cristel PELSSER, Professeur, Université de Strasbourg	Évaluatrice
M. Stefano SECCI, Professeur associé, Université Pierre et Marie Curie	Évaluateur

TELECOM ParisTech
école de l’Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

T H È S E

Contents

List of Figures

List of Tables

Improved resiliency for inter-datacenter network connections

Abstract:

Nowadays, Internet services as well as applications delivered using Cloud Computing are hosted in large datacenters. The Cloud Service Providers (CSP) commit to give to their customers access to their infrastructures with a high level of availability. They also target a high level of reliability of the data processed in their computing infrastructure. In order to respect those commitments, CSPs replicate the applications they run and the associated data in remote datacenters. From a CSP's perspective, ensuring a high level of availability and reliability must be achieved while keeping the operational and capital expenditures as limited as possible. The coordinated operation of cloud applications from a set of remote datacenters requires using a performant and resilient connection scheme. In that extend, most CSPs deploy a mesh of private protected links rented from Internet service providers between their datacenters. *A priori*, such a connection scheme is costly and poorly flexible. Besides, from a CSP's point of view, including a new datacenter in the desired network is an expensive approach.

The goal of this PhD thesis is to allow CSPs to ensure their datacenter interconnections' resiliency using a flexible and affordable connection scheme. In that extend, we take advantage of the fact that datacenters are most of the time connected to the Internet through several Internet service providers. We design an overlay network architecture, referred as Kumori. Kumori stands for "cloudy" in Japanese. This architecture aims to detect and to react quickly to link or node failures affecting inter-datacenter communications over the Internet. This overlay consists in routing inflection points placed at Internet Exchange Points (IXP). The Kumori architecture is managed by a central controller.

After the description of the Kumori architecture, we evaluate its characteristics in terms of performance and resiliency. First, we compare Kumori's performance to the performance achieved by RON (Resilient Overlay Network), another overlay network architecture aiming at enhancing resiliency over the Internet. We then, characterize the resiliency benefits provided by the Kumori architecture. To do so, we evaluate the number of disjoint paths that it can establish between the datacenters of two CSPs: Atos and Amazon. For the sake of this evaluation, we build a directed graph representation of the Internet using three public data sources in which the graph nodes represent the various operators' geographical points of presence (PoP). Our Internet representation allows us to take into account the differences between network operators in the Internet. One of the major challenges we faced during this evaluation is related to the very large size of the graph we obtained, and to the algorithmic complexity of the path diversity search algorithm. At last, we discuss an evaluation of Kumori's costs of operation in order to evaluate the economical potential of this architecture.

Conception et mise en œuvre d'overlays réseau dynamiques pour la résilience du Cloud : Vers une flexibilité et une résilience accrue du Cloud Computing

Résumé:

Aujourd'hui, les services Internet ainsi que les applications de l'informatique en nuage, ou "*Cloud Computing*", sont hébergés au sein de grands centres de calculs et de stockage (datacenters). Les opérateurs de services "*Cloud*" (ou Cloud Service Provider (CSP)) s'engagent à fournir à leurs clients un haut niveau de disponibilité de leurs infrastructures et de fiabilité des données traitées. A cette fin, la duplication des applications et des données clients les plus massives peuvent-être amenées à se faire dans des datacenters distants. Du point de vue des CSPs, garantir un haut niveau de résilience doit se faire au moindre coût, tant en termes d'investissements matériel que de coûts opérationnels. L'opération coordonnée des services "*Cloud*" répartis sur plusieurs datacenters distants nécessite une connectivité performante et résistante aux pannes entre ces sites. A ce titre, la plupart des CSPs mettent en place un maillage de connections redondées à usage privatif louées auprès des fournisseurs de services Internet. *A priori*, un tel schéma de connectivité est coûteux, peu flexible, le délai d'inclusion d'un nouveau datacenter dans le maillage souhaité pouvant s'avérer prohibitif du point de vue de l'utilisateur final.

L'objectif de cette thèse est donc d'assurer la résilience des connections inter-datacenter au moyen d'une approche qui soit à la fois suffisamment dynamique et d'un coût de mise en œuvre inférieur à la constitution d'un maillage de lignes louées auprès des opérateurs. Dans cette démarche, nous nous appuyons sur le fait que les datacenters sont souvent connectés par le biais de plusieurs opérateurs à Internet. Nous définissons alors une architecture réseau superposée (overlay), que nous désignons par "Kumori" (le terme Kumori signifie "nuageux" en japonais) permettant de détecter et de réagir rapidement à des pannes de liens ou de nuds sur Internet entre différents datacenters. Cet overlay se compose de points d'inflexion de routage placés en coïncidence avec des IXPs (Internet Exchange Points). L'architecture Kumori est supposée être supervisée par un contrôleur centralisé.

Une fois l'architecture Kumori définie, nous cherchons à en évaluer les caractéristiques en termes de performance et de résilience dans un contexte le plus réaliste possible. Dans un premier temps, nous comparons les performances de l'architecture Kumori à celles de l'architecture RON (Resilient Overlay Network), une autre architecture visant à améliorer la résilience des connexions sur Internet. Nous cherchons ensuite à déterminer le gain en termes de résilience obtenu grâce à l'architecture Kumori vis-à-vis de l'architecture RON en évaluant le nombre de chemins divers pouvant être établis entre les datacenters de deux grands opérateurs de services "*Cloud*": Amazon et Atos. A cette fin, nous constituons une représentation d'Internet sous forme d'un graphe orienté à partir de trois jeux de données publics. Les nuds de ce graphe représentent les points de présence géographiques des différents réseaux

opérateurs sur Internet. Cela nous permet de prendre en compte les différences de taille entre ces réseaux. L'un des challenges les plus importants que nous avons rencontré lors de cette évaluation est lié à la taille très importante du graphe que nous avons manipulé et à la complexité algorithmique de la recherche de chemins disjoints que nous avons réalisée. Enfin, nous présentons une évaluation économique du coût d'opération de l'architecture Kumori afin d'en évaluer la pertinence économique.

Remerciements

A l'heure où une aventure passionnante et intense se termine, faire le bilan du chemin parcouru, se remémorer l'ensemble des discussions, des travaux et des échanges qui ont contribués à faire de cette aventure quelque chose de spécial et se souvenir de toute l'aide dont on a bénéficié est un exercice particulier.

En premier lieu, je tiens à remercier Maurice Gagnaire qui a été mon directeur de thèse et qui m'a perpétuellement accompagné et soutenu dans mes travaux. Je remercie aussi vivement Luigi Iannone, en particulier pour nos échanges sur mes travaux. Le travail que j'ai accompli a fortement bénéficié de vos apports, de vos remarques et de vos encouragements à poursuivre certaines pistes.

Je tiens à exprimer ma gratitude à Josué Kuri, Jérémie Leguay, Cristel Pelsser et Stefano Secci qui ont accepté de me donner de leur temps pour participer au jury de cette thèse. Je remercie particulièrement Philippe Owezarski et Olivier Bonaventure de m'avoir fait l'honneur d'accepter de prendre une part importante dans l'évaluation de mes travaux en étant rapporteur de ma thèse. Ma reconnaissance et ma gratitude sont d'autant plus fortes que j'ai le plus grand respect pour les travaux des membres de ce jury, et que ces travaux ont parfois inspiré certaines pistes que j'ai explorées.

Au cours de ma thèse, j'ai eu la chance et l'opportunité de confronter mes idées avec l'équipe d'IIJ Innovation Labs. Les échanges que j'ai eu avec les chercheurs de ce laboratoire ont fortement influencé mes travaux, et j'ai énormément appris, scientifiquement et humainement au contact de mes collègues à cette occasion. A ce titre, je remercie Kenjiro Cho de m'avoir permis de vivre cette expérience, ainsi que Megumi Ninomiya, Keiichi Shima, Randy Bush, Kazuhiko Yamamoto, Ray Atarashi, Yojiro Uo, Hideaki Nii, Tomonori Izumida, Mijung Kim, Lachlan Kang, Kenichi Takagiwa et Eiiti Wada. Je remercie tout spécialement Cristel Pelsser avec qui j'ai eu l'occasion de travailler plus particulièrement lors de ce séjour de recherche. J'ai énormément appris à ton contact, et j'espère que nous pourrons continuer certains travaux au-delà de cette thèse.

J'ai eu la chance d'effectuer ma thèse au sein du département INFRES de Télécom ParisTech. J'ai trouvé au contact de mes collègues un environnement amical et intellectuellement stimulant. Je remercie particulièrement Wenqin, Yimeng, Kévin, Jean-Sébastien, Sawsan, Ahmed, Céline, Amal, Felipe, Isabel, Mario, Jean-Louis et Xavier d'avoir contribué au plaisir que j'ai eu à travailler au sein de ce laboratoire.

Mes collègues au sein du département R&D de Worldline m'ont particulièrement soutenu lors de ma thèse. Ils ont compris mes déambulations en pleine réflexion dans les couloirs, m'ont écouté et m'ont donné leur avis lorsque je les ai sollicité. Si il m'est compliqué de citer tout le monde, je remercie particulièrement Stéphane, Paul-Edmond, Quentin, Li, Yacine, Tony, François-Julien, Colombe, Guillaume, Jean-Baptiste, Maxime, Eric, Denis et Nicolas. Je remercie tout particulièrement Jean-Claude, moins par ce qu'il est mon chef que pour m'avoir détourné d'une voie qui ne me conviendrait pas et pour m'avoir motivé et appuyé dans mon projet de thèse.

Sans son soutien, je ne me serais pas lancé dans cette aventure qui m'a changé.

Mes amis ont joué un rôle important lors de ces derniers mois. Ils m'ont aidé à garder un équilibre et m'ont parfois détourné de mon ordinateur pour me permettre de mieux y revenir. Je remercie donc Claire, Thomas, Axelle, Thomas, Lucie, Julie, Géraldine, Mathieu, Clément, Sarah, Jean-Marc, Chantal, Thomas et Nicolas pour leur amitié et pour les moments que nous avons passé ensemble.

Enfin, je remercie ma famille, notamment mes parents Anne-Marie et Patrick, Gilles et Laurence, mon frère Christophe, mes demi-frères et soeur Hugo, Gauthier et Dalann, Julien et Marina ainsi que mes grand-parents. Ils ont compris mon souhait de me lancer dans ce doctorat plusieurs années après avoir quitté les bancs de l'école et m'ont soutenu dans cette démarche personnelle. Enfin, j'ai bénéficié du soutien indéfectible de ma compagne, Evelyse, qui a tout fait pour m'aider au cours de cette aventure. Elle a compris que je passe de longues soirées et des week-ends sur mes travaux, m'a suivi au bout du monde, a trouvé normal que je me lève en pleine nuit pour vérifier quelque chose et m'a réconforté quand les choses ne marchaient pas comme je l'entendais. Désormais, une autre aventure nous attend, aider Mathias à grandir en lui donnant tout notre amour, tout notre soutien et toute notre attention.

Part I

English manuscript

Chapter 1

Introduction

Contents

1.1 Context

Information Technologies (IT) have been strongly impacted by the emergence of Cloud computing. A consensual definition of Cloud Computing has been proposed almost ten years ago by Ian Foster [FZRL08]:

> *A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.*

In other terms, thanks to Cloud Computing, companies or users may rent or buy externalized computing power, storage capacity, input/output interfaces or software services depending on their immediate needs. With Cloud Computing, end-users do not have to invest in their own infrastructure or licenses to fulfill their computing needs. Since its emergence ten years ago, Cloud Computing has strongly impacted both the hardware and the software industries. This quick and on-demand provisioning model for IT services promises to cover user's needs elastically.

Cloud computing defined by Ian Foster takes itself its roots in the concept of Utility computing, a term coined by John McCarthy in 1961. It gained popularity with the

development and generalization of Web applications accessible via a browser. Thanks to those Web applications, software is sold or delivered as-a-service rather than downloaded, installed and paid for through a one time license fee. Such a principle is referred to as the "Software as a Service" (SaaS) model. In the middle of the 2000's, Amazon applied a similar model to sell IT infrastructures or platforms through the Amazon Web Services. Amazon launched the Elastic Compute Cloud offer in 2006. This commercial offer proposes services accessible to any professional, academic or private user connected to the Internet. The concept of **Public Cloud** refers to this very open mode of operation for Cloud services provisioning. It assumes that, *a priori*, a same hardware or software resource can be shared in time to process successively tasks from different customers. The major Cloud Service Providers (CSP) such as Google, Amazon, OVH, IBM, Microsoft etc. exploit very large datacenters enabling to manage on a same site up to several hundreds of thousands of servers. Millions of virtual machines may be activated/deactivated simultaneously on these servers.

After a first period of skepticism, the Cloud computing model has now become mainstream. While at the beginning Cloud Computing was used to fulfill secondary IT needs, companies now tend to use those services even for their critical operations. The elastic scalability requirements associated to the necessity to guarantee a certain level of confidentiality to the end-user has led to the emergence of new system architectures presented as "Private Cloud". These new system architectures rely on the virtualization of both the hardware and software resources in the Cloud datacenters. Both Private Cloud and Public Cloud necessitate flexible virtualization and sophisticated orchestration techniques to satisfy the very dynamic workload submitted to the datacenters. Under Public Cloud, the hardware resources assigned to the end-users are generic in the sense they are imposed by the CSP. In other terms, the same set of hardware and software resources are proposed to the clients of a same CSP. At the opposite, Private Cloud enables not only to isolate the hardware (and eventually the software) resources assigned to each end-user, but also to customize these resources according to the clients requirements. Since Private Cloud uses the same technologies, interfaces and data models as Public Cloud, Public Cloud can be used to offload part of the work performed by private Clouds in case of heavy load. Such an approach is known as the Hybrid Cloud model.

As Cloud computing is gaining maturity, standards emerge either *de jure* through standardization bodies or *de facto* given the dominance of a few players on the market. This standardization allows several smaller Cloud Service Providers (CSP) to propose similar interfaces and methods to access their services. Those standard services can then be used interchangeably, while providers compete on price, availability or quality of service (QoS). The QoS provided by a CSP is roughly measured in terms of cost-efficiency ratio. For a given job, this efficiency can be expressed, by the average

and the maximum processing delays, in including the time for the CSP to set the associated VM and software environment. Efficiency must also take into account the robustness of the hardware/software resources provided by the CSP to the end-user. Indeed, the processing of a job strongly depends on the statistical multiplexing of various tasks sharing a same CPU, RAM, disk and I/O interfaces. It is under the responsibility of a CSP to decide of the amplitude of such statistical multiplexing. *A priori*, the higher the multiplexing rate, the higher the cost effectiveness of a server. Meanwhile, the higher the jobs multiplexing rate, the higher the probability to be subject to a job interruption due for instance to RAM overflow.

Besides the major CSPs that operate huge data centers interconnected at global scale (IBM, Google, Facebook, Amazon, Microsoft etc.), thousands of small/medium size actors have also emerged at a national or regional scale. Due to the disparity of scale between these two types of actors, two approaches have been observed these last ten years in order to reinforce the position of the small CSPs. These two approaches are known as Cloud Brokering ([DS14]) and Cloud Federation ([RBL+09]). On the one hand, Cloud Brokering can be compared to airline alliances that enable independent airlines to sell via a same Web portal tickets between any source and any destination in the world. On the other hand, Cloud Brokering finds its justification in the principle that union makes force. In other terms, a Cloud Federation associates small-medium size CSPs that may provide the same type of services but that decide to cooperate in order to increase globally their customers pool. For that purpose, the partners of a same federation accept to share their hardware and software resources. Specific business models still remain to be designed for that purpose.

In this context, Cloud resiliency remains at the date of publication of this thesis an open problem. Cloud resiliency is independent from the way a CSP manages the activation/deactivation of the virtual machines (VM) onto the physical machines (PM) of his datacenter. It may be declined in two complementary perspectives. The first one consists in guaranteeing the reliability of the PMs of the datacenter itself. This reliability strongly depends on the efficiency of the cooling process of the various racks in which are installed the PMs. The second type of failures refers to a disruption of the long-haul network connectivity between the end-users and the datacenter, and among the datacenters themselves. This thesis focuses on this second type of failures. From a network perspective, resiliency necessitates that the single or multiple data connections from the end-user's premises to the single or multiple datacenters hosting the jobs submitted by this same end-user are robust. Robustness means that these connections dispose of the sufficient capacity to proceed to the necessary data exchanges all along the computing task. More precisely, this type of resiliency assumes that there is no single point of failure in the connectivity between the end-user and its CSPs datacenter. The resiliency of the connectivity is particularly important in the case of critical services

such as banking or industrial production monitoring. In the worst case, even if a job has been computed correctly at the selected datacenter, a transient network connectivity failure occurring during the transfer of the results of this job from the datacenter to the customer premises becomes *de facto* unusable. Such a situation corresponds to a negative income for the CSP.

Today, major CSPs ensure the resiliency of their inter-datacenter wide area network by deploying private long-haul connections between their datacenters [JKM+13]. Those connections are either built on purpose by the CSP or sourced from multiple network providers. They are doubled in order to implement an active-passive failover strategy. This strategy suffers from several drawbacks. First of all, this connectivity scheme can be quite expensive for smaller CSPs, as it requires an upfront investment from the CSP to pay for the installation of the private link to the network connectivity provider. Besides, setting up such a private link between remote locations can take several weeks or months, which is quite long for some agile CSPs. It has to be noticed that, in the context of Cloud Federations, the CSPs of the federation interact with each other. In such a context, it is not economically viable to establish a private connection between each member of the federation. Such a constraint does not exist in the context of Cloud Brokering where a single connection is necessary between the end-user and the Web-portal of the broker.

1.2 Goals of the thesis

In this thesis, our goal is to design and evaluate the feasibility of an alternative to the dual private connections strategy used by most CSPs to interconnect their datacenters. This solution will ensure the resiliency, *i.e.* the resistance to link or node failure, of inter-datacenter connections. Our strategy will consist in trying to offer a large set of alternative disjoint paths in the Internet on which the inter-datacenter network traffic can be diverted in the event of a failure. Our solution will be targeted at smaller CSPs. It will offer those CSPs a middle ground between exchanging traffic between datacenters over the plain Internet and using a private dedicated connections network deployed by a network services provider. Our objective is to enable the CSPs to control for their inter-datacenter connectivity. Thus, the solution we will design needs to be as independent as possible from Internet services providers (ISPs) and from their Service Level Agreements (SLAs).

To address these design objectives, we propose Kumori, an overlay network architecture exploiting the principles of Software-Defined Networking (SDN). We have chosen to name the architecture after the Japanese word for "cloud" to indicate that it targets the requirements of CSPs. The Kumori architecture allows a CSP to control the

network traffic flowing among its servers inside each datacenter and in the Internet outside the datacenters' realms. The Kumori overlay differs from classical overlay networks designed to enforce resiliency in two ways: first, routing inside Kumori is controlled by a centrally logical controller inspired by the central controller in SDN or by the Path Computation Element (PCE) in Multiprotocol Label Switching (MPLS) networks. The use of this central entity ensures that the routing decisions are coordinated within the overlay, and that a globally optimal state has been reached. Besides, unlike classical resiliency-oriented overlays, the nodes used by Kumori to steer network traffic between the various datacenters of a same CSP are located at Internet Exchange Points (IXPs). Indeed, IXPs correspond to neutral environments within the Internet. They offer far richer connectivity possibilities than what can be offered to an edge node in the Internet.

1.3 Structure

The purpose of this thesis is to describe into details the Kumori architecture, namely, the overlay we have designed to facilitate the interconnection of CSPs datacenters while guaranteeing the survivability of these interconnections in case of link or node failures. In the remaining of this manuscript, we propose a quantitative performance evaluation of the Kumori architecture. We also evaluate the economical characteristics of the Kumori overlay architecture. This thesis is organized as follows.

In the current chapter (Chapter 1), we present the context of our work and introduce into more details the problem we want to tackle, as well as the requirements of the original Kumori overlay architecture proposed in this manuscript.

In Chapter 2, we draw an overview of the current strategy used by cloud services providers to interconnect their distant datacenters. We then recall various applications of the Software-Defined Networking (SDN) concept applied to wide area networks (WAN) (Section 2.1). In Section 2.2, we present and discuss previous approaches that have been proposed to enhance Internet resiliency. Section 2.3 introduces several graph representations of the Internet and methods to evaluate path diversity using those Internet topological representations. We underline the current lack of accuracy of these representations at the level of the Point of Presence (PoP). This prevents us from evaluating path diversity at this granularity level properly. Section 2.4 highlights the rich connectivity ecosystem proposed by Internet exchange providers (IXP). We analyze the pros and cons of various recent investigations highlighting the possibilities offered by the use of a Software-Defined IXP. We underline in Section 2.5 why previous projects aiming at enhancing Internet resiliency do not fulfill the requirements of the alternative solution we propose in this thesis.

In Chapter 3, we describe into details the Kumori architecture, a software-defined overlay that Cloud Services Providers may use to enhance the resiliency of their inter-datacenter connections. Section 3.1 details the requirements of this architecture. Section 3.2 presents the constituting elements of the Kumori architecture and how these elements interact together. Section 3.3 details the mechanisms that can be used by the nodes constituting the Kumori overlay to steer network traffic from one datacenter to another. Section 3.4 details the mechanisms allowing the Kumori architecture to detect a failure and to detour traffic around a detected link or node failure. Section 3.5 summarizes the key Kumori features. Finally, Section 3.6 presents the key performance indicators according to which we shall evaluate the Kumori architecture in the next chapters.

Chapter 4 details a first quantitative performance evaluation of the Kumori architecture in comparison with edge-oriented overlay networks such as the Resilient Overlay Network (RON) architecture. In this chapter, the performance of a path is evaluated in terms of number of hops. Section 4.1 discuss the metrics we consider in our performance comparison. In section 4.2, we present the methodology and the dataset we have exploited for this evaluation. Section 4.3 details the obtained performance results. We observe that Kumori yields different results depending on the size of the client CSP. For small size CSPs, Kumori enables to access a similar number of the shortest and diverse paths compared to the path that can be reached in using a classical edge-oriented overlay. For more popular CSPs, the Kumori architecture reduces the number of nodes required to access a similar set of equally short diverse paths among the CSP's datacenters.

In Chapter 5, we describe the method we use to build a a graph representation of the Internet at the PoP level. Section 5.1 stresses the issues related to the construction of such a topology. Section 5.2 shows how we combine several datasets to constitute a PoP-level representation of the Internet. Section 5.3 summarizes our achievements and gives high level characteristics of the PoP level graph we have obtained.

Chapter 6 details our evaluation of the benefits of the Kumori approach in terms of path diversity in using the PoP-level Internet graph built previously. Section 6.1 depicts the rationale of our evaluation, consisting in the comparison of the number of edge-diverse and node-diverse path available among a CSP's PoP pairs in the Internet and in using the Kumori architecture. Section 6.2 describes the algorithm we have used to evaluate path diversity in the various investigated configurations. Section 6.1 details our methodology. Section 6.4 presents the main results we have obtained. We then highlight the benefits provided by the Kumori architecture in terms of edge and node path diversity among PoP pairs of a same CSP. This improvement is more significant in two cases: for small size CSPs and for nodes that do not benefit from a high topological path diversity.

We pursue our evaluation of the Kumori architecture in Chapter 7 where we detail the economical aspects of our work. Section 7.1 provides a background on the economics of the relationships between network services providers and their customers in the Internet's core. Section 7.2 details the cost structure of various interconnection methods used by Cloud Services Providers to connect their datacenters between them and to the outside world. Section 7.3 presents the cost structure of the Kumori architecture. Section 7.4 details our comparison of Kumori's operational costs with the operational costs of a connectivity framework consisting in a set of protected private links between the datacenters of various CSPs. Our comparison highlights that Kumori's profitability compared to the classical inter-datacenter connection strategy used by CSPs depends on the ratio between the cost of transit network and the operational cost of required leased lines. Given the price dynamics in the market and the increase of inter-datacenter traffic volumes, we argue that the Kumoris architecture will become more and more attractive in the following years.

Chapter 8 concludes our manuscript. We summarize in Section 8.1 the main advances we have proposed in this thesis in matter of Cloud resiliency. Finally, Section 8.2 proposes a few perspectives in the continuation of our work.

1.4 Contributions

Along this thesis, I have had the opportunity to present my work in several occasions in the form of public communications:

- At the "Congrès DNAC" in 2014, during a talk entitled *"Vers un usage des SDN pour améliorer la connectivité inter-datacenters"*;
- At the "Journées RESCOM" in 2015, during a talk entitled *"Evaluation de la diversité de chemins sur Internet: dúne granularité au niveau des AS à une vision au niveau des points de présence"*.

Besides, I have presented three research papers:

- *A SDN-based network architecture for cloud resiliency* [FG15], co-authored with Maurice Gagnaire, which has been presented at the IEEE Consumer Communications and Networking conference in 2015. The work presented in this article is detailed in Chapter 3;
- *Kumori: Steering Cloud traffic at IXPs to improve resiliency* [FPG16], co-authored with Cristel Pelsser and Maurice Gagnaire, which has been presented at the 12th IEEE International Conference on the Design of Reliable Communication Networks in 2015. This paper has been granted the Best paper award at this conference. The work presented in this paper is detailed in Chapter 4.

- *"A Dynamic Offer/Answer Mechanism Encompassing TCP Variants in Heterogeneous Environments"* [FG14] at the International Conference on Advanced Networking Distributed Systems and Applications (INDS) in 2014. This article has been co-authored with Maurice Gagnaire. Although the contribution of this paper refers to IP traffic, it does not explicitly refers to Cloud resiliency. This is the reason why it has not been explicitly commented in this PhD manuscript. We have chosen to silence this contribution in this document for the sake of consistency

Chapter 2

Problem description and state of the art

Contents

2.1 Area overview

The aim of this thesis is to design a novel overlay network architecture enabling Cloud Service Providers (CSP) to interconnect their datacenters in order to increase the resiliency of their own infrastructures. To do so, it is necessary to better understand the

pros and cons of the solutions today adopted by the major CSPs for guaranteeing such a resiliency. The solution we propose to solve this problem relies on multiple concepts inspired by Software-Defined Networking (SDN). In a first step, we briefly review the existing resiliency techniques adopted by the ISPs. We then recall the basic principles of SDN applied to Wide Area Networks (WAN). In a second step, we provide a state of the art in matter of network resiliency. Then, we present existing Internet representations and methods to evaluate path diversity in the Internet. At last, we highlight open questions that will be addressed throughout the thesis to conclude this chapter.

2.1.1 Existing resiliency techniques

In order to prevent their services from going down if a whole datacenter (DC) fails or if a major disaster affects a whole region, CSPs usually deploy their services in several DCs spread around the globe. The services running in those distant DCs are synchronized and backed up using high capacity network links. Major CSPs such as Amazon, Facebook or Google build their own network out of optical fiber links they deploy or rent from infrastructure operators to interconnect their DCs [JKM+13]. Figure 2.1 presents the B4 network, Google's inter-datacenter wide area network as it was in 2011. This private long-haul links infrastructure spanned three continents to connect Google's 12 global datacenters. Due to its intrinsic cost, this strategy is only accessible to the few majors CSPs that operate at a world scale.

Figure 2.1 – Google's B4 network: an inter-datacenter wide area network [JKM+13]

As a replacement, smaller players rent dedicated private links or MPLS circuits from large Internet Services Providers (ISP). Thus, they often create a nearly-full mesh between their DCs. This full mesh is composed of over-provisioned links rent from several providers. The combination of over-provisioning and of the multi-vendor strategy is used to protect the CSPs from network equipment failures (links and nodes) between

DCs.

This private connectivity strategy used by the CSPs to ensure the resiliency of their inter-datacenter communications is difficult to enforce to secure the connectivity of the various users. Indeed, establishing a private connection between two networks is a long and costly procedure. CSPs such as Amazon try to address this issue. For instance, Amazon's AWS Direct Connect offering allows the end users to connect with a direct route to Amazon from a set of IXPs [AWS]. Yet, in this case, Amazon's customers need either to be present at one of the IXPs where Amazon is, or rely on an ISP to reach those IXPs with a sufficient QoS and resiliency.

Besides, in most cases, the resiliency of private connections used by smaller CSPs is ensured by network service providers (the telcos) directly concerned by the failing link(s). A fiber cut that occurs at the physical layer implies in general the disruptions of multiple connections at the application layer. The quality of service such as the mean and maximum packet transit delay between the edge nodes of two remote datacenters and the mean and maximum recovering delay of the data flow (at the TCP level) after failures is agreed upon in a contract between the CSPs and their network service providers. Thus, in case of physical link (optical, electrical, or radio) failure, CSPs rely on the ability of their network service providers to respect their contractual commitments. We estimate that this situation is not satisfactory for CSPs willing to control their own infrastructure. The solution we design in the remaining of this chapter enables CSPs to keep control as much as possible of the resiliency of their connections.

2.1.2 SDN paradigm for WAN

At the beginning of the years 2010's, while Cloud computing gained popularity and maturity, the networking world has also seen the rise of a transforming concept: Software-Defined Networking (SDN). SDN has its roots in the work of Martin Casado, Scott Shenker and Nick McKeown [MAB+08]. SDN is a technological concept in which a logically central entity, the **controller**, is managing the way network packets are forwarded by a set of networking equipment. In SDN, the network's behavior is programmable dynamically. The controller is thus hiding the network's configuration to the end-to-end application layer. In that regard, SDN can be seen as an enabler for network virtualization. SDN has initially been presented in [MAB+08] as a means to allow researcher to use campus networks for research on new networking paradigms. Today, SDN is used in many other contexts, ranging from experimental testbeds to heavily loaded production networks.

In particular, in the last few years, the ideas behind the SDN concept have been applied to the optimization of Wide Area Networks (WAN). This is what we call the Software-Defined WAN (SD-WAN). SD-WAN is used by companies or CSPs to control

the way their network traffic is routed between the local sites or branches of the company. SD-WAN is sometimes presented as an intermediate between the exclusive use of MPLS circuits or private dedicated connections between the local branches. In SD-WAN, resiliency and QoS are traditionally provided by means of over-provisioning, and the use of the plain Internet where the traffic is routed in best effort. In this context, SD-WAN allows network administrators to discriminate the type of traffic that flows on the MPLS circuits or on the Internet infrastructure.

SDN can also be used in a WAN context to steer the path taken by network flows between branches or sites in a dedicated connections fabric. For instance, in B4 [JKM+13], Jain *et al.* show the use of SDN to control the way datacenter to datacenter traffic is routed over Google's WAN production network which is composed of dedicated private links. In this typical case, the use of SDN together with the ability to influence the execution of some network-demanding operations allow Google to use some links in their WAN infrastructure at nearly 100% of their capacity, while the average link usage reaches 70%. Such utilization rates are far higher than the common over-provisioning method in which by design private WAN links are doubled. Each link should never be used at more than 50% of its capacity.

This example from Google tends to outline the feasibility of using a logically centralized controller to coordinate the way datacenters are interconnected over long distances. In this thesis, our goal is to use the principle of a centralized control to enforce nodes and links resiliency for inter-datacenter connectivity.

2.2 Network resiliency state of the art

Network resiliency can be tackled in many ways. Yet, most of the literature on the subject adopts the network provider's point of view. In the following paragraphs, we are putting the light on some research projects that can be used by network users to enhance their connections' resiliency. Several approaches can be adopted. In a first step, we examine how overlay networks can be used to improve resiliency. Then, in the same perspective, we analyze the benefits of multihoming (*i.e.* the fact that datacenters or enterprise networks are connected to several ISPs) and multipath. After these concepts, we introduce several projects relying on a centralized controller is used to enforce QoS and resiliency. At last, we briefly describe previous works using traffic rerouting in the Internet to facilitate failures bypassing.

2.2.1 Overlay

The Detour [SAA+99] project relies on an overlay network to route data traffic in using alternative links with a better QoS. In their article, Savage *et al.* observe that, statistically, for 30 to 80% of the failure cases observed on real scenarios in the Internet, there is for each primary path between two nodes an alternate path with better characteristics in terms of bandwidth, packet loss or round-trip time. From this observation, the authors suggest to use an overlay network in which the end nodes are connected via tunnels. Andersen *et al.* later proposed another approach in a project called Resilient Overlay Network (RON) [ABKM01]. RON consists in an overlay network architecture built with resiliency in mind. In this project, the overlay is composed of nodes that actively measure the characteristics (available bandwidth, round-trip-time, inter-packet jitter) of the links that connect them to the other nodes of the overlay. This active measurement enables to react dynamically to failures very quickly. At evidence the speed at which failures can be detected clearly imposes limits to the scale of the considered network configuration. In their article, Andersen *et al.* evaluate that beyond 50 nodes, the overlay cannot be used practically anymore without sacrificing the reactivity to failure events.

Gummadi *et al.* [GMG+04] have also addressed the issue of network resiliency. In their article, the authors outline that one-hop source routing can be used to route traffic efficiently around most failures on a path between two remote nodes in the Internet. A random node in an overlay is used to route the traffic around a detected link failure. Source routing is used to control the way traffic is routed through this node. Compared to RON, permanent link monitoring is not used, yet the proposed approach should enable to provision, for similar network configurations, alternative paths in case of link failure in shorter delays than those observed with the other approaches. Meanwhile, the simple strategy exposed in this work cannot avoid the last hop link failures if the destination node is not multihomed.

In the projects we have presented this far, the nodes constituting an overlay collaborate in a decentralized fashion. In [HHL+09], Ho *et al.* deal with the situation where the destination node is effectively not connected to at least two upstream nodes. Ho *et al.* show that using a centralized approach allows to discover more detour paths than using a decentralized approach. Besides, in their project, the authors outline the benefits of taking into account the underlying network topology in the constitution of a resilient overlay. Similar ideas have been exploited in the Resilient Overlay for Mission Critical Applications (ROMCA) project [ZP09]. In this architecture, a central directory is used by the nodes to facilitate the other nodes' discovery. A decentralized routing protocol is used for the specification of the links between the nodes of the overlay. To take the network topology into account, path diversity among the overlay is regularly

measured using the ICMP-based traceroute technique. Link protection and resource allocation to the benefit of applications are achieved in using the RSVP-TE protocol inherent to the MPLS protocol.

2.2.2 Multihoming

Besides overlay networks, multihoming can be used to enhance network resiliency, as demonstrated by Akella *et al.* [AMS+03]. In their article, the authors have shown that being connected to 4 ISPs is enough for an entreprise to achieve an effective resiliency provided that the chosen ISPs have few overlapping network paths. This work opens the opportunity to relax the constraint on network connectivity providers to achieve a better resiliency. The authors warn that oscillations between providers on a short time scale may induce routing instability.

Other projects have combined multihoming with the approaches presented in Section 2.2.1. For instance, in [HWJ08], Han *et al.* have designed a topology-aware network overlay using multihoming in order to enhance the resiliency properties obtained in the Detour [SAA+99] or the RON [ABKM01] projects. For these other projects, nodes are placed in the network taking into account network topology. This topology information is also used in the construction of alternative detoured data paths to try to redirect traffic through only one node. Multihoming allows the overlay to bypass last hop link failures. The applicability of the approach presented in [HWJ08] seems hardly feasible in a context where the network is not under control, nodes placement being crucial.

2.2.3 Multipath

The Overlay and the multihoming techniques increase resiliency by allowing data traffic to be redirected onto another path once a failure has been detected. Another approach may consist in combining multiple paths or links in a virtual connection to prevent the effects of a single link failure. With the SmartTunnel [LZQL07] approach, Li *et al.* have proposed to use logical end-to-end tunnels combining several paths to ensure resiliency. SmartTunnel combines Forward Error Correction (FEC) and multipath to cope quickly with bursty packet losses that FEC is not able to correct. The multiple paths constituting a tunnel are chosen to be topologically diverse.

2.2.4 Centralized control

Multipath is also used in CORONET (Controller based Robust Network) [KST+12]. This DARPA project's goal is to design a scalable fault tolerant network architecture with multipath support. In this network architecture, a centralized SDN controller steers the way traffic is routed. Even if this project is heavily oriented towards optical networks, the considered resilient architecture is quite generic. A single control plane is used whatever the dataplane. Similarly to our architecture, CORONET is designed to be used in a Cloud application context. A noticeable difference with our approach is that CORONET is not an ISP-independent solution: it uses information about the network state coming from physical network routers, in particular optical network routers.

Several approaches have been studied and proposed so far as Internet drafts or RFCs to address resiliency issues in MPLS networks. These efforts have been reviewed in two articles. In [RI07], Raj *et al.* present the different approaches used by the MPLS community for Fast Reroute. Fast Reroute aims at speeding up network recovery after a link or a node failure in order to reach the 50 ms recovery time target specified by the ITU-T at the physical layer. The approaches presented in this survey use MPLS mechanisms such as RSVP-TE or LSP to set up, advertise and reserve resources for alternative paths in case of link or node failure. The concepts presented in this survey cannot be fully applied in our architecture because the control of the network is required to deploy most of these fast reroute techniques. In [PCG+13], Paolucci *et al.* review the issues and research work done on the Path Computation Element (PCE) in MPLS networks. PCE can be used by network operators to enhance network survivability in an inter-domain context, *i.e.* in a context where MPLS networks in different ASes need to cooperate. PCE consists in the usage of a centralized network element to control the routing policy. We propose to exploit this property of PCE to be used as a resiliency policy manager in MPLS networks. The approaches presented in [PCG+13] are not applicable to our context because domain administrators need to collaborate to provide information about their respective networks. This is not an option for our architecture since we want to enable the CSPs to be independent from the network connectivity providers.

Since the emergence of the SDN concept, several projects have been dedicated to the resiliency of both the controller and the data plane of SDN networks. In this matter, the objective is to reach carrier grade restoration performance (typically less than 50 milliseconds for real time interactive applications, according to ITU-T standards). In most projects addressing the problem of the resiliency of SDN controllers, each SDN controller cooperates with other controllers located in the neighborhood. In case of failure of a controller instance, another controller instance located in the surroundings can take in charge the control of the data plane equipment of the failing controller. In the

DISCO project [PBL14], SDN networks are divided into regional areas. Thus, each area (referred above as surrounding) is made of a set of network equipment (IP routers or packet switches) steered by a same controller. The multiple controllers of a public network regularly exchange messages to synchronize their operation and to update their knowledge of the installed forwarding policies. In case of failure of an area controller, the management of the equipment steered by this controller is redistributed onto the other controllers located in the surroundings. In the **Orion project** [FBG+14], the controllers managing an SDN network are organized in a logical hierarchical tree of clusters. Thus, several controllers at level n operating in a same surrounding are supervised by master controller operating at logical level n + 1. Ideally, a master controller at level n + 1 is located at the geographical center of the cluster of controllers it protects. In a same cluster, controllers communicates with the other members of their cluster by means of "Hello" messages. In case of failure of a controller at logical level n, the master controller of this router that operates at logical level (n + 1) of the tree designates a backup controller at level n for the equipment previously controlled by the failing controller.

In SDN, the data plane needs to be monitored by the central controller to detect failures. Both passive and active monitoring techniques can be adopted for that purpose. The OpenNetMon project [vADK14] and the Payless project [CBAB14] both propose a method to passively monitor SDN networks. In those projects, OpenFlow messages are used by the controller to gather statistics on the network flows passing through the data plane equipment. Both projects use the fact that, in OpenFlow, the rules sent by the controller to the switching/routing equipment to route the various network flows have a variable time validity. When a rule expires, the multiple network equipment concerned by this expiration send to the controller a *FlowRemoved* message. This message contains a set of statistics on the packets belonging to this network flow that were sent and received by the equipment. In both the OpenNetMon and the Payless projects, the accuracy of the monitoring depends on the time validity of the rules provided to the controller. Yet, short time validity puts a stress on the controller. In order to reduce this stress on the controller, both projects adopt an adaptive mechanism to lengthen (or to shorten respectively) the monitoring intervals when the situation is stable (or fluctuant respectively). While passive monitoring has the advantage to piggyback on the exchanges of the controller with the network equipments it manages, it may be a bit too slow to detect a sudden failure in the network topology. In order to facilitate a faster failures detection in SDN networks, other projects complement passive monitoring with active probing. In [SSC+11] and in [AAK14], Bidirectional Forwarding Detection (BFD), a Hello protocol described in [KW10a] and [KW10b], is used to detect actively failures on the link between two nodes. In [SSC+11], BFD is combined with a pre-computed path protection strategy similar to what can be done in MPLS fast reroute to achieve carrier-grade failure recovery in a large size SDN network.

Those applications of resiliency principles to SDN networks are very interesting with regards to our work. Yet, we need to determine whether those approaches are adapted to an overlay network spanning remote sites. Indeed, in a global overlay, both aspects lay be an issue. The first aspect refers to the latency to reach the controller. The second aspect refers to the fact traffic flows in overlay tunnels rather than in controlled links may also be an issue.

2.2.5 Relaxing routing constraints to enhance network resiliency

Several research projects evaluate the benefits of new routing paradigms in the Internet to tackle failures. In [WZMS07], Wu *et al.* characterize the capacity of the Internet to resist against a set of specific failure scenarios. In this paper, the Internet is represented as a directed graph at an Autonomous System (AS) level in which a classic routing policy model, the *Valley-free* [GR00] model, is used. In their findings, the authors highlight the fragility of the Internet in some failure scenarios. They present some possible relaxations of routing policies beteen AS to address some of these weaknesses. Later, in [HCC+12], Hu *et al.* also consider relaxation of routing policies to address major Internet failures following natural disasters. They acknowledge the role of Internet eXchange Points in the connectivity of the Internet. They suggest to relax routing policies at specific IXPs and to set up peering agreements for a limited period of time between some ASes to facilitate the failure recovery process.

This state of the art reveals the potential of using overlays to enhance end-to-end connections' resiliency. Rather than repairing failures, those overlays rely on their capacity to detour traffic around potential problems to ensure a destination's reachability. Yet, in those overlays, the coordination among nodes is not centralized, which may lead to a suboptimal use of available connectivity resources. Besides, the use of SDN concepts to foster network resiliency has also been addressed in previous work. Nevertheless, in those SDN projects, the controller and the network equipment under its control were located in a same network or administrative zone. The use of such approaches in a WAN context might be problematic.

2.3 Evaluating path diversity in a graph representation of the Internet

Beyond the design of the Kumori overlay network architecture we propose, a major part of our work has consisted in the evaluation of the benefits provided by this architecture compared to alternative solutions presented in 2.2. In order to perform fairly this comparison, we have considered some of the available representations of the Internet

graph from the recent literature. We have also investigated from the recent literature some of the methods proposed these very last years enabling to evaluate path diversity in the Internet.

2.3.1 The Internet topology and its representations

After the Internet gained popularity at the end of the years 1990's, it became more appropriate to consider its network topology as an evolutive ecosystem than as a static construction following simple design rules [CMM99]. Several research projects have investigated and compared various techniques to measure and derive the Internet infrastructure. In this matter, the Center for Applied Internet Data Analysis (CAIDA) is a collaborative organization (academia, vendors, telcos and governmental institutions) promoting a better collaboration between all these actors for a coherent expansion of the Internet infrastructure. For that purpose, CAIDA is maintaining a large Internet measurement tool known as **Archipelago** (Ark) [Hyu06]. Through active probing, Archipelago helps the members of the CAIDA alliance to have a good knowledge of the current state of the Internet topology. The Archipelago tool is often used in a set of research projects to provide an up-to-date view of the Internet infrastructure at the Autonomous System (AS) granularity. Besides Archipelago, other research initiatives such as iPlane or DIMES are also dedicated to the same objective. The iPlane project [MIP+06] focuses on a daily measurement of the traffic flows transiting through the Internet infrastructure. For that purpose, iPlane proceeds to a daily summary of traffic measurements derived from Traceroute or Paris Traceroute probes. As a result, iPlane is a daily outlook of a partial, router-level, topology of the Internet. Besides, iPlane propose a representation of the Internet at the operators' point of presence (PoP) level. This representation is built from the Traceroute measurements collected by iPlane. iPlane clusters the various IP addresses appearing in the Traceroute measures using the DNS entry associated with those IP addresses. DIMES [SS05] is another effort from the research community to study the structure and topology of the Internet. This distributed research project uses crowdsourced measurements performed by software agents run voluntarily. The DIMES agents run Traceroute and ping regularly between one another. The results are collected by the DIMES project to constitute a similar dataset as iPlane. From DIMES's dataset, Feldman [FS08] proposes a method to automatically generate Internet PoP-level maps. This method is based on an analysis of the delay observed in the DIMES Traceroute measurements between two IP addresses. Feldman applies his method to determine the PoP-level structure of the 100 largest ASes, and to observe its evolution week after week. Since 2010, another consortium known as RIPE NCC provides a new probing tool: RIPE ATLAS [RIP10]. It consists of roughly 9000 probes voluntarily hosted, attached to several networks around the globe.

The router and AS-level views proposed by the research community so far both have shortcomings. The typical AS-level Internet graphs represent each AS as a single node. This is hiding the diversity inside large ASes. On the other hand, router-level graphs detail the Internet topology to reveal both long distance links between ASes' geographical points of presence (PoP) and short distance links within each PoP. Since ASes adopt highly redundant connectivity architecture within their PoPs to ensure resiliency within the PoP, it is sufficient to be able to discover only long distance links between PoPs to reveal intra-AS path diversity. Unfortunately, existing PoP-level Internet representations revealed in our study of the state of the art both have shortcomings. The iPlane PoP-level Internet representation is far too detailed to be considered as a correct PoP-level maps: a comparison of iPlane's data with actual research network, CSP or ISP topologies (Géant, Amazon, IIJ) stresses that fact. Besides, even if the method used by Feldman is interesting, data from the DIMES project are not available nor updated anymore, and the latest data are too old to give an accurate view of today's Internet. Thus, in our work, we aim at building an up-to-date PoP-level Internet graph.

On the Internet, the relationships in the inter-AS graph are not symmetric. This is due to the commercial nature of peering or transit relationships between autonomous systems (AS). Gao and Rexford [GR00] have expressed the effect of this asymmetry as a simple routing model known as *Valley-free*. The *Valley-free* model applies to customer-provider, peer-to-peer or provider-customer logical links. The *Valley-free* model has been questioned since some route announcements violating the *Valley-free* principles have been observed [QMM07]. This routing model is also criticized for not encompassing the complexity of the relationships between ASes (Automous Systems) in different tiers [RWM+11]. To apply Gao's routing model, it is necessary to infer the type of the Inter-AS relationship. Several algorithms have been proposed to infer this relationship, based on the relative position of ASes in paths [Gao01], on the partial views of the AS graph at different vantage points [SARK02], [OPW+10] or on heuristics applied to a simplified AS graph [DBEH+07]. Given the importance of inter-AS relationships, it is necessary to take this information into account to get a view on routable paths in the Internet.

2.3.2 Characterizing and measuring Internet resilience

One of the initial objectives of the design of the Internet infrastructure is to make the network topology resilient to nodes' failures and fiber cuts. As it is widely admitted, the probability that multiple events of that type occur at the same time is negligible, excepted in the case of natural disaster (for instance an earthquake). In the context of Cloud environment, which is the topic of this thesis, Internet resiliency consists in rerouting the IP packet flows concerned by the failing link or the failing node onto a

backup route. Such a rerouting must be achieved in a delay that remains compatible with the Quality of Experience (QoE) expected by end-users. This principle led to the decision to locate preferably the routing intelligence at the edges of the network. As the Internet grew from a handful of nodes in the 1970's to the scale it has today, some researchers have worked on the characterization of Internet's resiliency. In [TMSV03a] and in [TMSV03b], Texeira *et al.* has dealt with this issue. These authors characterize the resiliency of Sprint's ISP network as well as of generated ISP network topologies in using the Rocketfuel approach [SMW02] at the Point of Presence (PoP) level. A PoP is a geographical location where a given ISP deploys a set of routers. This work also introduces the notion of PoP-disjoint and link-disjoint paths. ISP networks performance in terms of failure recovery are quantified by means of the cumulative distribution function (CDF) of the number of alternative paths that can be found between PoPs. While this work assesses the importance to look at the PoP level to determine the properties of a network, it is limited to the scope of single Autonomous Systems (ASes). In other terms, the case of multi-ASes configurations is out of the scope of this thesis. Later, in [RS11] and [RJS14], Rohrer *et al.* proposed another method to evaluate path diversity: the "path diversity function". According to the authors, classical graph theory metrics such as the diameter, the node degrees or the node centrality cannot describe accurately the resiliency properties of a graph. To address the limits of those classical graph metrics, the path diversity function is used to compute a set of diversity scores characterizing path diversity at the AS level in the Internet. This diversity score is comprised between 0 and 1. It can be used to characterize the diversity of two paths, of a set of paths between two nodes or of all the paths within a whole graph.

2.3.3 Finding possible paths in an Internet graph representation

In order to evaluate the path diversity between two points in the Internet, it is necessary to be able to find all the paths in this graph that comply with the *Valley-free* routing policy in a directed graph representing the Internet. Indeed, as it has been highlighted by Erlebach *et al.* [EHM+06], working on an undirected Internet representation does not render the complexity of BGP routing between ASes. In this article, the authors also demonstrate that finding vertex-disjoint valley-free paths in a directed AS-level Internet graph representation is a NP-hard problem. Through a graph transformation, the authors manage to reduce the complexity of the problem and to find a polynomial time solution to this problem. Klöti *et al.* [KKAD15] acknowledge the complexity of this problem, and suggest another graph transformation methodology to find valley-free AS paths in the Internet. While the results presented these two previous works are interesting to determine diverse paths at the AS level, it is difficult to apply the graph transformation described in those two research projects to a PoP-level Internet graph. In PoP-level graphs, the presence of intra-AS relationships makes it difficult to

transpose the graph transformations presented in the articles.

2.4 Internet exchange state of the art

For a long time, the exact role and impact of IXPs on the Internet was unknown. IXPs correspond to dedicated places in the Internet where ASes agree to peer to exchange traffic on the peering basis if they find a mutual benefit. This peering assumes in general a compensation business model, where operators compensate the imbalance in exchanged traffic. In the recent years, some studies have focused on determining where the IXPs are located [AKW09] or who is a member of those IXPs [KAK+16]. In [ACF+12], Ager *et al.* take the example of a large European IXP to show that the role of those exchange points in the Internet is under-estimated. They show that ASes tend to peer more than what route servers or other macroscopic data tend to show. Besides, the IXPs and the proximity they allow between ASes play an important role in the flattening of the Internet shown in [DD10], [GILO11] or [GALM08]. Indeed, those articles show that the Internet is a less pyramidal construction than what was previously envisioned. They show that Content Delivery Networks (CDN) or large content providers are present at those IXPs to peer directly with regional Internet Services Providers, bypassing large Tier-1 operators.

From a resiliency perspective, the role of IXPs to increase path diversity has been highlighted in [HCC+12] or in [CHW+11]. In those articles, the authors suggest that relaxing *Valley-free* routing policies temporarily at peering points can help some networks in the recovery of their connectivity after a natural disaster or a node failure. Both studies use an asymmetric model of the Internet, but in the topological model they consider, the Internet graph is modeled at the AS level.

Besides, the idea to use IXP's neutral locations to steer traffic between endpoints has emerged. The use of IXPs to deploy new services is made possible by projects such as the Software-Defined Internet Exchange(SDX) [GVS+14]. SDX combines traditional peering using the Border Gateway Protocol (BGP) with the use of a SDN controller to support elaborated peering use cases such as application-specific peering, inbound traffic engineering or traffic redirection through middleboxes. In that extend, SDX allows several actors such as ISPs, CSPs or transit operators to control the way their network traffic is managed at the IXPs. Stepping on this project, proposals made in [KKR+15] and in [KKR+16] suggest using vantage points located at the IXPs to steer the way traffic is routed from one node to another in the Internet, thus controlling path resiliency or quality of service. In this proposal, the authors make the assumption that when an AS is present at two IXPs, then a "pathlet" exists in this network between those two IXPs. This pathlet can be associated with specific QoS properties guaranteed by

the operator. The use of pathlets between IXPs is a major difference with the design we adopted in Kumori where best effort packet transfer is assumed between the nodes of our overlay.

From an economical perspective, the neutrality of IXPs with regards to large ISPs and the possibility to use technological innovations like SDX can be used to introduce a new actor in the Internet connectivity ecosystem: the Overlay Services Provider (OSP). This type of actor as been introduced in [ZDA07], where Zhu *et al.* show the economical possibility for an OSP to provide a better connectivity service than traditional ISPs in terms of QoS. This new actor provides a better network QoS by positioning multihomed routers at IXPs and by dynamically selecting the best operator between the routers. This economical space shows potential applications and perspectives beyond research for the work presented in this thesis.

2.5 Open topics

The state of the art we presented in the previous chapters shows some voids between the objectives we pursue in the design of the Kumori architecture and the possibilities offered by available research work.

First of all, traditional solutions to ensure resiliency through path protection and guaranteed SLAs require a control over the routing elements constituting the network. In that extend, they can only be used to secure inter-datacenter connections network connectivity providers operating infrastructure at a large scale. Those solutions are often proposed to CSPs to help them secure the interconnectivity of their infrastructure. In such a case, the CSP does not have a grip on the operation of the resiliency strategy. They can only make sure that the penalties associated with the impossibility to protect some paths against failure are sufficiently dissuasive. In our view, this lack of operational control by the CSP is a major drawback of traditional resiliency solutions.

CSPs can have a better control over the path taken between the nodes constituting their infrastructure using overlay networks, multihoming strategies or multipath techniques. Those concepts suffer from some lacks. First of all, they are deployed at the edge of the network, often at the CSP's datacenters. Those datacenters may not benefit from a rich Interconnectivity ecosystem from which they could choose their network provider, depending on the datacenter's location. This limited connectivity of the overlay nodes limits the richness of the paths that can be taken between the nodes in the overlay. In Kumori, we would like to use overlay nodes placed at IXPs in order for those nodes to benefit from a large choice of ISPs. We expect that this will result in a larger path diversity, and then, in a better resiliency.

In most of the overlay techniques we have presented above, the routing logic is distributed. This design decision comes from the "intelligence at the edge" principle of the Internet. Yet, it can result in the difficulty to reach global utilization optimums for the connectivity of the overlay nodes, while a centralized control could help tackle those inefficiencies. As shown in recent works associated with the rise of Software-defined networking, this network control centralization allows a range of interesting use cases that we would like to consider in our Kumori architecture.

Chapter 3

The Kumori architecture

Contents

3.1 Design objectives

In this thesis work, our main goal is to replace the private link connectivity framework used by most Cloud Services Providers (CSP) to connect their distant datacenters and presented in section 2.1.1 by an overlay architecture. This overlay architecture takes advantage of the fact that most datacenters are attached to several ISPs to get access to the Internet. With such an approach, the cost of setting up or maintaining resilient connections between datacenters shall be reduced. Besides, using this architecture shall accelerate setting up the connectivity for a new datacenter.

Using our architecture, we want to achieve the same level of resiliency as in a dual private link strategy, or even perform better. Typically, we target a rerouting of the data traffic quicker than BGP. Currently, BGP achieves, such a rerouting in less than a minute. While our ideal objective would be to approach the performance level of Fast Reroute in MPLS, we want to be able to react to a failure in a few seconds.

In the methods used today to provide resiliency to datacenter connections, the network operator is in control. In practice, the CSP can only bet that the operator will respect its Service Level Agreement (SLA) contracts without any technical lever to react to an observed failure. With our architecture, we want to give the CSPs the control of their network connectivity. Thus, our architecture needs to be as independent as possible from the network operators. This translates in requirements regarding the information exchange between the CSPs and the Internet Services Providers (ISP) or the placement of the nodes constituting our overlay.

Our last goal is to allow applying resiliency strategies per flow or per application deployed in the datacenters since every network traffic in a datacenter is not equal. CSPs may want to apply different resiliency strategies to flows depending on their priority, on the application requirements or on their loss of revenue in case of failure. Designing those policies is beyond the scope of this paper, yet we need to provide methods to allow per-flow resiliency strategies. The proper application of routing policies on a per flow basis is coupled with a wish to control the network flows from server to server. This involves being able to control those flows within the CSP datacenters and outside those datacenters in the Internet.

To summarize, we are trying to achieve the following objectives with our Kumori architecture:

- Give CSP operational control over their connectivity;
- Provide a lower cost solution compared to the establishment of dual private links or MPLS circuits between datacenters;
- Deploy more quickly than traditional resiliency strategies;
- Control network traffic resiliency dynamically and on a per flow basis;
- Control the network traffic from server to server, inside and outside the CSP datacenters.

3.2 Kumori architecture overview

In Kumori's design, we have translated our objectives into principles to guide our detailed design. First of all, we want Kumori to be used by CSPs independently from the ISPs they are in contract with. This requirement has led us to the use of a network overlay to control the traffic outside the CSP's datacenter. Besides, we have to use relatively neutral locations in the Internet to place our overlay nodes. To perform a dynamic control of the routes taken by the network flows in the overlay, we decided to use a central controller to steer the various nodes' behavior. This way, we avoid the

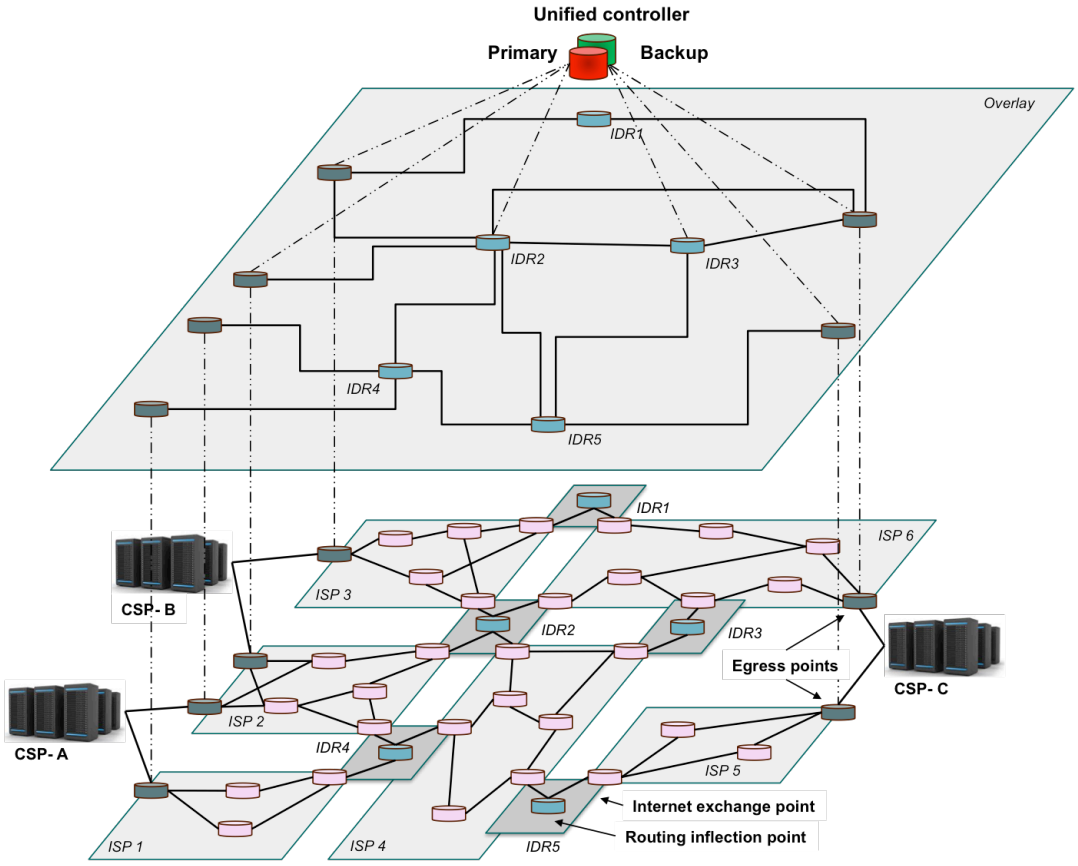


Figure 3.1 – Inter-datacenter architecture connecting three datacenters including the egress points, the routing inflection points and the unified controller

design or reuse of a distributed control protocol for which the convergence time might be problematic, and we centralize the routing decisions to enhance the CSP's control. At last, our wish to steer traffic on a perf low basis from server to server has led us to design Kumori as a two-fold architecture. Some elements are used within the CSP's datacenters while an overlay is used between the datacenters. In the following sections, we present each aspect of the Kumori architecture.

3.2.1 Inside the datacenter

Inside each datacenter, we control the path taken from the servers to the egress point using either Software-Defined Networking (SDN) or Segment Routing, depending on the technologies used inside each datacenter. As presented in 2.1.2, SDN is a technology allowing a central controller to influence or update the routing policies applied by a set of networking equipments through an open interface. In order to use SDN inside a CSP's datacenter, those equipments must allow this control via a communication protocol. If it is the case, then Kumori can use the interfaces to those

equipment's control plane to steer the network traffic from the various servers in the datacenter to an exit node. SDN is a relatively new technology, and it is far from being deployed in a majority of datacenters today. Besides, the equipments deployed today can not be updated easily to support SDN as the possibility to influence an equipment's control plane in real time from an external interface was not planned in those equipment's design.

In order to steer network traffic flows within datacenters that are not ready to use SDN, our intention is to use Segment Routing (SR). SR is currently under standardization by the IETF. It is an implementation of source routing principles. Under SR, an originating node can specify which nodes or links a packet should traverse using identifiers called segment IDs. Routers exchange information about the segments they manage using a link state interior gateway routing protocol such as OSPF. In our architecture, SR can be used instead of SDN for deployment reasons. Indeed, using SDN requires that all the switches and routers in the CSP's datacenters are replaced, while it is targeted that SR will be usable on actual routers provided a firmware update. This capacity to be deployed at the cost of a software update coupled with the ability to control a packet's route from the source is one of the reasons why we consider SR as a good technology to accompany the deployment of SDN in datacenters. In SR, the list of the segments to use for a given network flow is given by a Path Computation Element (PCE) similar to the functional building block specified in MPLS. In our architecture, this role is given to the unified controller.

3.2.2 Outside the datacenter

Outside the realm of each datacenter, Kumori consists in a set of overlay nodes which are controlled by a central software element. The overlay nodes which compose our system are located at various IXPs. They can also be installed inside some large ISPs' networks or colocated with Content Delivery Network (CDN) cache nodes provided appropriate business agreements. The nodes are deployed by the CSP, thus are not under the control of any network connectivity provider. They are used to route the network traffic between two datacenters around a link failure or following a more performant route according to some QoS metrics. They learn network traffic routing rules from the central controller. They perform regular tests to assess the state of the links that connect them to some of the other nodes in the overlay. The central controller is regularly informed of those tests' results. Thus, it can determine the most appropriate routing policy among the overlay.

Datacenters are connected to the overlay nodes through a set of exit points, the egress nodes. Each egress node is associated to an ISP that connects the datacenter to the Internet. Thus, egress nodes are preferably located by the CSP close to the

router managing the connection to the ISP. In our architecture, we suggest that CSPs use at least two ISPs to connect to the Internet. Indeed, as we have seen in [AMS+03] or [HWJ08], this multihoming strategy is a mean to ensure connectivity resiliency all the way to the last hop.

3.2.3 Detailed architecture elements description

The routing inflection points

The routing inflection points have an essential role in our architecture. They are used to influence the way a given network flow is routed over the Internet between distant datacenters. Indeed, they can divert network traffic from the route advertised between the various ASes following the Gao-Rexford policy. They are coordinated by a central controller.

In Kumori, we want those nodes to be able to influence data traffic routing while keeping control over them. To make it possible, our routing inflection points are located at various IXPs. Indeed, IXPs are neutral locations in the Internet where we can benefit from a very rich AS connectivity. There, we can place a hardware platform and negociate peering agreements with other networks. Besides, the routing inflection points can alternatively be placed at private peering facilities provided by large ISPs or in large ISPs caching server racks. Placing routing inflection points there may allow us to influence the way the traffic is routed within large ASes networks. Placing routing inflection points at those location is possible provided that we keep the ability to control the routing decisions performed by those inflection points.

The routing inflection points make regular measurements on the links that connect them together to detect failures. We want to use non-intrusive methods in order to test link availabilities within the overlay. The routing inflection points regularly inform the central controller about the tests' results. Then, this information can be used by the controller to make appropriate changes to its routing policy.

Between the routing inflection points, we want to avoid using encapsulation or tunneling. Indeed, we want to reduce overhead in case several nodes are used to divert the traffic. Instead, we want the routing nodes to use a SR header in case the traffic is carried out over IPv6 or a matching filter rule to identify the packets belonging to a given network flow and apply appropriate routing rules. One of the problems we foresee is the potential filtering of the SR header on the path by a middlebox. In case we detect such a filtering, we will need to use an encapsulation scheme.

In many ways, the routing inflection points behave in like manner to SDN switches. Indeed, SDN switches can count packet belonging to a given network flow, match

packets against a set of header fields or rewrite packets on the fly. Besides, the information that the overlay nodes have to communicate to the central controller can easily fit in OpenFlow [Ope13] messages. The main difference between our architecture and a more classical SDN network is that our overlay nodes are not directly connected, and they are not controlled by the administrator of the network domain they are located in.

Egress node

In Kumori, the egress points are responsible for routing network traffic from the datacenters' networks to a given ISP and vice versa. We choose to associate each egress point with a single ISP to avoid creating single points of failure.

Beyond their ISP selection role, the egress points are used to bridge the SR computed inside the datacenters' networks and the coordinated overlay routing computed between the datacenters. To do so, the egress points listen to segment announcements advertised using the interior gateway routing protocol. Then, they provide this information to the unified controller. The egress points also announce segments in order to be reached properly from inside the datacenter. Besides, those advertised segments can also be associated to a routing policy that will be applied in the overlay.

At last, the egress points regularly monitor links inside datacenter's networks to assess their availability. The measurements are taken from a set of end servers to the egress points. Given the fact that the routers used inside the datacenter may not have the same counter capabilities as SDN switches, they cannot be used to test each network link. This results in a scarcer set of tests compared to what can be done within the overlay. Yet, this scarcity will not be a major issue because the datacenter's administrators often have other tools to ensure their network's stability.

Controller

The unified controller Kumori's brain. It controls the routing policy among our architecture, inside each datacenters and in the overlay. The controller is responsible for retrieving information from the overlay nodes in order to maintain the routing policy over time and make sure that the resiliency objectives are met. In that extend, it is necessary that the controller has a view both on the datacenters network and on the overlay, and that it controls the routing policy in both areas.

Our unified controller inherits properties and functionalities from both the SDN controller and the PCE in SR. Inside the datacenters, it provides the servers with a list

of segments to use to route their traffic given the requirements they have. Between the datacenters, it updates regularly the various routing inflection points with a set of forwarding rules that translates the overall routing policy.

As this central element is critical in the Kumori architecture, it can be considered as a single point of failure or, at least, as a bottleneck in our system. These concerns are associated to both resiliency and scalability issues. Controller resiliency can be achieved using classic replication methods from the IT industry. The scalability issue is more complex to solve. Several projects investigate this question in the SDN community. They adopt either federal approaches or hierarchical methods.

The Distributed multi-domain SDN controllers (DISCO) project [PBL14] is an example of federal controller distribution. In this project, controllers are responsible for areas in the network, *i.e.* clusters of network equipments that are geographically close from one another and from the area controller. Area controllers synchronize together by exchanging routing messages. in DISCO, scalability comes from the possibility to create more areas when the number of equipments in the network is growing. DISCO also addresses controller resiliency issues by configuring, for each network equipment, a backup controller besides the main area controller. This backup controller is the area controller of an adjacent area, and is contacted when the main controller becomes unreachable.

On the other hand, the Orion project [FBG+14] adopts a hierarchical approach to SDN controller scalability. In this project, controllers are organized in a hierarchical tree. Global routing policy decisions are taken at the top of the tree, and leaf controllers take more fine grained decisions. Scalability can be obtained by adding more leaves in the tree, or even by adding a hierarchical level if the number of equipments in the network grows importantly.

In the course of our work, we did not have the time to evaluate those two approaches in our context. Thus, we choose to leave the detailed study of this question for future work in our project.

3.3 Kumori and traffic steering

In Kumori, we need to use a proper mechanism to control the way network flows are routed among the overlay. The typical method to operate this control over traffic routing is to use tunneling techniques. Indeed, tunneling is widely used to abstract or virtualize networks, and several protocols have been developed: the Virtual Local Area Network protocol (VLAN) [Jef14], the Virtual eXtensible Local Area Network protocol (VXLAN) [MDD+14], the Locator/ID Separation Protocol (LISP) [FFML13], the Stateless

Transport Tunneling Protocol for Network Virtualization (STT) [DG16], the Generic Routing Encapsulation protocol (GRE) [FLH+00], the Generic Network Virtualization Encapsulation protocol (Geneve) [GGS16], or the Generic UDP Encapsulation protocol (GUE) [HYZ16]. Those protocol differ in the encapsulation header they use, in the kind of traffic they encapsulate and in their maturity. In Kumori, we would like to encapsulate IP traffic in order to tunnel it over a wide area IP network. Thus, we will favor protocols such as GRE, GUE or Geneve that encapsulate traffic in UDP or UDP-like frames.

Besides tunneling traffic from one point in the Kumori architecture to another, we need a method to route traffic among several routing inflection points if needed. Again, several mechanisms can be used to perform this routing.

First, each node of the Kumori overlay can ask the controller whenever they receive traffic flows that they do not know yet. This method is typical of Software-Defined Networking setups where the data relaying nodes ask the controller for flow entries related to the traffic they have to manage. This method has a drawback: at flow establishment, the request to the controller introduces a delay that can be quite important in a global setup. The impact of this drawback can be limited by installing flow entries proactively in the kumori nodes.

To avoid the necessity for Kumori nodes to contact the central controller on the arrival of unknown network flows, we can use the source routing principles that we already presented in Section 3.2.1. To do so, we can use the Segment routing header in the traffic flowing among the Kumori overlay. This use case has been foreseen by the Segment routing designers in [PFF+16]. To steer traffic in the overlay, the first (ingress) node asks the controller for a list of segments to be used for a given network flows. the list of segments to use to route the traffic is encoded in the form of segment routing headers in the packets to be sent. The behavior of the routing inflection points then depends on the segment they receive rather than on a dynamic flow entry they get from the controller. The major drawback of this method is that the segment routing header can be used either on a MPLS or on an IPv6 dataplane. Unfortunately, we are not sure of the possibility to use IPv6 between the various routing inflection points, even if the global IPv6 traffic is growing.

Recently, Yong and Hao [YH16] proposed an alternative mechanism to stitch tunnels and create a path between two nodes in an overlay. In this tunnel stitching mechanism, tunnels are identified by unique identifiers. Each node in the overlay maintains a table where, for each considered overlay instance, relationships between an incoming tunnel and potential following tunnels are maintained. To route traffic form an ingress node to an egress node in the overlay, the ingress node encapsulates the traffic to the desired next hop, and indicates a next tunnel identifier. At each hop, the traffic is decapsulated, the identifier of the next tunnel is retrieved according to the running overlay instance, and the traffic is re-encapsulated to reach the next hop in the overlay. This mechanism

is intermediate between the full control operated by SDN and the source routing operated in Segment Routing. The ingress node can control the path taken by the traffic it sends from end to end by choosing both the next hop and the overlay instance to use for this specific traffic. We think that this mechanism can be an interesting mechanism to route traffic among a relatively small overlay because it addresses both previous mechanisms' drawbacks. Tunnel stitching requires that the tunneling protocol allows adding an option in the encapsulation header. As Geneve [GGS16] or GUE [HYZ16] headers can be extended using type-length-value (TLV) optional fields, those encapsulation protocols will be used in the Kumori architecture.

3.4 Kumori and resiliency

The main use case for Kumori consists in ensuring the resiliency of inter-datacenter connections. For that purpose, we need to be able to route traffic using several disjoint routeable paths, while enabling fast failure detection and traffic redirection on an alternative path.

Our design aims at making sure that, at any time, multiple paths can be used to route network flows between two hosts located in distant datacenters. Datacenters are connected to the global Internet through several ISPs. This multihoming strategy prevents the datacenters from relying on a single ISP, thus on a single point of failure. Inside the datacenter, the servers and the various egress points are connected through multiple paths. In the overlay, our goal is to set up enough nodes to make sure that the datacenters can be reached using multiple paths. To ensure path disjointness, we aim at taking the network's topology into account in the choice of the nodes and their placement, in a similar way as in [HWJ08]. In Chapter 6, we will present a method to evaluate the PoP-level path diversity that Kumori allows to access, as well as the gain provided by the architecture compared to routing inter-datacenter traffic over the plain Internet.

The egress points and the routing inflection points regularly monitor the quality and the metrics of the network flows passing through them. In the past years, several projects have investigated link monitoring in SDN networks. The OpenNetMon project [vADK14], for instance, presents a method to monitor link state at edge switches using the OpenFlow message exchange. In this method, Van Adrichem *et al.* uses the fact that flow entries have a limited time validity to allow the SDN controller to retrieve network flow statistics. The SDN switches send those statistics to the controller using *FlowRemoved* OpenFlow messages. The OpenNetMon project also proposes an adaptive timing scheme to poll the network flow statistics accurately while avoiding generating too much overhead when the situation is stable. In the Payless

project [CBAB14], Chowdhury *et al.* also use the OpenFlow protocol to exchange network traffic statistics at an adaptive pace.In the Kumori architecture, using a similar network monitoring method will allow the central controller to detect abnormal traffic conditions. As presented in both the OpenNetMon and the Payless projects, the capacity to detect failures quickly using a passive monitoring strategy is correlated with the frequency of the feedback provided to the controller by the nodes. In those two project, an adaptive polling mechanism balances the wish to react quickly with the need to avoid overloading the controller or mistaking a measurement artifact for a node or link failure.

To confirm the passive detection of a failure event or to detect those failures more quickly, other projects prefer to use an active probing mechanism. In [SSC+11] and in [AAK14], passive monitoring using OpenFlow messages is completed by the use of Bidirectional Forwarding Detection (BFD) described in [KW10a] and [KW10b]. BFD is a Hello protocol used to detect faults between two nodes in a network. It is designed to keep a low overhead while detecting failures quickly. In [SSC+11], BFD is combined with a pre-computed path protection strategy to achieve carrier-grade failure recovery in an Internet Services Provider network. In Kumori, we will use BFD to confirm failures detected using passive monitoring as we do not target sub-50 ms recovery times.

In Kumori, resilience is achieved by providing a set of independent paths to reroute traffic in case of failure. Detouring the network traffic around a detected node or link failure is done in our architecture using both Segment Routing inside the datacenters and the centrally controlled overlay between the datacenters. Inside the datacenters, servers are informed of a failure by the unified controller that computes an alternative route and provides them another list of segments to route their network traffic. Between the datacenters, the routing inflection points receive a new set of forwarding rules. Using those new rules, the overlay nodes are able to react to detected failure while preserving the required infrastructure connectivity.

The unified controller plays a critical role in ensuring resiliency in our architecture. Indeed, it is responsible for assessing that a network failure occurred thanks to the measurements it receives from the various elements in the architecture. It has a global view on the overlay and on the datacenter network's topology. Thus, it is able to compute alternative routes and to make appropriate changes to the forwarding rules applied in the overlay. This alternative route computation can be done on the fly, or in advance considering a set of potential failure scenarios. In that regard, we aim at reusing concepts and developments done in the MPLS community around Fast Reroute that were presented in [RI07].

3.5 Summary

In this chapter, we presented the design of Kumori, a SDN-controlled network overlay spanning both the datacenter domains and the inter-datacenter wide area network. Inside the datacenter, the overlay is used to steer the network traffic from a server to the most appropriate egress node. Those egress nodes are the points of contact between a datacenter and an network connectivity provider used to route traffic over the Internet. Between the CSP's datacenters, Kumori consists in a set of routing inflection points placed at IXPs. There, routing inflection points benefit from a rich connectivity. Thus, they can use a proper path to relay traffic to a CSP's datacenter while avoiding detected failures. Among the overlay, the traffic routing decisions are taken by a central controller. This controller gathers information about the network state from the egress nodes and from the routing inflection point to detect failures. When such a problem is detected, it is responsible for coordinating the behavior of the overlay's nodes and for sending them traffic routing instructions.

3.6 Kumori's performance indicators

In Kumori's design, we took advantage of the lessons learned by previous research initiatives presented in the state of the art. Yet, we are not sure that this solution is beneficial for cloud services providers willing to use it to enhance their inter-datacenter connections' resiliency. Thus, we need to evaluate Kumori's properties and how it integrates in today's Internet.

First of all, we would like to determine how Kumori performs compared to well known edge-oriented overlay networks such as the Resilient Overlay Network (RON) [ABKM01]. To do so, we would like to evaluate the **path length** of alternate paths provided either by Kumori or by the RON overlay between nodes belonging to a CSP's distant datacenters. We also want to determine whether this observed performance is similar whatever the CSP using the overlay or if some factors underlying a CSP's infrastructure design have an impact on this performance.

Besides, we want to make sure that Kumori enhances inter-datacenter network resiliency for CSPs. In that regard, we want to evaluate the **path diversity** among a CSP's node pairs obtained on the Internet and through Kumori. This experiment will allow us to measure the gain provided by our architecture. We also want to perform this evaluation for several CSPs to determine if Kumori's benefits are homogeneous or differ depending on the CSP.

The maintainability and the cost of an overlay depends on the number of nodes that are needed to ensure a proper service of this overlay. While evaluating the path

performance and the path diversity gains provided by Kumori, we will also determine the optimal **number of nodes** needed in Kumori to get an optimal performance.

At last, a novel overlay architecture such as Kumori needs to be economically interesting for its users and operators in order to be deployed on the field. Thus, we will evaluate Kumori's profitability by comparing its **operational cost** to the cost of a classical inter-datacenter connectivity solution.

Chapter 4

A first performance evaluation of the Kumori architecture

Contents

4.1 Evaluation metrics

In our evaluation, we compare Kumori to other network overlays aiming to reinforce the resiliency of connections between edge nodes in the Internet. The main difference between Kumori and those projects is the location of the overlay nodes: in Kumori, the routing inflection points are located at various IXPs while in other overlays the nodes are located at the edge. For the rest of the evaluation, the RON overlay is used as the edge overlay project to which Kumori is compared.

In this comparison, we would like first to determine whether Kumori and RON can provide alternative paths with similar performance characteristics. To that extend, we consider the delay on the paths as the considered performance indicator. We compare the latency of the shortest paths accessible using both architectures. Besides, we want to evaluate the cost of deploying and operating Kumori compared to RON. In that regard, we make the assumption that this cost depends on the number of nodes

participating in the overlay. We compare the number of nodes needed in Kumori and in RON to access the same number of alternative paths.

4.2 Evaluation methodology

In this first evaluation, we have chosen to use data extracted from the iPlane dataset [MIP+06] on the 15^{th} of February 2015. This dataset has two advantages. First of all, unlike simulations, it represents actual links that were observed and measured in the Internet on that day. Even if the dataset is not representing the whole Internet, it is rather significant. Besides, unlike data extracted from route servers, iPlane can reveal transit links that are used to route actual traffic while they remain invisible in BGP tables. Those two advantages make iPlane an interesting dataset to use. Yet, some work is needed to make it directly suitable to our study.

The raw iPlane dataset takes the form of archives of *traceroute* measurements performed daily as well as some summarized datasets. Those summarized datasets gather all the inter-PoP links observed on a given day in the *traceroutes* and associate them an average of the performance metrics that have been measured. One of those summarized datasets gives the latency of the links between the routers revealed in the iPlane measurements, the loss rates on those links, the association between observed IP addresses and routers and the Autonomous Systems (AS) the routers belong to. Those observations are performed every day since the inception of the iPlane project. In our evaluation, we use data summarizing the measurements done on the 15^{th} of February 2015. Using this data, we have built a graph representing the links between the routers that could be observed on that day. We have used the Python programming language to parse the iPlane dataset files and the igraph library to build the graph and manipulate it. The graph we obtained takes the form of an undirected weighted graph with 190,028 vertices representing the routers and 916,390 edges representing the observed links between routers. We have chosen to use this observed link latency as the weight associated to each edge.

In the iPlane dataset, the evaluation of the latency between routers raises two issues with regards to our evaluation. First, if traffic has been observed on a link but latency cannot be evaluated accurately, a negative cost of -9,999 is given to the link. Yet, the igraph library does not accept negative values as an appropriate value to measure an edge cost. To solve this problem, we gave every node with a negative latency a weight equal to twice the maximum weight observed in the dataset. Second, some links have a latency equal to 0. Indeed, in iPlane, measures are rounded to the millisecond, and very fast links are given a zero cost. Yet, this cost does not take into account the switching cost at the routers. To take this switching cost into account, we give every link

with a zero cost a minimal cost equal to twice the latency measured by doing a *ping* on the loopback interface of a linux server running Ubuntu 14.04 LTS, *i.e.* 0.5 ms.

Once our undirected weighted graph obtained, we need to spot the routers belonging to the CSPs or to the IXPs in order to determine and evaluate the shortest paths between the spotted nodes.

In order to identify the nodes belonging to the CSPs, we have selected the major global providers from market data gathered by Gartner, to which we have added some interesting regional actors. The result of this first identification step is a list of 13 companies. Then, we looked after the ASes managed by those companies in Hurricane Electric's BGP toolkit [HE-]. We obtained a list of 133 interesting ASes. At last, we looked in iPlane's dataset after the nodes belonging to those ASes.

The identification of the nodes belonging to the various IXPs was more complex. Indeed, there is no centralized database of the existing IXPs, and by extension, no data about IP prefixes or ASes belonging to IXPs. Nevertheless, scarce data can be obtained from the PeeringDB [Pee], a database where network managers provide voluntarily information about their peering policy, or from Packet Clearing House [**?**], a non-for-profit research institute that operates routing measurement facilities at several IXPs around the world. We first cleaned the data we found in both databases and associated them in order to find the IP address subsets used by the various IXPs. Then, we found the nodes that were present at an IXP by using the IP to node mapping given by the iPlane dataset.

As a result of this identification phase, we identified 1,604 nodes belonging to a CSP and 2,177 nodes present at an IXP out of the 190,028 vertices in the graph. In the next phase we keep all the vertices in the graph, and we use the identified nodes to evaluate both RON and our Kumori architecture.

To properly evaluate both our architecture and RON's capacity to route traffic between routers belonging to CSPs, we removed all the edges linking two nodes associated to the same CSP from the graph we obtained by parsing the iPlane inter-router latency dataset. By removing those edges, we make sure we compare our architecture to RON rather than to the CSP's interconnection strategy. Then, we looked after the shortest paths between all the CSP routers pairs, between the IXP routers pairs and between the CSP routers and the IXP routers. With the resulting shortest paths sets, we compared the paths obtained using the RON architecture and using the routing inflection points located at the various IXPs that form the inter-datacenter part of our Kumori architecture.

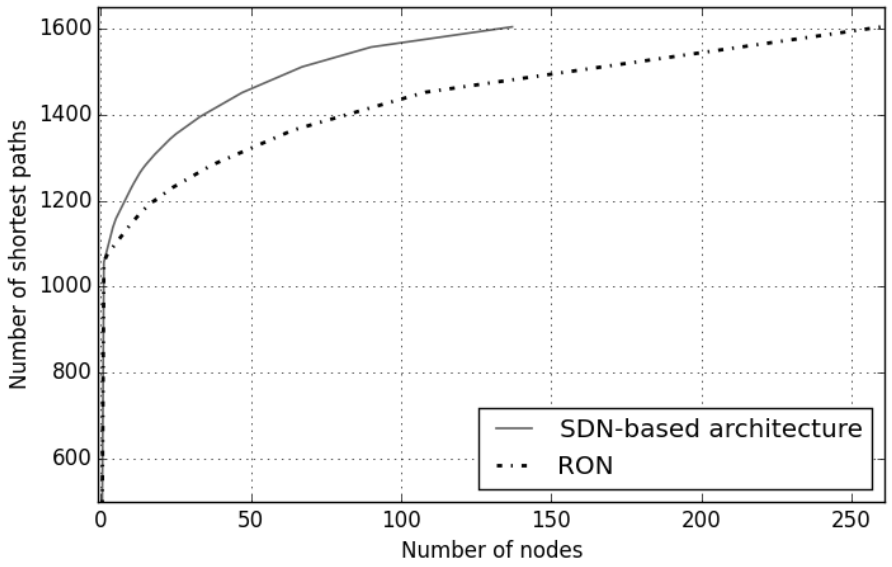


Figure 4.1 – Cumulative distribution of the number of nodes needed to access the maximum number of shortest paths

4.3 Performance results

We compared the Kumori architecture with RON in two steps: First, we considered a large, imaginary CSP federating the CSPs PoPS we identified in our evaluation set. Then, we analyzed the result for each specific CSP in our evaluation set.

4.3.1 Global results

In this section, we first look at the results we obtained considering the entire set of PoPs belonging to a CSP whatever the provider they belong to.

First, our measurements show that the paths accessible using the Kumori architecture have a smaller or equal cost in terms of latency than the paths provided by a RON overlay for 1,255,950 CSP PoP pairs over the 1,285,606 possibilities. That represents 97.5% of the cases. If we consider a strict performance improvement with regards to latency, our architecture has better performance for 73,511 CSP nodes pairs over the 1,285,606 possibilities. It thus represents a strict performance improvement in 3.1% of the cases. This result shows that our architecture has similar performance as RON in the vast majority of the case, but does not show a drastic performance improvement most of the time.

4.3.2 Detailed results / CSP

After this first evaluation, we compare the number of nodes needed in the Kumori architecture and in the RON overlay network to route data traffic between the CSP node pairs. Figure 4.1 presents a plot of the cumulative distribution of the number of nodes needed in our architecture and in RON. The plot shows that 47% less nodes are needed to use all the shortest paths in our architecture compared to RON. If we consider 80% of the shortest paths between the CSP node pairs, only 15 nodes are needed in Kumori while 38 nodes are needed in RON. As the cost of operation of an overlay network depends on the number of nodes to deploy, this result shows that our SDN-based architecture can be less expensive to deploy and operate than a RON overlay.

4.3.3 Results analysis for specific CSPs

In the results we obtained in Section 4.3.1, we considered that the PoPs belonging to the twelve large CSPs we selected belonged to the same, large CSP. Yet, those CSPs we have included in our study do not form an homogeneous group of actors, as the largest CSP in terms of PoPs in our set has roughly 200 times more PoPs than the smallest CSP. In this section, we study the gains provided by our architecture for each CSP in our set. We have also included two Cloud research infrastructures for the sake of comparison: WIDE and Géant.

In this study, we compare the Kumori architectures with a RON overlay for two metrics, and we plotted the results in Figure 4.2. On this figure, each dot represents a specific CSP. The diameter of the dot is proportional to the number of PoPs associated to the CSP. We choose to color the dots in red if the RON architecture cannot be used for scalability reasons *i.e.* when the number of nodes needed to reach all the shortest paths is bigger than 50. This scalability limit has been initially presented in [ABKM01].

First, to compare the performance of the alternative paths accessible using both architectures, we evaluate the proportion of the paths accessible via our architecture that are strictly shorter than the paths accessible via RON. This measurement is the x coordinate of the plots representing the various CSPs on Fig. 4.2. Then, to compare the cost of operation of both architectures, we compared the number of nodes needed in Kumori to access all the shortest paths with the number of nodes needed in RON to access those shortest paths. We compute the difference between the two numbers, and divide this difference by the number of nodes needed to access all the shortest paths in RON. The resulting proportion is the y coordinate of the plots representing the various CSPs on Fig. 4.2.

On the figure, we can see two groups of points: one at the right of the figure, and another in the middle at the top of the figure. The first group of points is corresponding

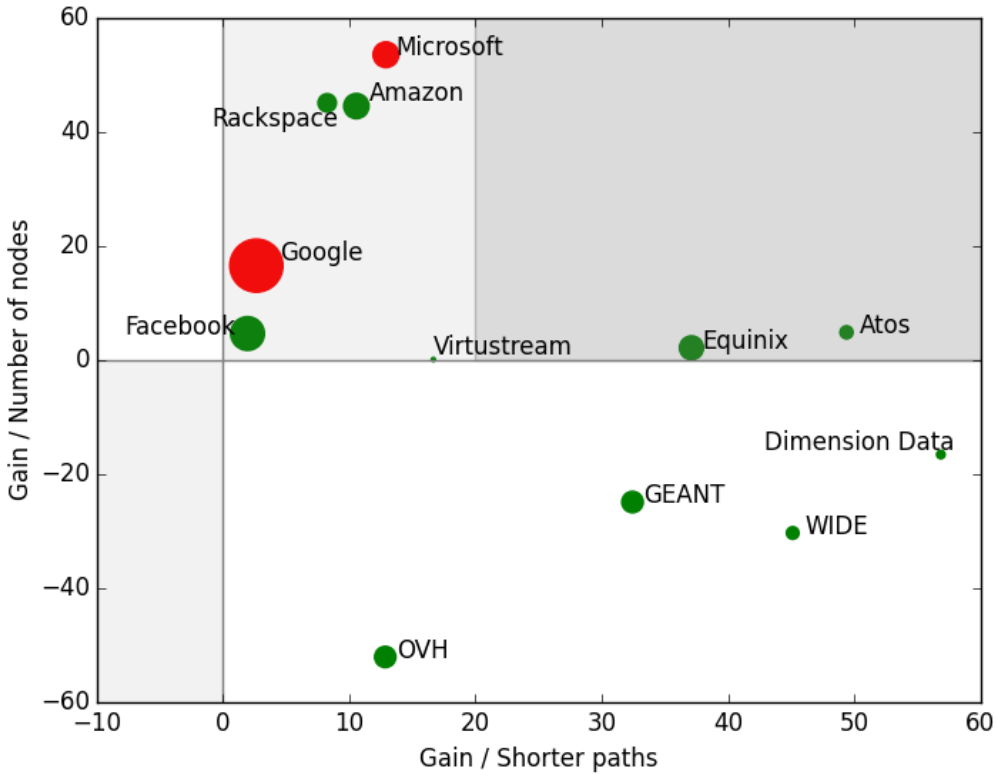


Figure 4.2 – Gain of Kumori Vs. RON in terms of path lengths and number of nodes needed to access the maximum number of shortest paths.

to the smallest CSPs in our evaluation set as well as the two Cloud research initiatives. For those CSPs, our SDN-based overlay architecture provides an access to more short paths than RON while using up to 20% more nodes in the overlay. On the contrary, the other group is corresponding to larger CSPs. Compared to RON, our architecture provides access to short paths using a lower number of nodes.

We explain those results by the difference between small and large CSPs regarding their connectivity. Large CSPs have optimized their network architecture to lower the cost to deliver network traffic to their customers. Amazon for instance proposes his customers to connect directly to his network at several points of presence through its Direct Connect offering. Our results show that the strategy adopted by those large CSPs is translated in a relative proximity of the CSP PoPs with the various IXPs. On the contrary, smaller CSPs often rely on their network connectivity provider to reach the Internet and their customers. Therefore, their PoPs are relatively farther from the IXPs than PoPs belonging to the large CSPs.

In this comparison between Kumori and RON, we can see that in every case, our architecture provides benefit on at least one dimension. The benefits that our architecture provides are quite different depending on the connectivity strategy adopted by the CSP. This result reinforces our wish to implement and deploy our overlay to

evaluate it on the field.

4.4 Conclusion

Our numerical results show that the benefits the CSPs can obtain from Kumori depend on their size. For rather small CSPs such as Dimension Data, the paths accessed via our architecture are shorter than those accessed using RON in at least 32% of the cases. For larger CSPs such as Amazon, our architecture gives access to a similar set of shortest paths between the PoPs using up to 53% less overlay nodes. Thus, Kumori gives a solution to a major scalability issue related to RON. We explain those results by the difference between the connectivity strategies adopted by the CSPs. Large CSPs are well interconnected at IXPs to optimize their network costs while smaller CSPs still depend on large ISPs to connect their DCs to the Internet.

While this first evaluation sheds some light on the potential of the Kumori architecture, the results we obtain regarding the relative performance or number of nodes required to use Kumori in comparison with the RON overlay are not sufficient to give us a proper measurement of the resiliency gain provided by Kumori. Yet, enhancing the resiliency of inter-datacenter connections is a key objective of Kumori. Besides, the graph on which we have performed our measurements is undirected, while the commercial nature of the relationships between Autonomous Systems (AS) in the Internet results in a need to represent the Internet as an asymmetric graph. We will see in the next chapters of this thesis how we managed to address those limits to properly qualify the resiliency gain provided by Kumori.

Chapter 5

Building a PoP-level representation of the Internet

Contents

5.1 Problem statement

The first evaluation presented in Chapter 4 has outlined the benefits of the Kumori architecture. These benefits depend on the topological characteristics of the network of each CSP, and of its connectivity to the Internet. Thus, we have observed that the Kumori architecture enables to reduce end-to-end path length in number of hops. For other CSPs, the Kumori architecture provides similarly performing paths with a reduced number of nodes than edge-oriented overlays. Beyond a reduction of the transit delays through CSPs' networks, our goal is also to evaluate the benefits of the the Kumori architecture in terms of network resilience. In our approach, resiliency is obtained by allowing a CSP's node pairs to benefit from a variety of paths. At evidence, offering an higher path diversity favors an increased resilience of the network in case of links or nodes failures. In this matter, node failures are systematically more constraining than link failures since they can be assimilated to simultaneous distinct link failures. In case

of a connectivity failure, intuitively, the more diverse paths, the more possibilities to route around a link or node failure.

Previous works have investigated path diversity on the Internet at the AS level or at the PoP level simply within a single ISP. In our own evaluation, we wish to evaluate path diversity among CSP's nodes at the PoP level. We justify this objective by the observation that ASes can be very diverse in term of scales, regional presence or peering policies. In our point of view, considering AS paths traversing a regional ISP's network or a Tier-1 operator's network as similar is very simplistic because inside those two ASes, the PoP-level path diversity can be very different.

In order to evaluate path diversity between the nodes of a CSP, we need a PoP-level directed Internet graph. Since such a graph with up to date data cannot be found in previous works, our first objective has been to build a PoP-level representation of the Internet. Given the importance of Internet eXchange Points (IXPs) which accommodate a growing share of the total inter-domain network traffic, we need to take them into consideration. We first determine the set of participants at each IXP. Then, we locate those IXPs in the Internet graph to connect them to the right PoPs in the different participant ASes. For the ISPs themselves, we determine what are their PoPs, how they interconnect to form their backbone, and how PoPs from different ISPs connect between each other.

5.2 Building a PoP-level topology

To build the expected PoP-level topology, we rely on four existing data sources:

- The iPlane router-level Internet graph [MIP+06],
- A directed AS-level Internet graph representation retrieved from the DRAGON project [SVLR14],
- The IXP membership data from PeeringDB [Pee],
- The IP geolocation information provided by MaxMind in the GeoIP2 database [Max].

The building process of our PoP-level topology consists in four steps. First, we sanitize the data we retrieved from iPlane to repair misleading mappings between IP addresses appearing in the data set and their associated AS (*cf.*, Section 5.2.1). Then, looking at AS membership data for each IXP, we determine which router from member ASes is present at each IXP (*cf.*, Section 5.2.2). In a third step, we cluster routers together to reconstitute each AS's PoPs (*cf.*, Section 5.2.3). Finally, we associate the

results obtained from the three previous steps to build the PoP-level directed Internet graph we will use in our evaluation (*cf.*, Section 5.2.4).

5.2.1 Sanitizing the iPlane Data Source

The iPlane project [MIP+06] produces a set of data obtained from traceroute measurements achieved daily between several vantage points. Those traceroutes are filtered and processed to produce a summary of the relationships between routers in the Internet. The routers in the iPlane dataset have several interfaces that appear in the traces. The daily data summary from the iPlane database gives the average latency on the links between routers, and the loss rate between those routers. Besides, every two months, the iPlane project produces an IP to AS mapping to associate each router to the AS it belongs to. From this dataset, our initial goal has been to build an undirected graph of the Internet at the router level from the daily iPlane data summary. In order to get a sufficient coverage while keeping relatively fresh data, we have built this graph from daily data summarizing traceroutes performed between the 13^{th} of June and the 14^{th} of July 2015.

We have then noticed some inconsistencies in the data extracted from the iPlane dataset. For instance, we have noticed that the routers belonging to AS 3303 account for 8% of the routers in the dataset, while there was no reason for the ISP using this AS number to appear with such a larger number of routers. Besides, the interpretation of some AS to IP prefixes associations in the iPlane data has appeared ambiguous. In order to be able to reconstitute a correct router-AS mapping, we have then taken the initiative to pre-process the iPlane dataset in the light of information provided by the Hurricane Electric's BGP toolkit [HE-].

After this step, we have obtained a router-level representation of the Internet with **417,638 vertices** representing the routers and **7,687,300 directed edges** representing the links revealed by the iPlane measurements between those routers. Besides, we also obtained a proper association between each router and the AS it belongs to. Figure 5.1 shows a schematic representation of the topology obtained at this stage for a small set of fictional ASes.

5.2.2 Inferring IXP Membership

In [ACF+12], Ager *et al.* underline that the role of IXPs in today's Internet is not properly considered. They have pointed out that the number of peering links at those IXPs is far more important than what route servers monitoring may suggest. We would like to take into account the importance of IXPs in our PoP-level Internet topology. To do so, we

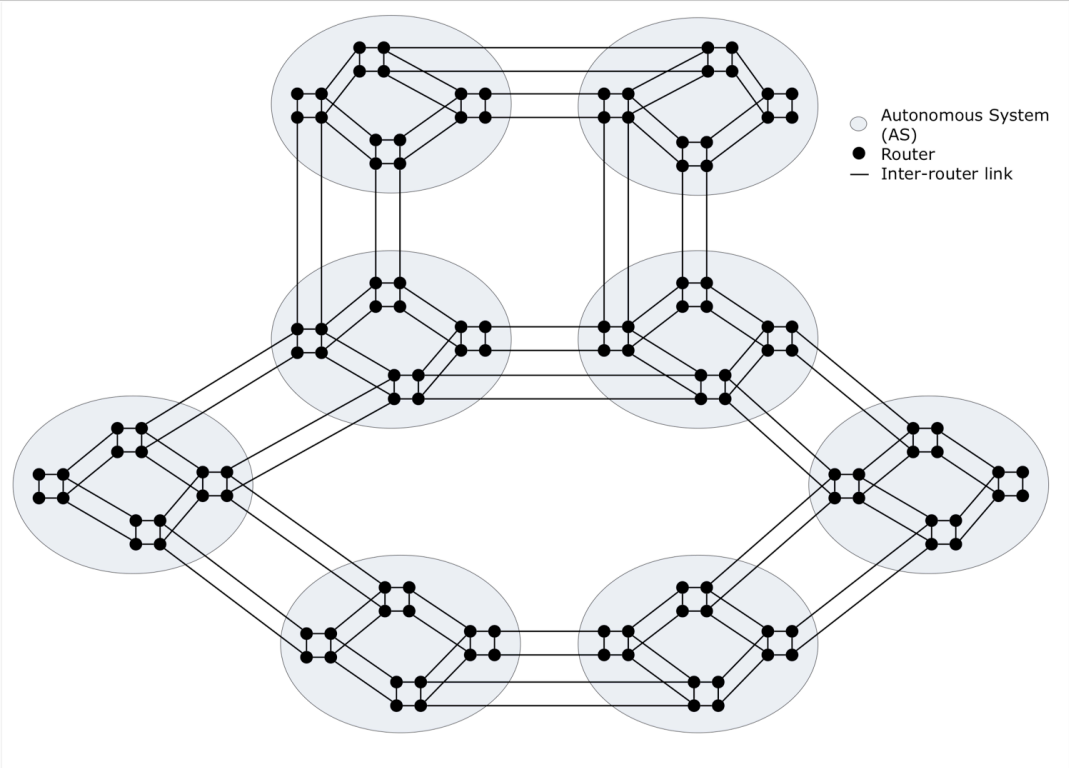


Figure 5.1 – Schematic representation of the topology obtained after sanitizing the iPlane dataset.

also need to locate the IXPs in the router level topology we have built using the iPlane dataset.

To this end, we first use the information provided by the PeeringDB database [Pee]. For most IXPs, the PeeringDB database provides information about the IP address ranges used by the network elements connected at each considered IXP. So, in a first step, we have parsed the iPlane dataset to determine which routers are associated to an IP address belonging to an IXP's address range. Afterwards, by comparing the list of routers at each IXP and the list of AS members of each IXP, we have observed that we only had very partial information about router IXP membership. Indeed, we have found a corresponding router for only 3.5% of the total number of AS memberships declared to IXPs in the PeeringDB database.

This observation leads us to apply two other methods to associate routers to each IXP. First, for each IXP, we look at the list of member ASes to determine those for which we cannot associate a participating router. We then consider all the routers from those ASes to find those that are connected to a router participating in the IXP. This procedure allows us to associate more routers to each IXP: at the end of this operation, we find a corresponding router for 45.4% of the total number of AS memberships declared to IXPs in the PeeringDB database.

At this stage, for several IXPs, we still have AS members with no router attached to the IXP. To solve this issue, we use a last technique to increase the router-IXP association. In this last step, we use geolocation information provided by the Maxmind's GeoIP2 database [Max]. Using the IP address range information from the PeeringDB database, we determine the location of each IXP according to the GeoIP2 database. For the ASes for which we are missing a router membership, we are able to determine which router is closer to the IXP. Routers located further than 30 kilometers away are not considered as IXP participants.

At the end of this three stages process, we obtain a list of routers connected to each IXP. In this list, we are able to reconstitute 62.1% of the total number of AS memberships to IXPs declared in the PeeringDB database. We can explain this partial membership information by the fact that we have tried to rebuild membership information for all the IXPs listed in the same PeeringDB database, regardless their importance or the region where they are located. A similar study made by Lodhi *et al.* [LLD+14] outlines comparable membership accuracy results in average. This same study demonstrates that IXP membership information is less accurate for smaller IXPs or IXPs in developing regions. Figure 5.2 shows a schematic representation of the topology obtained after the IXP membership determination work we have done.

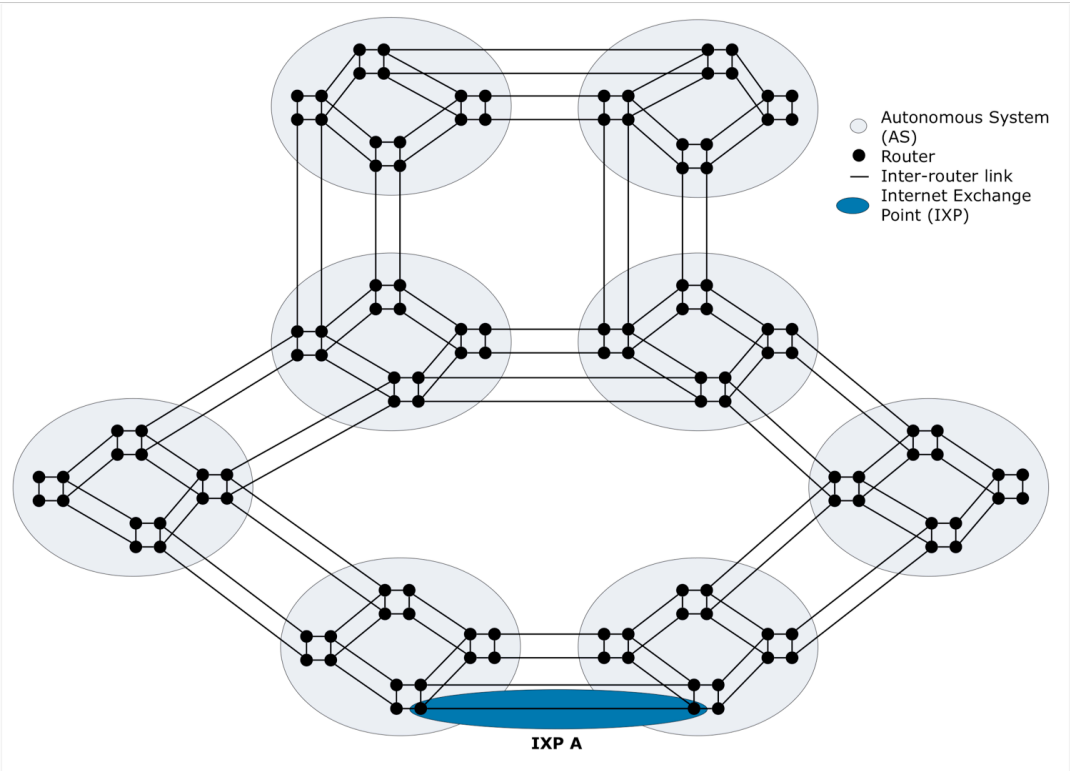


Figure 5.2 – Schematic representation of the topology obtained after determining routers' IXP membership.

5.2.3 From Routers to IXPs and PoPs

At this stage, we have a graph representing the Internet at the router level, with an enriched list of routers participating in the various IXPs. This Internet representation details every inter-router links considered in the iPlane dataset. We have observed for some networks (Sprint, Amazon, IIJ or Géant) that the number of routers in our topology is more important than the number of geographical PoPs declared by those networks. Indeed, ASes deploy several routers at each of their PoP to ensure those PoPs' resiliency to local failures. At a large scale, the failure of a single router in a PoP does not have any impact on the global resiliency of an AS, except if this failure is correlated with other failures at the same PoP. Thus, considering the Internet topology at the PoP level is sufficient to characterize intra-AS path diversity at a large scale. The complexity of the algorithm we are going to use to analyze path diversity in the Internet grows with the number of edges and nodes in the associated graph. Working at the PoP-level also enables to reduce the runtime of the algorithm without significant impact on our path diversity assessment.

To reconstitute the PoPs of the various operators, we clustered routers together according to their relative proximity. Multiple techniques exist in graph theory to cluster vertices together in a graph. In order to choose the most appropriate technique, we compare the clustering algorithms implemented in the igraph library [igr]. We then use these algorithms to cluster the routers belonging to IIJ, Amazon and Géant in our topology. The comparison we perform shows that two algorithms tend to give results that are close to our ground truth: the Infomap algorithm [RB07] and the community random walk algorithm [PL05]. The good results obtained using those two algorithms can be justified by their design principles of both algorithms. The Infomap algorithm is clustering vertices together in order to optimize the *map equation* of the graph, a mathematical construction designed to capture the underlying structure of the graph. In our graph, this underlying structure is underlaid by the design patterns directing the architecture of the routers at the PoPs. The community random walk algorithm is based on the fact that a random walker in the graph tends to spend more time in dense communities. In our graph, those communities are constituted by the routers that are tightly connected at each PoP. Given the ability of both algorithms to cluster IIJ, Amazon, and Géant's routers into meaningful PoPs, we have decided to run them for each AS in our topology to cluster routers together. In case of a diverging result, we choose to keep the most detailed PoP representation for each AS.

After this step, we obtain a graph representing the Internet at the PoP level. This graph is made of **148,926 vertices** representing the PoPs and **1,041,271 directed edges** representing the inter-PoP links. Figure 5.3 shows a schematic representation of the topology obtained after clustering the routers together to retrieve PoPs for each AS.

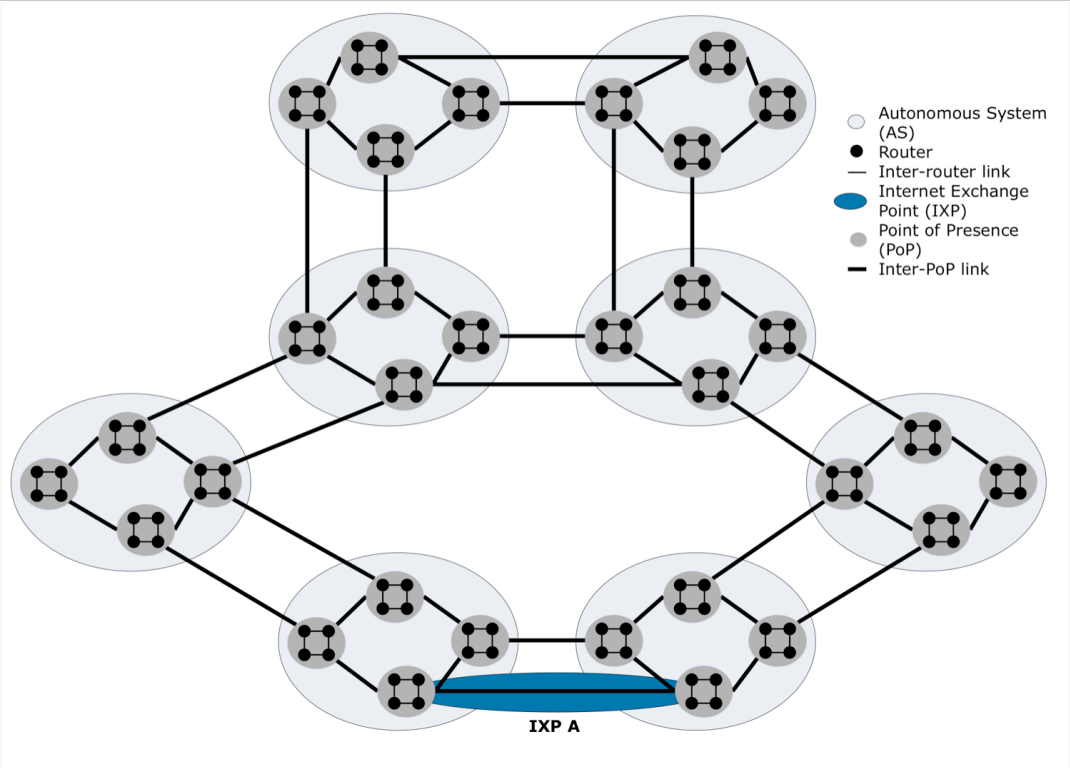


Figure 5.3 – Schematic representation of the topology obtained after clustering routers into PoPs.

5.2.4 Considering BGP policies

In the last step of our graph building process, we need to tag the directed edges between the PoPs to take into account the asymmetry of inter-AS relationships in the Internet. The topology from the DRAGON project [SVLR14] gives us necessary information regarding the edges linking PoPs that do not belong to the same AS. In DRAGON, a directed edge between two ASes can be tagged as *customer-to-provider* ($C2P$), *provider-to-customer* ($P2C$), *peer-to-peer* ($P2P$) or *unknown* (UN). We apply the same tags in our graph to the inter-AS PoP links. After this step, the edges within a single AS remain untagged. We choose to tag them as *internal* (IN). These edges will be treated differently (*cf.*, Section 6.2). Figure 5.4 shows a schematic representation of the topology we obtain after tagging the inter-PoP links with the appropriate relationship. This topology will be used in the path diversity evaluation presented in Chapter 6.

5.3 Conclusion

At the end of the multi-step process described in this chapter, we obtain a PoP level Internet representation taking the form of a directed graph. In this graph, the directed edges linking the PoPs are tagged according to the type of relationship between the AS

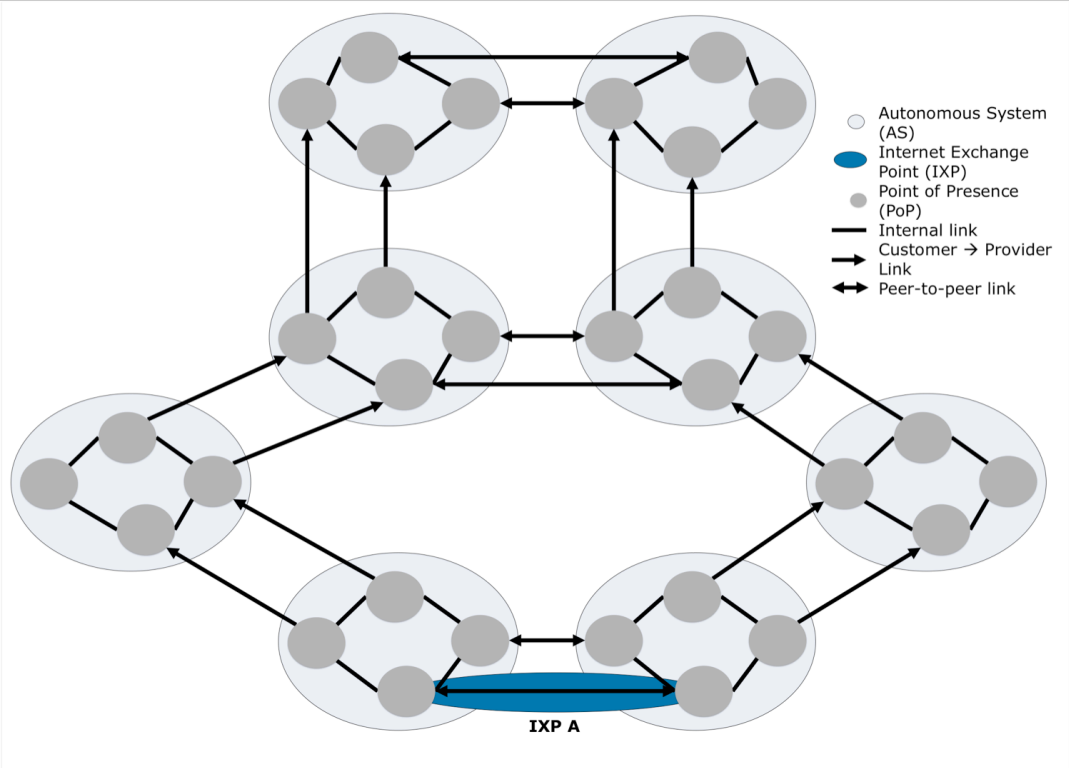


Figure 5.4 – Schematic representation of the topology obtained after tagging the inter-PoP links.

they belong to in the Internet. IXPs are represented as logical nodes connected to the PoPs in which their member routers were clustered. Given the data from which this graph has been built, it is an accurate representation of today's Internet at the PoP level.

The resulting graph obtained is constituted with 148,926 vertices and 1,041,271 directed tagged edges. This graph is scarce, as indicated by its density which is equal to $4.69.10^{-5}$. The average degree in this graph is equal to 13.98, and its diameter is equal to 12.

Working with a PoP level representation of the Internet to measure path diversity is challenging given the large size of the graph. Yet, manipulating this graph is more feasible than working at the same router level as the iPlane topology. Besides, evaluating path diversity at this level of granularity is necessary to take into consideration the diversity of the autonomous systems in the Internet. Indeed, AS level representations of the Internet tend to hide the differences between large tier one Internet services providers and more regional or local providers.

Chapter 6

Path diversity evaluation

Contents

6.1 Evaluation presentation

In this chapter, we present our evaluation of the resiliency benefits associated with the use of the Kumori architecture. With this architecture, we want to increase the resiliency of inter-datacenter communications by increasing the number of routable paths than can be taken in the event of a node or link failure on the preferred path. We thus aim at comparing the number of paths made available by the Kumori architecture in various configurations with the number of available paths on the Internet.

In this evaluation, we are using the PoP-level, asymmetric Internet representation we presented in Chapter 5. We design and develop an algorithm to find disjoint paths between PoPs respecting the Gao-Rexford routing policy in this graph representation. With this algorithm, we look after four types of diverse paths between two PoPs:

- **Direct edge-disjoint paths**: Those paths between two PoPs are not sharing any edge, while they might share one or several node. We describe them as direct because we do not use Kumori to access those paths. The number of direct edge-disjoint paths between two PoPs is the number of available paths between those two PoPs on the plain Internet.

- **Direct node-disjoint paths**: Those paths between two PoPs are not sharing any node. Thus, by construction, they are not sharing any edge either. We describe them as direct because we do not use Kumori to access those paths.

The number of direct edge-disjoint or node-disjoint paths between two PoPs is the number of available paths between those two PoPs on the plain Internet.

- **Kumori edge-disjoint paths**: Those paths are edge-disjoint paths between PoPs that cross a Kumori node located at a given IXP.

- **Kumori node-disjoint paths**: Those paths are node-disjoint paths between PoPs that cross a Kumori node located at a given IXP.

The number of Kumori disjoint paths found between two PoPs is compared with the number of direct disjoint paths to evaluate Kumori's benefits in terms of resiliency.

We look after the number of disjoint paths among the PoP pairs of two cloud services providers, Amazon and Atos. We evaluate the benefits of Kumori by observing the increase in the number of disjoint paths available for the PoP pairs of those two CSPs. Regarding Kumori, we look at the impact of some factors in the design and deployment of the architecture such as the number of relay nodes at IXPs or the choice of the IXPs where the relay nodes are located.

6.2 Description of the disjoint path discovery algorithm

First of all, we need to determine how many disjoint paths can be taken between every CSP PoPs on the Internet directly or through an IXP. We use the PoP level directed graph representation of the Internet we built in chapter 5 for this path evaluation.

6.2.1 A first valley-free path traversal algorithm

As mentioned in Section 2.3.3, the problem of finding *valley free* paths in our PoP-level directed graph representation of the Internet is a complex issue. At the AS level, this problem is NP-hard. In previous works presented in [EHM+06] or in [KKAD15], graph transformation methods were suggested to transform an AS-level directed graph representation of the Internet and thus ease the discovery of policy compliant paths in the Internet. Those transformations use the fact that, in such a graph, most relationships are asymmetric and that routable Internet paths can only cross a single $p2p$ AS relationship. Yet, in our PoP-level internet graph, we consider every intra-AS relationships as symmetric and we do not limit the number of intra-AS edges a routable path can cross. Thus, we cannot apply the suggested transformation methods on our graph.

The impossibility to apply previous results in our PoP-level directed graph representation of the Internet led us to adopt another method to discover routable paths. In a first approach, we tried to search for *valley free* paths in our architecture without applying any transformation nor simplification to our graph. We applied a depth-first search algorithm because of its smaller memory footprint in comparison with a breadth-first search algorithm. In this path search, we translated the *valley free* constraint by keeping in mind the type of the last inter-AS edge that has been traversed. If the last inter-AS edge was tagged with *customer to provider*, then every inter-AS edge can be traversed in our path search. On the contrary, if the last inter-AS edge traversed was tagged with *peer to peer* or *provider to customer*, then only *provider to customer* inter-AS edges can be traversed next. In any case, the *internal* edges can always be traversed.

We have chosen to ignore the *unknown* edges in our path search as we could not apply them *valley free* routing policy rules. The details of this initial algorithm is presented in Algorithm 1.

We implemented this first naive path search algorithm in Python. It allowed us to lay down the foundations of our *valley free* path traversal algorithm, yet, we also realized that given the complexity of our PoP level graph, this naive method would not provide results in reasonable time: running this algorithm for a single PoP pair chosen in our PoP-level graph took several weeks on a 8 core machine.

6.2.2 Algorithm Optimization

After our first implementation, we looked after possible methods to speed up path search. As the complexity of our path search algorithm grows with the numbers of edges and vertices in the graph, we first looked after vertices to remove in order to

Algorithm 1 Initial Depth-first valley-free path search

Require: $start \leftarrow$ starting node,
$destination \leftarrow$ destination to reach,
$graph \leftarrow$ Internet representation graph,
$node \leftarrow$ current node,
$in \leftarrow$ state describing the last inter-AS edge crossed,
$explored \leftarrow$ list of previously crossed nodes,
$queue \leftarrow$ queue containing $(node, in, explored)$ tuples.
Ensure: $node = destination$
$incoming \leftarrow c2p$
$node \leftarrow start$
$queue \leftarrow INSERT((start, c2p, explored), queue)$
while $queue$ is not $empty$ **do**
$(node, incoming, explored) \leftarrow POP(queue)$
if $node = destination$ **then**
return $explored$
end if
$children \leftarrow CHILDREN(node, graph)$
for $child$ in $children$ **do**
$edge \leftarrow EDGE(node, child, graph)$
$next_{in} \leftarrow TYPE(edge)$
if $child$ in $explored$ **then**
$continue$
end if
if $next_{in}$ is $internal$ **then**
$explored \leftarrow explored + node$
$queue \leftarrow (child, in, explored)$
end if
if in is $c2p$ or $next_{in}$ is $p2c$ **then**
$explored \leftarrow explored + node$
$queue \leftarrow (child, next_{in}, explored)$
end if
end for
end while

reduce the PoP level graph's size. We considered the directed AS level internet graph and we looked in this graph after *valley-free* AS paths between the CSP PoPs and from the CSP PoPs to the logical nodes representing the IXPs.

Besides, we looked at possible methods to further enhance the algorithm. Considering that *valley-free* disjoint paths is a subset of all disjoint paths in a directed graph, we looked at possible solutions to this problem in the literature. Itai *et al.* [IPS82] prove that bounding the length of the path drastically reduces the complexity of the disjoint path discovery algorithm. Making this assumption in our case makes sense because, as explained by Kühne *et al.* [KA12], the average AS path length in the Internet is relatively stable at 4.3 AS hops between 2010 and 2012 despite the increase

in the number of AS. Besides, the Internet is flattening according to Dhamdhere *et al.* [DD10]. We introduced such constraint in our depth-first search algorithm. First, we limited the AS path length to 4 in our AS level path search. Then we ran our algorithm for increasing path length limits starting from 3 at the PoP level. Before running each step, we prune previously found paths from the PoP level graph to reduce the number of paths to explore. We present the impact of these path length limits on the discovered path diversity in Sections 6.4.2 and **??**.

In a last step, we further improved our algorithm by computing the shortest path length between every vertices in our PoP level graph. To simplify this computation, we removed the constraints associated to the *valley-free* routing policy. We obtain the shortest path length for every PoP pair. We then use this shortest path length information for two purposes. First, knowing the length of the shortest possible path to a destination PoP can help us abandon the exploration of some paths. Indeed, at each vertex, we subtract the shortest path length to the path length limit. If the result is negative, meaning that the shortest path is longer than the path length limit currently explored, we abandon the exploration of this path. Besides, we used the shortest path metrics at each step to rank the possible successors of a given nodes before putting them in the queue of nodes to explore. This modification has been inspired by the A* algorithm [HNR68]. We use the computed shortest path distance as a metric indicating the most appropriate candidate successor to visit to reach the destination quicker. The final version of our algorithm is presented in Algorithm 2.

6.2.3 Algorithm execution

We implemented our optimized algorithm in C++ using Cython [BBC+11], a Python to C/C++ compiler. We have chosen to implement the algorithm in C++ for performance and memory footprint reasons. We then ran our disjoint path search algorithm on a cluster composed of 9 double compute blades each equipped with 2 Intel Xeon processors running at 2.6 GHz. The disjoint path search jobs execution has been managed and split among the computing processes using SLURM [YJG03], a job scheduler used in many Linux or Unix-based HPC (High Performance Computing) clusters.

6.3 Evaluation Methodology

In this section, we present how we use the paths we found with the algorithm described in the previous section to evaluate the diversity of those paths and compare several methods to access disjoint paths. In our diversity evaluation, we use two metrics

Algorithm 2 Depth-first valley-free path search with path length limit

Require: $start \leftarrow$ starting node,
$destination \leftarrow$ destination to reach,
$graph \leftarrow$ Internet representation graph,
$node \leftarrow$ current node,
$in \leftarrow$ state describing the last inter-AS edge crossed,
$explored \leftarrow$ list of previously crossed nodes,
$metrics \leftarrow$ dictionary associating each node the length of the shortest path to destination,
$queue \leftarrow$ queue containing $(node, in, explored)$ tuples.
Ensure: $node = destination$
$incoming \leftarrow c2p$
$node \leftarrow start$
$queue \leftarrow INSERT((start, c2p, explored), queue)$
$list_{temp} \leftarrow$ possible candidates list
while $queue$ is not $empty$ **do**
  $(node, incoming, explored) \leftarrow POP(queue)$
  if $node = destination$ **then**
    return $explored$
  end if
  if $length(explored) > limit - metrics[node]$ **then**
    return $failure$
  end if
  $children \leftarrow CHILDREN(node, graph)$
  for $child$ in $children$ **do**
    $edge \leftarrow EDGE(node, child, graph)$
    $next_{in} \leftarrow TYPE(edge)$
    if $child$ in $explored$ **then**
      $continue$
    end if
    if $next_{in}$ is $internal$ **then**
      $explored \leftarrow explored + node$
      $list_{temp} \leftarrow (child, in, explored)$
    end if
    if in is $c2p$ or $next_{in}$ is $p2c$ **then**
      $explored \leftarrow explored + node$
      $list_{temp} \leftarrow (child, next_{in}, explored)$
    end if
  end for
  $list_{temp} \leftarrow SORT(list_{temp}, metrics)$
  $queue \leftarrow list_{temp}$
end while

elaborated in previous work. First, we use the cumulative distribution of the number of diverse path found between several node pairs, as presented by Texeira *et al.* [TMSV03b]. In our work, we intend to apply this method to the PoP-level, directed Internet graph we presented in Chapter 5 and thus go beyond the limits of single ASes.

Besides, we use the diversity score introduced by Rohrer *et al.* [RS11]. This diversity score is comprised between 0 and 1, and can be used to characterize the diversity of two paths, of a set of paths between two nodes or of the paths within a whole graph. We will use this diversity score in our methodology to provide a synthetic score assessing the path diversity among PoPs.

6.3.1 Path Diversity Cumulative Distribution

The first metric we use is the path diversity cumulative distribution introduced by Texeira *et al.* in [TMSV03a] and in [TMSV03b]. In this evaluation, we adapt it and compare the edge-diverse paths cumulative distribution function and the node-disjoint paths cumulative distribution function to the maximum diversity cumulative distribution function. The maximum diversity cumulative distribution function is computed by determining, for each pair of nodes we consider, the minimum between the source's outbound degree and the destination's inbound degree. The resulting number is the maximum number of paths that can be found for the pair. This maximum diversity is an upper bound for the edge-disjoint path diversity and the node-disjoint path diversity. The closer we approach the maximum diversity, the better the capacity of a given system to access the existing diverse paths between two given nodes.

6.3.2 Diversity scores

The second metric that we use to characterize path diversity is the diversity scores presented by Rohrer *et al.* [RJS14]. The authors introduce a set of metrics to characterize diversity. A path P between a source s and a destination d is defined as a vector containing all links L and all intermediate nodes N traversed by that path. This is expressed by:

$$P = L \cup N \tag{6.1}$$

Using this definition, Rohrer *et al.* define the diversity score of two arbitrary paths P_a and P_b. This diversity score is given by the formula:

$$D(P_a, P_b) = 1 - \frac{|P_a \cap P_b|}{|P_a|} \tag{6.2}$$

Where $|P_a| \leq |P_b|$. This diversity score has a value comprised between 0, meaning identical paths, and 1, meaning completely disjoint paths.

From this diversity score between two arbitrary paths, Rohrer *et al.* compute a global diversity score for a set of paths $\{P_0...P_k\}$, the Effective Path Diversity (EPD). This

EPD is given by the formula:

$$EPD = 1 - e^{-\lambda k_{sd}} \tag{6.3}$$

Where k_{sd} is a measure of the added diversity for all the considered pairs. This measure is given by the sum:

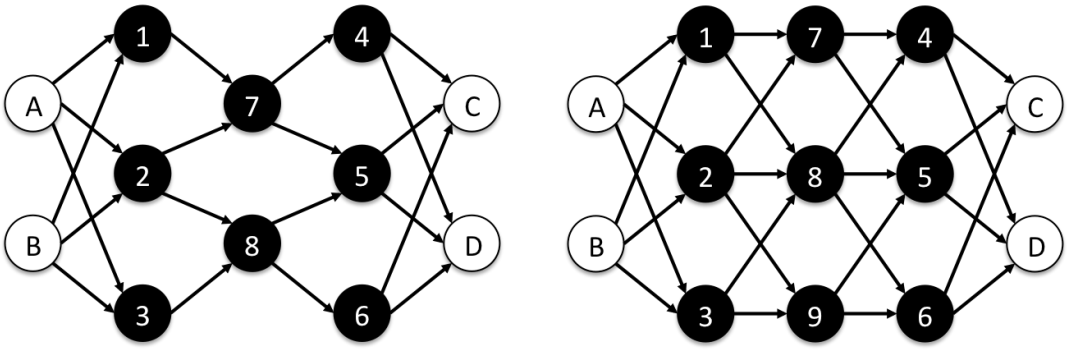
$$k_{sd} = \sum_{i=1}^{k} D_{min}(P_i) \tag{6.4}$$

Where $D_{min}(P_i)$ is, for a path P_i, the minimum diversity score obtained by computing the diversity score to every other paths in the set $\{P_0...P_k\}$ and λ is a constant factor scaling the importance of providing diversity among the path set. In our evaluation, we choose to set λ to 1. The EPD of a set of paths is also comprised between 0 and 1. The EPD of a set containing less than 2 paths is equal to 0.

The Effective Path Diversity score of a graph is given by the average of the EPD scores of the sets of disjoint paths between every $source$ and $destination$. In our diversity evaluation, the Edge-disjoint EPD is the mean of the EPD scores of the sets of edge-disjoint paths between every node pairs we consider, while the Node-disjoint EPD is the mean of the EPD scores of the sets of node-disjoint paths.

6.3.3 Metrics evaluation on two example graphs

In order to provide an intuitive understanding of the ability of the metrics presented in this section to capture path diversity in a graph, we compare two mock-up graphs. Those two graphs are presented in figure 6.1.



(a) Graph 1 (b) Graph 2

Figure 6.1 – Example evaluation graphs

We observe that the first graph is presenting a smaller node path diversity than the second graph where there is no bottleneck in the middle of the graph.

For the first graph, the cumulative distribution metric plot is represented on

figure 6.2. On the figure, we see that the node-disjoint cumulative distribution function is below the max diversity and the edge-disjoint cumulative distribution functions. This represents the impossibility to find three node disjoint paths between A or B and C or D given that nodes 7 and 8 are a cut. For the second graph, the three cumulative distribution functions are instead superposed.

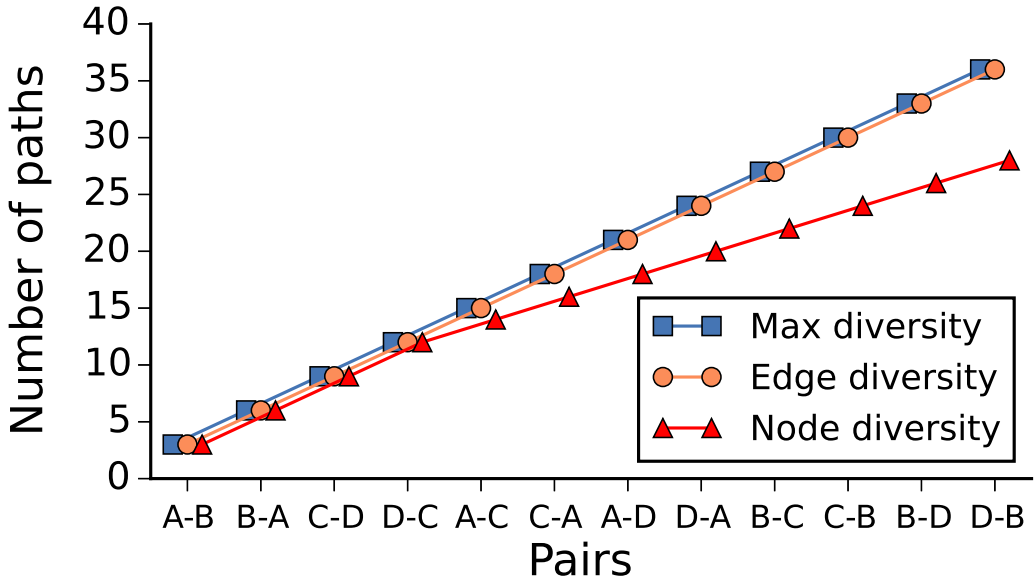


Figure 6.2 – Cumulative distribution of the number of paths between A, B, C and D for graph 1

Let's now look at the diversity scores. We consider the first graph to show how the diversity score is computed for an example pair, $A \to C$. For this pair, three edge-disjoint paths can be found between A and C: $P_1 =< A \to 1 \to 7 \to 4 \to C >$, $P_2 =< A \to 2 \to 7 \to 5 \to C >$, and $P_3 =< A \to 3 \to 8 \to 6 \to C >$. By calculating the diversity of the first two paths, we find:

$$D(P_1, P_2) = 1 - \frac{|P_1 \cap P_2|}{|P_1|} = 1 - \frac{1}{7} = \frac{6}{7} \tag{6.5}$$

The Effective Path Diversity $EPD_{A,C}$ of the pair $(A; C)$ can be calculated as:

$$\lambda k_{sd} = \sum_{i=1}^{k} D_{min}(P_i) = 2 * \frac{6}{7} + 1 = \frac{19}{7} \tag{6.6}$$

$$EPD_{A,C} = 1 - e^{-\lambda k_{sd}} = 1 - e^{-\frac{19}{7}} = 0.93338 \tag{6.7}$$

Table 6.1 shows the results obtained by completing the calculation for both graphs and both Node- and Edge- disjoint paths. Those results can help us interpret results on a unknown graph. First of all, the diversity scores for graph 2 are higher than graph 1,

which represent the gain in diversity provided by the addition of node 9. Besides, if we observe the node-disjoint $EPD_{A,C}$, we can see that the score has increased in graph 2, while the diversity score of two node-disjoint paths is equal to 1. This increase is due to the fact that in graph 2, 3 node-disjoint paths can be found between A and C while only 2 can be found in graph 1.

Table 6.1 – Diversity scores for graphs 1 and 2

Metric	Graph 1	Graph 2
Edge-disjoint $EPD_{A,C}$	0.93338	0.95021
Node-disjoint $EPD_{A,C}$	0.86466	0.95021
Edge-disjoint EPD	0.93899	0.95021
Node-disjoint EPD	0.89318	0.95021

6.4 Path Diversity improvements with Kumori

6.4.1 Internet Path Diversity for Two Real CSPs

We use the methodology we presented in the previous sections to measure the Internet path diversity among PoPs belonging to two real world CSPs, namely Amazon and Atos. Amazon is the largest Content Service Provider in terms of market share [**?**], while Atos is one of the largest European CSPs. Those two CSPs are connected in a different way to the Internet. Amazon is present through the ASes it manages at 52 different IXPs, while Atos is only present at a single IXP. Indeed, through its AWS Direct Connect offering, Amazon makes a lot of efforts to be tightly connected to popular ISPs and IXPs. Such a solution allows customers to connect with a direct route to Amazon from a set of IXPs [AWS].

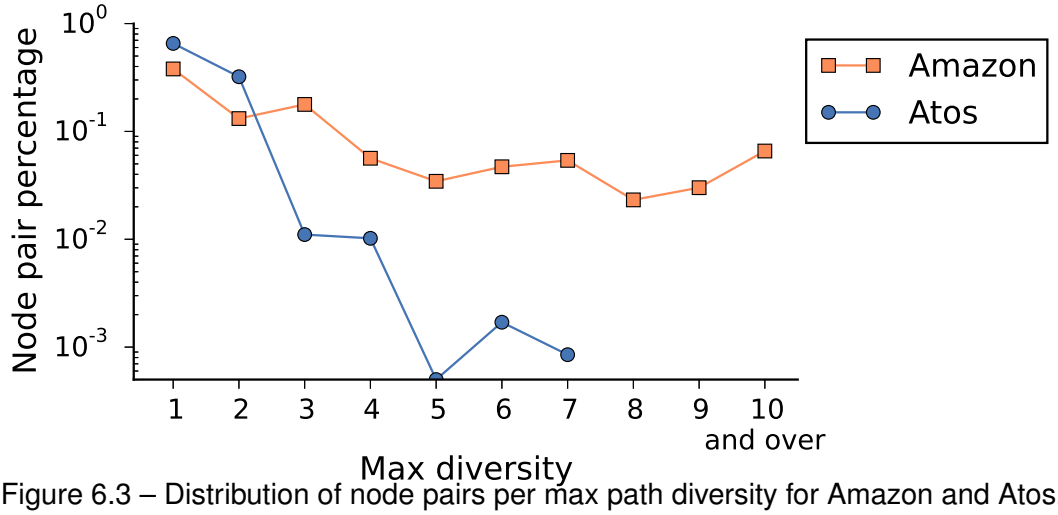


Figure 6.3 – Distribution of node pairs per max path diversity for Amazon and Atos.

On figure 6.3, we observe the distribution of PoP pairs as a function of the maximum path diversity (as defined in Section 6.3), for both Atos and Amazon. We can see from this figure that the maximum diversity of the Atos PoP pairs is less important than for Amazon. In particular, for nearly 70% of the Atos PoP pairs, only one path can be taken. We explain this difference as a direct effect of the connectivity strategies adopted by both CSPs. The efforts made by Amazon result in a lower number of PoP pairs with a limited path diversity. The efforts made by Amazon result in a higher number of PoP pairs with a large path diversity.

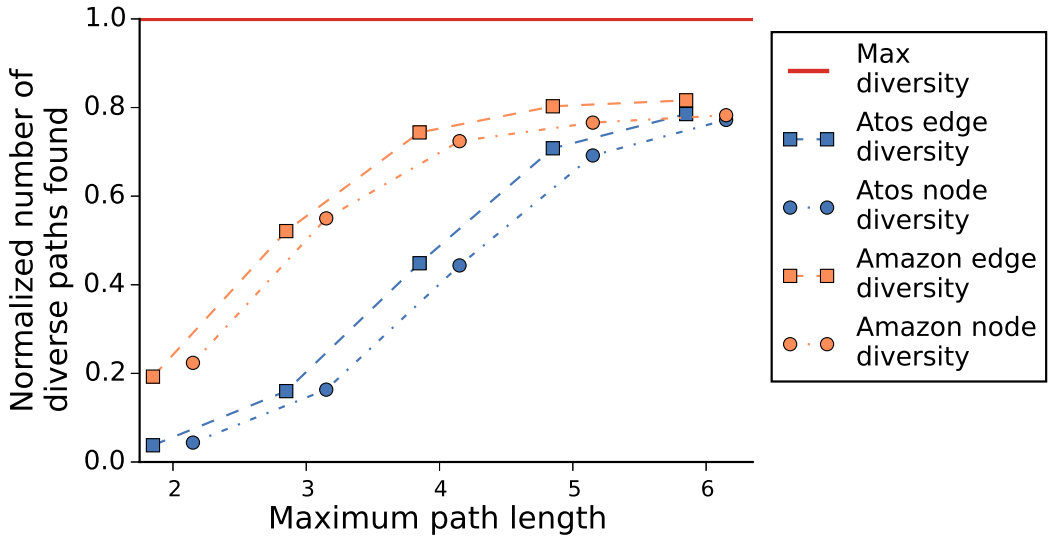


Figure 6.4 – Normalized number of edge-disjoint and node-disjoint paths found in the Internet according to the maximum path length for Amazon and Atos.

We look at the number of routable paths on the Internet that can be found using our algorithm among Atos's and Amazon's PoPs. Figure 6.4 is derived from the path diversity cumulative distributions computed for both CSPs. It shows the normalized number of edge-disjoint and node-disjoint paths found between each PoP pair for various maximum path length in both CSP's cases. The figure shows that the number of edge-disjoint or node-disjoint routable Internet paths that can be used among both CSPs PoPs is limited compared to their maximum diversity. At most, only 80% of the maximum diversity can be used. This limitation is related to the fact that not every existing topological path between PoPs can be taken in the Internet because of routing policies limitations.

Table 6.2 – Internet diversity scores for Amazon and Atos

Metric	Amazon	Atos
Edge-disjoint EPD	0.50637	0.17020
Node-disjoint EPD	0.50261	0.16431

We now compute the EPD scores presented in section 6.3.2 for both edge- and node- disjoints paths we found between Atos's and Amazon's PoP pairs. The results are presented in Table 6.2 and are relatively low. This means that the paths found among the PoP pairs follow the same edges or cross the same nodes.

6.4.2 Improving Resiliency Using Kumori

Our characterization of the path diversity between the PoPs of Atos and Amazon shows that both CSPs do not benefit from their full diversity potential. The diversity scores computed from the diverse paths found show a rather limited diversity, especially for Atos. The difference between the topological diversity highlighted by the maximum path diversity and the number of available routable paths tends to show that the limitations we observed are associated with the routing policies applied between ASes in the Internet.

In the following, we try to characterize the potential benefits of the use of the Kumori overlay in terms of resiliency using our path diversity evaluation methodology. Kumori takes advantage of the routing inflection points present at IXPs and uses traffic encapsulation to the routing inflection points to relax the restrictions imposed by BGP policies to the path that can be taken between two PoPs belonging to a given cloud services provider. Our hypothesis is that relaxing those policies at IXP will allow to increase the use of the topological diversity among PoP pairs.

We are using the path diversity evaluation methodology presented earlier to determine what are the benefits associated with the use of Kumori for Atos and Amazon. In this evaluation, we compare the routable path diversity for every PoP pairs of each CSP on the classic Internet and using a Kumori overlay with 5 routing inflection points. Our path diversity search algorithm reveals paths of length 7 and below. The routing inflection points we use are located at IXPs in Frankfurt, Sao Paolo, Hong-Kong, New-York and Seattle. This set of routing inflection points allow to have a broad geographic coverage.

Path Diversity Cumulative Distributions

Benefits of using Kumori in terms of path diversity: Figure 6.5 presents the average number of edge-disjoint and node-disjoint paths found for each PoP pair by our path search algorithm either using the Kumori overlay or on the classic Internet. The aim of such figure is to determine whether using Kumori increases the number of diverse paths found or not for the two CSPs. The figure shows that for Atos, using the Kumori architecture increases the number of accessible disjoint paths. For Amazon, there is a

benefit for edge-disjoint paths while we observe a degradation of the node-disjoint path diversity. We explain this degradation by the fact that we only used 5 IXP nodes for Kumori in our evaluation.

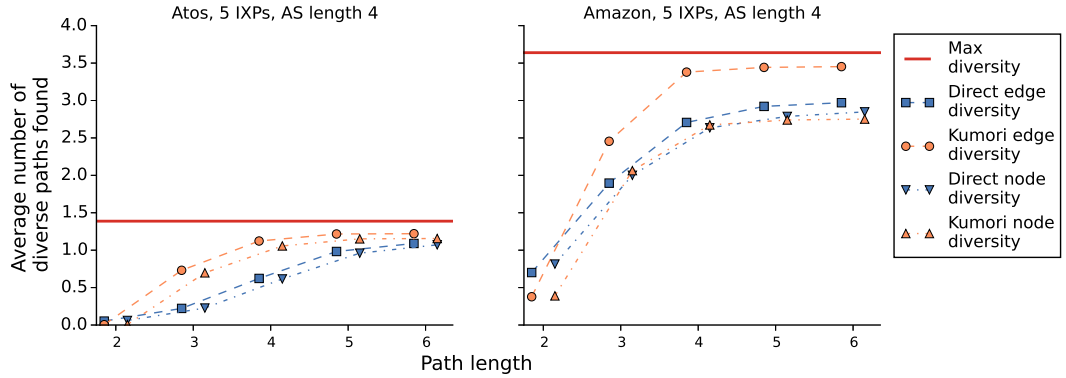


Figure 6.5 – Average number of diverse path found per maximum path length for Amazon and Atos

Impact of the maximum path length on the observed path diversity: We now compare the path diversity offered for different maximum path length on figure 6.5. This comparison is useful to determine whether there is a path length beyond which searching for path does not yield significant improvements. For both CSPs, we observe that with the Kumori architecture, searching for paths that are longer than 4 to 5 hops does not provide a significant improvement in terms of diverse paths found.

Observed Kumori benefits for different node categories: We consider now every PoP pairs and separate them into categories according to their maximum path diversity. Then, per each category, we compare the number of node-disjoint and path-disjoint paths found with Kumori and the classic Internet. Figure 6.6 shows the result, i.e., the normalized number of paths found per PoP pair category for both Amazon and Atos. In this figure, we observe that for both CSPs, the Kumori architecture yields a larger number of paths for node pairs with a low maximum diversity, while this benefit is less evident for node pairs with a higher maximum diversity. Note that, for Amazon, the edge-diversity efficiency of Kumori remains quite high, while the number of accessible node-disjoint paths tends to drop when the maximum diversity increases. We explain this effect by the limited number of IXP nodes used by Kumori in our evaluation. This small number of indirection nodes has a bigger impact on the diversity of PoP pairs that are already well connected.

Kumori Diversity Score

We evaluate the benefit, in term of path diversity, of the Kumori architecture using the diversity score presented in Section 6.3.2.

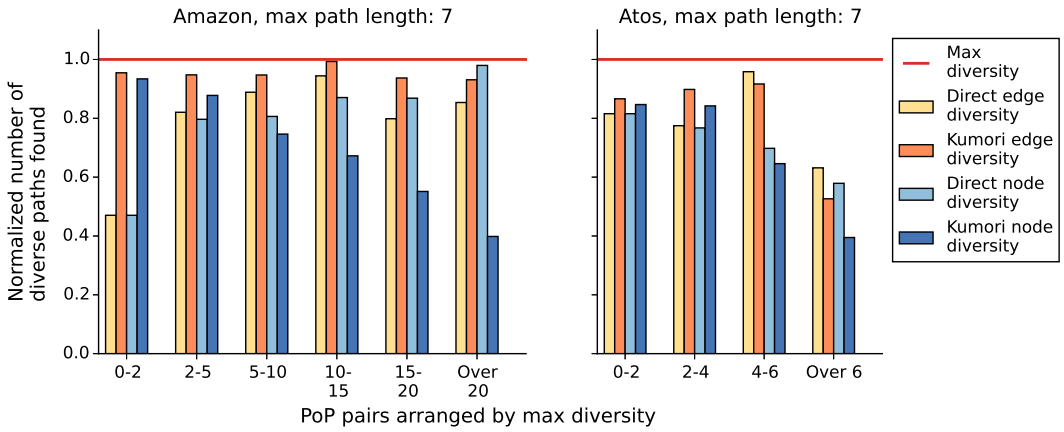


Figure 6.6 – Normalized number of paths found per node category for Amazon and Atos

Table 6.3 – EPD scores for the CSP PoP pairs

Type	Amazon			Atos		
	Direct	**Kumori**	**Kumori gain**	**Direct**	**Kumori**	**Kumori gain**
Edge-disjoint EPD	0.50637	0.55877	**10.35 %**	0.17020	0.23953	**40.73 %**
Node-disjoint EPD	0.50261	0.53413	**6.27 %**	0.16431	0.21383	**30.14 %**
Edge-disjoint EPD**, pairs with** $EPD_{Kumori} = 0$ **or** $EPD_{Direct} = 0$	0	0.87300	N.A.	0.02828	0.81320	N.A.
Edge-disjoint EPD**, no zero div.**	0.95303	0.95178	**-0.13 %**	0.85520	0.84459	**-1.24 %**
Node-disjoint EPD**, pairs with** $EPD_{Kumori} = 0$ **or** $EPD_{Direct} = 0$	0.13984	0.74271	N.A.	0.18118	0.66518	N.A.
Node-disjoint EPD**, no zero div.**	0.94669	0.93433	**-1.31 %**	0.86844	0.85199	**-1.89 %**

Table 6.3 shows the edge-disjoint and node-disjoint EPD scores obtained for all the CSP PoP pairs. We observe that using the Kumori architecture improves the overall diversity of the paths among both CSPs' PoP pairs. In both cases, the gain is higher for edge-disjoint path diversity than for node disjoint path diversity. Yet, we can observe that the gain in terms of diversity is far more important for Atos than with Amazon: it is roughly multiplied by a factor 4.

If we look at the last two lines in table 6.3, we observe that the Kumori architecture does not improve the edge-disjoint or node-disjoint EPD scores for node pairs that are already connected through 2 or more disjoint paths. Yet, some node pairs strongly benefit from using the Kumori architecture to increase their accessible path diversity.

This is the gain shown by the edge-disjoint and the node-disjoint EPD score obtained for the PoP pairs for which either EPD_{Kumori} or EPD_{Direct} is equal to 0 in table 6.3. Having a closer look at our data for the Amazon PoP pairs, using the Kumori architecture gives access to at least a second edge-disjoint path for 194 pairs that are connected through at most 1 direct edge-disjoint path. It gives access to at least a second node-disjoint path for 170 pairs that are connected by at most 1 direct node-disjoint path. We can conclude that Kumori provides diverse paths to a significant number of nodes that would otherwise only have one path to their destination. The gain is less important for nodes that already benefit from sufficiently diverse paths on the plain Internet.

6.4.3 Influence of Kumori architecture parameters on resiliency benefits

In the evaluation performed at this stage, we assumed the use of a Kumori overlay consisting in 5 routing inflection points picked to optimize their geographical diversity. Besides, in the execution of our path discovery algorithm, we have limited the AS path length to 4. Those choices can have an impact on the resiliency benefits of the Kumori architecture. In this section, we examine the influence of two parameters: the Kumori routing inflection points choice policy and the number of routing inflection points used in the overlay.

Influence of the routing inflection point choice policy:

First, we compare the average number of path found among the Amazon PoP pairs for two Kumori setups where 5 routing inflection points located at different IXPs are used. The two setups vary in the policy adopted to choose the location of the routing inflection points. For the first setup, we chose to place our routing inflection points in geographically-diverse IXPs. Our goal was to cover the various continents where Amazon is present. This is the policy we initially adopted in the main evaluation, with routing inflection points located at IXPs in Frankfurt (DE-CIX), Sao Paolo (PTT), Hong-Kong (HKIX), New-York (NYIIX) and Seattle (SIX). In the second setup, we used another policy to place our routing inflection points. We looked at the five largest IXPs in the PeeringDB database in terms of number of member AS, and chose to place our routing inflection there. As a result, the routing inflection points were placed in Amsterdam (AMS-IX), London (LINX), Frankfurt (DE-CIX), Sao Paulo (PTT) and Hong-Kong (HKIX).

The results of our path diversity evaluation is presented on figure 7.1. The figure shows a minor difference between the path diversity achieved with the two policies. Yet, it shows that in the setup of the Kumori overlay, it is better to favor geographical diversity over the IXP size in the placement of routing inflection points. Indeed, the IXP size in

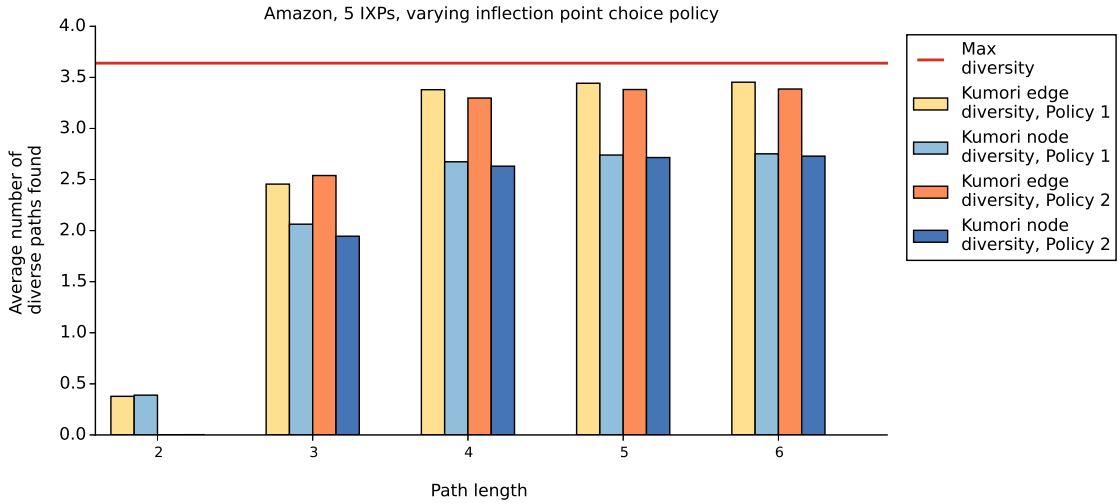


Figure 6.7 – Average number of path found among Amazon's PoP pairs depending on the routing inflection point choice policy

terms of number of member AS seems to be more a sign of the market diversity in different regions, and does not seem to ensure a higher path diversity.

Influence of the number of routing inflection points in the overlay:

We now compare different Kumori setups consisting in variable numbers of routing inflection points. In this experiment, we look at Kumori overlays with 3, 5, 7 and 10 routing inflection points. The IXPs where the routing inflection points are placed are chosen following a consistent policy where the IXPs with the largest number of participating ASes are favoured. We perform an edge-disjoint and node-disjoint path evaluation for both Amazon and Atos.

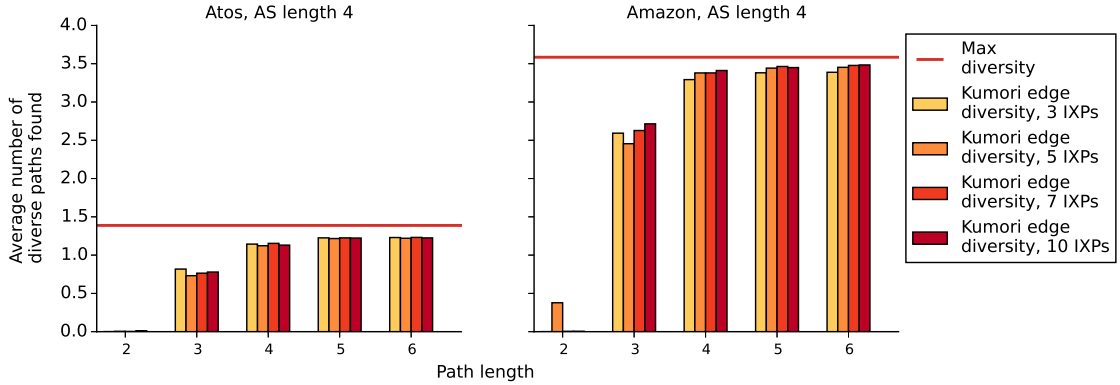


Figure 6.8 – Average number of edge-disjoint paths found among Amazon's and Atos's PoP pairs depending on the number of routing inflection points in the Kumori overlay

Figure 6.8 represents the evolution of the number of edge-disjoint paths according to the maximum path length for the various Kumori setups we consider. These

experimental results show that the number of routing inflection points in the Kumori overlay has a marginal impact on the path diversity obtained among both Amazon's and Atos's PoP pairs. Indeed, the performances of the various setups are indistinguishable for a path length of 5 hops or more.

The results that we obtained regarding the influence of the routing inflection point placement policy and of the number of routing inflection points in the overlay are very interesting for potential managers or users of Kumori systems. They show that it is more beneficial to setup an overlay with a limited number of geographically-diverse routing inflection points rather than setting up a larger number of nodes at larger IXPs. This property we have highlighted for Kumori can also be considered as a technical acknowledgement of the statement made by several local, smaller IXP managers that the IXPs they put in place help foster a more resilient Internet.

6.4.4 Kumori vs. Kotronis

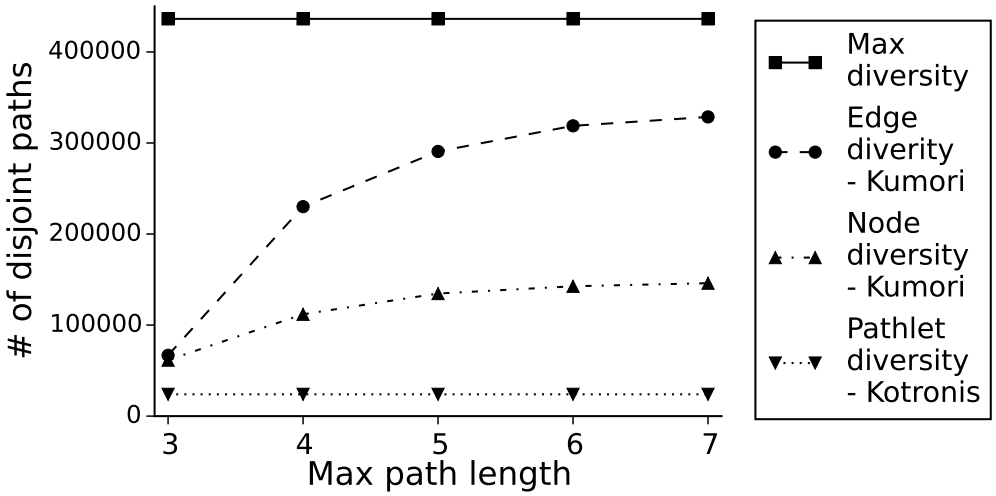


Figure 6.9 – Cumulative distribution of the number of edge-disjoint, node-disjoint and AS-disjoint paths between 54 largest IXPs

In this section, we compare the Kumori architecture with another overlay architecture using nodes at IXPs that has been presented recently. Indeed, the Kotronis overlay has been introduced in [KKR+16]. In the Kotronis overlay, some IXPs are instrumented in order to offer nodes alternative paths compared to the classic Internet. While in Kumori the motivation to access alternative paths is guided by a resiliency objective, Kotronis aims at offering better performance in terms of latency or jitter. Another difference between both architectures is that in their work, Kotronis *et al.* assume that between IXPs the traffic will follow a pathlet, *i.e.* a path composed of nodes within a single AS. Kumori proposes a more general approach, where paths just have to comply with the

Gao-Rexford Internet routing policy, potentially passing through several ASes. In the previous section, the resiliency results observed for Kumori still hold for the Kotronis overlay. Indeed, in our evaluation, Kumori paths only crossed a single IXP node, while the difference between the two overlays resides in the way inter-IXP paths are managed.

We compare these two architectures by looking after disjoint paths offered by both architectures between the 54 largest IXPs in term of confirmed AS members in our topology. We evaluate the diversity accessible using Kotronis by determining the number of AS-disjoint paths between the IXPs. We evaluate edge-disjoint and node-disjoint paths accessible using Kumori as in the previous sections. The results of this evaluation are presented in figure 6.9.

We can observe that the diversity is significantly higher using Kumori than using Kotronis. This is related to the choice made in the latter to limit possible paths between IXPs to pathlets located within a single AS. This choice comes from the aim of Kotronis to associate pathlets with QoS metrics controlled by a single Internet provider. For Kumori, edge diversity is significantly higher than node diversity, and it increases with the maximums path length. It shows that one of the major benefits associated with Kumori is its capacity to use Internet path between IXPs while relaxing Gao-Rexford routing policy at those IXPs. The results obtained by Kotronis show that this architecture is less suitable than Kumori to improve inter-PoPs path diversity.

6.5 Conclusion

The evaluation we have performed of the path diversity that Kumori allows cloud services providers to access and the comparison with path diversity between CSP PoP pairs over the Internet underlines the benefits provided by this architecture. On average, using Kumori enhances edge-disjoint path diversity by 10.35% for Amazon and by 40.73% for Atos, while it increases node-disjoint path diversity by 6.27% for Amazon and by 30.14% for Atos. A detailed study of our evaluation results shows that the benefits provided by Kumori are particularly interesting for CSP PoP pairs that suffer from a limited topological path diversity. We can also underline that even when the use of Kumori does not improve the number of accessible disjoint path significantly, Those disjoint path are generally shorter in terms of number of hops, which has an impact on their performance and on our ability to monitor them.

A closer look at the influence of several parameters on Kumori's performance has highlighted that Kumori's benefits can be achieved without having to deploy a large number of routing inflection points at the largest IXPs. Instead, our evaluation stresses that Kumori requires a set of carefully selected, geographically-diverse routing inflection

points to provide a better disjoint path diversity to its users. This finding is particularly interesting from an economical standpoint because the cost for setting up, operating and maintaining an overlay architecture such as Kumori depends on the number of nodes that constitute it.

Chapter 7

Kumori economics

Contents

In this chapter, we aim to evaluate the economical aspects inherent to the setting and to the management of the Kumori architecture introduced in the previous chapters. Beyond the benefits of the Kumori architecture in terms of inter-datacenters resiliency, we want to analyze the economical viability of this approach for Cloud Services Providers (CSP). In order to present the economics of the Kumori architecture, we first provide background information on the business relationships between autonomous systems (AS) in the Internet. We then detail the cost structure of several Internet interconnection methods. We evaluate the economical efficiency of the Kumori architecture compared to a mesh of long-haul private connections. Such connections are necessary to enable the setting and the operations of Cloud infrastructures.

7.1 Internet core economics background

The Internet consists in a set of heterogeneous networks made of layer-3 IP routers interconnected between each others by long distance layer-2 data links. These data links are themselves aggregated into layer-1 WDM (Wavelength Division Multiplexing) optical links. In this thesis, we only deal with layer-3 connections. The heterogeneous IP networks constituting the Internet are called Autonomous Systems (AS). They are managed by telecommunication operators (also designated by telcos), Internet services providers (ISP) and Cloud Services Providers (CSP). Service providers such as CSPs and data content providers or video providers provide services to the end-users (enterprises, public administrations or individuals). In this context, each AS serves a set of end-users. Its responsibility consists in connecting these end users to the rest of the world, *i.e.* to provide a method to reach any single other AS. From an administrative perspective, an AS is identified in the Internet using an AS number. This AS number is obtained from a regional Internet registry such as RIPE, ARIN, or APNIC. Besides, this AS also has a range of IP addresses of its own. An AS can get connectivity to the rest of the Internet using two kinds of mechanism: *transit* and *peering*.

Transit can be described as the service provided by an ISP to connect the end users of a smaller network to the rest of the Internet. In a transit connection, the smaller network pays to send and receive network traffic through the Transit provider. To do so, from a routing perspective, the transit provider advertises the IP prefixes belonging to its customer to the rest of the world, and advertises all the destinations it can reach to this one. The customer pays a fee to send and receive traffic from its transit provider. This fee is typically fixed depending on an agreed upper bandwidth consumption. An AS may have agreements with several transit providers for the sake of cost arbitrage or resiliency. In such a case, we qualify this autonomous system of multihomed AS.

On the other hand, *peering* is an interconnection method in which two ASes agree to exchange traffic from and to their respective end users, often free of charge. Indeed, contrarily to transit, peering is considered as settlement-free. From a routing perspective, both autonomous systems advertise to their peer the IP prefixes of their end-users or customers. A peering relationship is not transitive: a network cannot advertise its peers' end-user prefixes to its other peers. Most of the time, peering agreements are taken between networks that share a rather balanced traffic: their end users and customers send and receive a similar amount of traffic from one AS to one another.

In order to exchange traffic, Autonomous Systems need to connect to both their transit providers or their peers. This may take the form of private bilateral interconnection links connecting network infrastructures. Autonomous systems willing to connect to a rather large number of networks can settle at specific private

interconnection facilities or at Internet Exchange Points (IXP). Internet Exchange Points are progressively deployed in the Internet with Internet traffic increase. IXPs play the role of interconnection facilitators. In this context, autonomous systems can peer with other ASes or build transit link relationships with Internet Services Providers. In that extend, IXPs are kind of demilitarized zones where relationships and interconnections are facilitated. Autonomous systems that want to participate in an IXP need to pay to become a member. They usually pay a fee to the organization managing the IXP depending on the requested interconnection capacity at this IXP. ASes also pay for the colocation costs to host and operate their own infrastructure in this external environment.

Autonomous systems do not rely exclusively on peering or on transit to connect to the external world. Several autonomous systems adopt both strategies to optimize their connectivity costs while maintaining a sufficient level of resiliency or quality of service. Yet, transit and peering links cannot be traversed in the same way. As it has been explained in Section 2.3.1, the business relationships induced by transit or peering agreements result in an asymmetry of the relationships between autonomous systems.

Transit and peering are used to connect autonomous systems to the Internet. Internet connectivity is often provided with best effort quality of service. Some autonomous systems may need to connect remote datacenters or locations while maintaining a given quality of service in terms of jitter, (*i.e.* the variance of the time space between successive IP packets of a same data flow), latency (*i.e.* the average IP packet transit delay), guaranteed bandwidth or reliability. Reliability can be evaluated as the probability that a packet is lost due to the expiration of its TTL (Time-to-Live). In such cases, ASes can use long-haul private links operated by large network providers in the form of MPLS circuits or carrier-grade Ethernet interconnects. The price of those long-haul links depends on the link's capacity, its level of reliability and on the locations it interconnects.

The goal of the Kumori architecture is to provide an intermediate alternative between, on one side, the best effort connectivity provided by the Internet, and, on the other side, a private connectivity mesh consisting in long-haul dual private links provided by separate providers. In the next section, we detail the cost structure for the various connectivity methods presented previously in this chapter.

7.2 Network connectivity: analysis of the associated costs

The various connectivity methods presented in the previous section do not only differ according to their technical characteristics or their relative benefits in terms of connectivity. They also tend to differ in their cost structure for the customer autonomous

systems. In this section, we present those various cost structures. We then provide some estimates of their market value as of May 2016. Those prices are often covered by non-disclosure agreements between the providers and their customers. In this thesis, we use data gathered from communications made at RIPE meetings or at Internet operators conferences ([SHK15], [Bry16], [BBM16]). We supplemented these data with elements retrieved either from technical articles written by large CSPs ([Pri14]) or from information exchanged on the NANOG mailing list, a community gathering network administrators from all over the world ([SAH]). These data give us rough estimates of the prices on the global connectivity market. We shall use these data in the remaining of this chapter to calculate the cost and relative economical benefits of the Kumori architecture.

7.2.1 Transit

As explained previously, autonomous systems can use the transit services of some Internet services providers to get access to the Internet beyond their close neighborhood. This transit is a commercial service for which the autonomous systems pay a fee to the Internet Service Provider. On the global Internet market, transit traffic is paid according to the consumed bandwidth on the link on the basis of the 95^{th} percentile method. In this method, the bandwidth used by a customer autonomous system on its transit links with the ISP is measured at fixed time intervals. The topmost 5% bandwidth consumption measurement are excluded from the traffic measurement itself. The highest measured bandwidth is then used to compute the price of transit for this customer. Besides, customer autonomous systems often commit to a minimum bandwidth usage that covers the costs of operation of the transit links. The unit price of a transit link is calculated for a Megabit per second per month [SHK15], [BBM16], [Bry16].

The cost of Transit is not uniform around the globe. In some very well connected regions, such a Western Europe or Norther America, transit costs roughly 1$ per Mbps per month, while in Australia, transit can cost up to 18 $ per Mbps per month. In our economical evaluation of the Kumori architecture, we consider an average price for transit in 5 large regions: Asia, Northern America, Europe, Australia and Southern America. Those prices are computed from the data presented in [SHK15], [BBM16] and [Bry16]. Table 7.1 summarizes the prices we shall consider in our economical evaluation.

Customer AS can buy transit from Internet Service Providers on the basis of the costs provided in Table 7.1 if they can carry their Internet traffic to a large collocation site where Internet services providers are present or to an Internet Exchange Point. To connect their datacenters to those interconnection points, the customer AS uses a short

Table 7.1 – Transit price in five different global regions of the world

Region	Transit price ($ per Mbps per month)
Europe	1
Asia	9
Northern America	1
Southern America	17
Australia	18

distance interconnect. Such an interconnect corresponds to a private dedicated line. The cost of such leased lines depends on various parameters:

- The maximum bandwidth that the customer autonomous system is willing to use,
- The distance between the datacenter and a location owned by the dedicated line's provider,
- The service level agreements on this dedicated line,
- Soft parameters such as the customer's ability to negotiate,
- The provider's market position.

For the sake of our evaluation, we give an estimation of the cost of IXP interconnections with a 10 Gbps capacity in Table 7.2 taking advantage of information we extracted from presentations made by Telegeography analysts ([SHK15], [BBM16], [Bry16]).

Table 7.2 – Price for 10 Gbps IXPs interconnections in five different global regions of the world

Region	Interconnection price ($ per month for 10 Gbps)
Europe	3000
Asia	9000
Northern America	3000
Southern America	9000
Australia	9000

7.2.2 Peering

When an autonomous system exchanges traffic in a nearly symmetrical way with another network, it may establish a direct peering relationship with this other autonomous system. As presented earlier, peering can be operated on dedicated private lines connecting two networks. Yet, they can be accommodated at private peering facilities or at Internet Exchange Points. Internet Exchange points are neutral hosting facilities in the Internet where several market players can meet and interconnect their infrastructure. The role of those Internet Exchange Points has been highlighted

in [ACF+12]. In this paper, the authors underline the growing importance of traffic exchanges at a large European IXP. Internet Exchange Points (IXP) have various governance models. Depending on the region they operate, IXPs are managed either by not-for-profit organizations or private companies that act as intermediate between autonomous systems.

To connect to an IXP and peer with other networks, autonomous systems need to have concluded an agreement with this IXP. In most cases, an AS becomes a member of an IXP by buying a port on one of the IXP's switch. The price of these ports depends on the company managing this IXP and on the bandwidth that the AS is willing to use to exchange its traffic through this IXP. If we study the price list gathered by the NANOG community in [SAH], the prices for the IXP ports in the various global regions we consider tends to be relatively similar, as the IXPs compete to attract networks on their marketplace. We summarize the average port prices we shall use in our economical study in Table 7.3.

Table 7.3 – Price for a 10 Gbps IXP port in five different global regions of the world

Region	Port price ($ per month for 10 Gbps)
Europe	1400
Asia	1000
Northern America	1200
Southern America	2000
Australia	2000

Beyond the intrinsic cost for using a port at an IXP, autonomous systems willing to exchange traffic at a given IXP must invest in their own routing equipment to get access to one or several ports of the IXP. Thus, an AS must pay for those equipment and for the collocation fees at the IXPs that includes facility's rental and power provisioning. Besides, ASes have to pay the IXP interconnection fees to carry their data traffic from their datacenters to the IXP's hosting facility. In our economical evaluation, we assume that, in average, each autonomous system places two routers at each IXP they want to participate. The estimated cost for such an operation is around 4000$ for each router. We have obtained the average colocation fees in the various global regions from the Telegeography presentations ([SHK15], [BBM16], [Bry16]) that are publicly accessible. Table 7.4 summarizes the colocation prices we consider in our economical modeling.

Table 7.4 – Colocation price for half a rack in five different global regions of the world

Region	Colocation price ($ per month for half a rack)
Europe	1400
Asia	1000
Northern America	1200
Southern America	2000
Australia	2000

7.2.3 Peering and Transit in the Internet today

Peering and transit have respective technical and economical benefits for autonomous systems willing to connect to the Internet. In our economical evaluation of the Kumori architecture, we are interested in determining how customer networks, especially Cloud Services Providers, use transit or peering to convey their traffic. In a recent technical blog post [Pri14], Cloudflare, a Cloud Services Provider with a global footprint, has published for the networking community some information on its use of either peering or transit in the various global regions in the world. In this article, Cloudflare underlines that globally the share of their peered traffic increases permanently. Yet, the ratio of the peering traffic volume to the transit traffic volume observed at their various IXPs differ depending on the global region. Thus, in Europe, Southern America, Asia or Australia, Cloudflare manages exchange from 50 to 60% of the data traffic generated by these various regions of the world, while in Northern America, only 20% of the traffic is peered.

In our economical evaluation of the Kumori architecture, we shall take this distribution between peering and transit into consideration for the evaluation of the cost of the network traffic flowing through routing inflection points. Table 7.5 presents the percentage of peered traffic observed by Cloudflare in the various global regions of the world.

Table 7.5 – Peered traffic percentage in five different global regions of the world

Region	Peered traffic percentage (% of the network traffic)
Europe	50%
Asia	55%
Northern America	20%
Southern America	60%
Australia	50%

7.2.4 Long-haul links

Autonomous systems may want to keep their WAN traffic isolated from their own Internet traffic for the sake of their operational safety. Indeed, several large Cloud Services Providers use today a mesh of private, long haul links to interconnect datacenters located in different regions of the world. These same CSPs also aim to ensure a sufficient level of resiliency or quality of service to their clients. Those private links are either built on purpose, which represents large investments, or rented from Internet Services Providers who build those links by taking advantage of their existing infrastructure.

According to the elements we could gather from Telegeography presentations

([SHK15], [BBM16], [Bry16]) and from discussions with network infrastructure buyers, the cost of long-haul links includes a long distance component and two access components. The long distance component is accounting for the price of the connection between two remote Points of Presence (PoPs) of the network connectivity supplier. The access components cover the cost of the connection from those PoPs to the customer's premises. The long distance component itself is composed of a submarine component (sometimes referred to as the wet capacity), and of a terrestrial component (sometimes referred to as the backhaul). Operating a terrestrial cable is more costly than operating a submarine one.

The Kumori architecture is intended to be a replacement for the typical dual private link connectivity mesh used by Cloud Services Providers to interconnect their datacenters. In this context, we need to determine an average price for long-haul private links in the various global regions and between these same regions. Table 7.6 presents the average long-haul link costs that we computed using the data retrieved from [SHK15], [BBM16] and [Bry16].

Table 7.6 – Average price for a 10 Gbps private connection within five different global regions of the world and between those global regions

Region	Link price ($ per month for 10 Gbps)
Europe	1600
Asia	42000
Northern America	4250
Southern America	30000
Australia	8500
Asia - Northern America	21050
Northern America - Europe	6700
Europe - Asia	30000
Northern America - Southern America	33900
Asia - Australia	30000
Australia - Northern America	60000

7.2.5 Cost trends

From [SHK15], [BBM16] and [Bry16], we can observe a regular price reduction for several cost elements that were presented in the previous sections. The price of long-haul private connections and equipment tend to drop by 20% every year. In parallel, transit price tends to drop by 30% yearly. As those cost evolutions are not identical, they may induce at the margin a bias in our economical evaluation of the Kumori architecture for the next years. This is why we intend study the impact of these cost reductions on Kumori's architecture economics.

7.3 Kumori cost

In the Kumori architecture, Cloud Services Providers use routing inflection points located at various IXPs in order to steer traffic in the Internet. For the CSPs, running the Kumori architecture involves placing those routing inflection points at IXPs around the globe. In this section, we detail the cost of setting up and operating a Kumori routing inflection point in various locations.

In order to place a Kumori element at an IXP, a CSP must become a member of this IXP. For that purpose, this CSP has to pay for the usage of an input/output port at the IXP's router. Then, this same CSP has to place two high-end SDN routers connected to the input/output ports of this IXP. The CSP has to pay for the colocation fees of these two high-end routers at the IXP premises. These fees correspond to the monthly cost for renting racks in the shelves of the IXP (including the powering and cooling of the two routers). In the IT industry, the cost of equipment such as routers or servers is spread over 3 years. Thus, every month, $1/36^{th}$ of the total equipment cost is paid on top of the colocation fees. At last, the relayed traffic for the Kumori routing inflection point is also charged by the IXP to the client CSP. If we want the infrastructure to sustain 10 Gigabits per second of traffic, then the total bandwidth that needs to be available at each routing inflection point is 20 Gigabits per second (input/output traffic).

The elements composing the global operating cost of a Kumori routing inflection point vary according to the region were this point is operated. Table 7.7 lists the costs we evaluated from the data presented in Section 7.2. We discriminate the various cost elements, and tackle two scenarios regarding the cost of the network traffic operated by the Kumori node. First, we consider that all the traffic is relayed using transit contracts. Then, we consider that a share of this traffic is relayed in using peering links.

If we compare the costs presented in Table 7.7, one notices that network connectivity accounts for the largest share in the global costs, while the equipment and IXP hosting fees are relatively similar across regions. Besides, there is a strong incentive to try to establish peering relationship at the same extend as CSPs such as Cloudflare to reduce the networking costs.

Table 7.7 – Cost structure for operating a single kumori node within five different global regions of the world

Region	Europe	Asia	Northern America	Southern America	Australia
IXP port fees ($/month)	1400	1000	1200	2000	2000
Colocation fees ($ / month)	1700	2000	1400	3000	3000
equipment costs ($ / month)	222.22	222.22	222.22	222.22	222.22
Networking costs for 10 Gbps transit ($ / month)	10240	92160	10240	174080	184320
Networking costs for 10 Gbps transit/peering mix ($ / month)	5120	41472	8192	69632	92160
Kumori node operational cost, Transit network traffic ($ / month)	23 802.22	187 542.22	233 02.22	353 382.22	373 862.22
Kumori node operational cost, Peering/Transit mix ($ / month)	13 562.22	86 166.22	19 206.22	144 486.22	189 542.22

7.4 Comparing the Kumori architecture and private WAN infrastructures

7.4.1 Evaluation topologies

In order to evaluate the economical relevance of the Kumori architecture, we compare its operational costs with the costs of an inter-datacenter mesh of long-haul dedicated links. In order to perform this comparison, we first looked after inter-datacenter networks used by cloud services providers in the literature. We identified three example topologies: the B4 wide area network used by Google, Facebook's inter-datacenter network and Amazon's wide area network infrastructure.

The B4 network topology has been presented by Jain *et al.* in [JKM+13], where the authors present the use of SDN by Google to optimize its use of the WAN connecting its datacenters. In the B4 network, 12 datacenters located in Asia, Europe and Northern America are connected using 18 long-haul connections spanning those regions. For the sake of evaluating the Kumori architecture, we make the hypothesis that Google is peering or buying Internet transit at 6 locations to serve its datacenters.

The Facebook inter-datacenter network topology is more difficult to determine. From Facebook's website, we know that the company uses 5 large datacenters in Northern America ([FBDf], [FBDb], [FBDc], [FBDd], [FBDa]) and 1 in Europe ([FBDe]). We assume that Facebook uses six long-haul links to interconnect its american datacenters together, and that two links are used to connect this mesh to the European datacenter.

Besides, we assume that Facebook is peering or buying transit for its datacenters at 3 locations, 2 in Northern America and 1 in Europe.

The Amazon inter-datacenter network has been presented to the public by James Hamilton, vice president and distinguished engineer for Amazon Web Services in 2014 during the AWS re:Invent conference [Van]. Amazon groups its datacenters in regions. A region can group from 2 to 5 datacenters together. Regions are connected together using long-haul links. They are each connected to 2 transit or peering facilities. The Amazon website [AMZ] presents a map of the various regions as well as the number of datacenters in each one of them. In our evaluation, we include both the existing and planned datacenters. This map is not revealing any information on the number and layout of the long-haul links connecting the regions together. We make reasonable assumptions on this existing infrastructure following the elements presented by James Hamilton.

Besides those three existing topologies, we have considered six other configurations to better study the economical relevance of the Kumori architecture. First, we grouped the three existing topologies in a large, global datacenter network. We summed up the datacenters and long-haul links used by Amazon, Facebook and Google, while keeping the number of peering sites equal to the number of peering sites used by Amazon. By evaluating the relevance of Kumori architecture for this topology, we want to evaluate the effect of mutualizing the architecture's elements for several CSPs on the architecture's operating costs. Then, we built five other network topologies by placing the whole B4 inter-datacenter topology in the five global regions we consider in our study, *i.e.* place the 12 datacenters and the 18 long-haul links constituting the B4 system in Europe, Asia, Northern America, Southern America and Australia. Given the difference between the connectivity markets in those regions, those five imaginary setups will help us understand how the long-haul and transit prices affect the economical benefits provided by the Kumori architecture.

7.4.2 Comparing Kumori and long-haul private link pairs connectivity

Using the cost elements presented in the previous sections, we compare the operational cost of Kumori's architecture with the cost for operating a private set of long-haul link pairs among the datacenters. In this evaluation, we have considered that every private long-haul link has a bandwidth capacity of 10 gigabits per second, and that Kumori's architecture uses the same amount of bandwidth on every logical link among the overlay. Every inter-datacenter private link consists in a pair of long-haul private links to emulate the classical resiliency strategy used by cloud services providers.

Regarding the Kumori architecture overlay setup, we detailed four cases. First, we considered that the CSP does not connect to peering sites to buy access to the

Internet, and we include the price for setting up and maintaining a presence at such interconnection facilities in the architecture's operational costs. Then, we make the assumption that this CSP already has an access to interconnection facilities. We consider the cost for increasing the link capacity from the datacenters to those facilities. For both situations, we have considered two connectivity provisioning strategies: In the first strategy, the CSP operating the Kumori overlay manages to establish peering relationships in the same extend as Cloudflare ([Pri14]). In the second strategy, the CSP has to pay for transit to send all its traffic.

Table 7.8 presents the operational costs for the different inter-datacenter connectivity solutions we have studied and for the various topologies we have considered. In this table, the price of the long-haul connectivity scheme is taken as a reference, and the price for various Kumori overlay setups is compared to this reference. In order to facilitate an easier understanding of this table, red cells indicate that the kumori overlay setup operational cost is higher than for a long-haul links connectivity. At the opposite, the green cells indicate that the kumori overlay setup operational costs are lower.

Table 7.8 – Cost comparison between the Kumori architecture and a private link connectivity strategy. Prices are given as $/month

Setup	Private connectivity	Kumori, Peering infra setup, full transit	Kumori, Peering capacity increase, full transit	Kumori, Peering infra setup, transit/ peering mix	Kumori, Peering capacity increase, transit/ peering mix
Google's B4	289 100	634 804,44	568 471,11	370 612,44	304 279,11
Facebook	79 933,33	158 813,33	131 846,67	125 021,33	98 054,67
Amazon	1 532 733,33	3 157 146,67	2 800 791,11	1 707 674,67	1 351 319,11
Global	1 901 766,66	3 571 306,67	3 148 951,11	1 972 330,67	1 549 975,11
B4 Europe	62 400	226 417,78	170 484,44	144 497,78	88 564,44
B4 Asia	1 516 800	1 608 337,78	1 481 004,44	797 329,78	669 996,44
B4 Northern America	157 800	222 417,78	169 484,44	189 649,78	136 716,44
B4 Southern America	1 084 800	2 935 057,78	2 795 724,44	1 263 889,78	1 124 556,44
B4 Australia	310 800	3 098 897,78	2 959 564,44	1 624 337,78	1 485 004,44

The first observation we can make from this table is that a CSP willing to use the Kumori architecture should be ready to negotiate peering agreements to lower the cost for relaying traffic at the Kumori inflection points. Indeed, in Kumori's operational costs, the cost for relaying network traffic at the inflection points accounts for a large share of the total cost. Besides, considering the results we obtained for the B4 topology in Asia and in Northern America, we can see that the benefit for using Kumori's architecture

increases with the difference in price between transit and long-haul links. Indeed, if we consider an inter-datacenter network with the same private link density as the B4 topology, we observe that using the Kumori's architecture becomes profitable when the transit to long-haul links cost ratio is below 2.2, as it is the case for the B4 Asia topology (from 0.98 to 2.18) or for the B4 Northern America topology (1.86). This price difference between transit and long-haul connectivity also explains the good results obtained with the Kumori architecture for the Amazon topology and for the global topology.

7.4.3 Effects of chronological price reductions on the Kumori architecture's operational costs

As explained in section 7.2.5, the price of the various components of the Kumori architecture's operational cost varies from one year to another in a disparate manner. Indeed, transit price drops more rapidly than the price of equipment and of long-haul links. Figure 7.1 shows the effect of this price reduction over five years for five connectivity strategies studied in section 7.4.2 and for four inter-datacenter topologies: Google's B4, Amazon, Facebook and our global imaginary topology. In this comparison, we assume that network traffic does not need to increase. The observation of the cost evolution curves outlines that the use of the Kumori architecture tends to be more interesting in the future. We explain this phenomenon by the fact that the transit price decrease rate is higher than the decrease rate of long-haul links' prices. As a result, the transit to long haul links cost ratio diminishes year after year, and goes under 2.2. If we combine this effect with the increase in network traffic demand in the future, we foresee that the Kumori architecture's profitability will be higher in the future than it is today in some global regions.

7.5 Conclusion

Besides its technical possibilities, the Kumori overlay network architecture can be used profitably by Cloud Services Providers to replace the long-haul link pairs used up to now to connect remote datacenters under certain conditions. Our economical study reveals that the Kumori architecture's economical profitability depends on the transit to long-haul link cost ratio when the Kumori architecture is used by a single CSP. Indeed, the lower the cost for transit, the more profitable the Kumori architecture. Today, the northern American and Asian connectivity markets expose properties that make the Kumori architecture an appropriate solution to use from an economical perspective. Given the price trends for both transit and long-haul links, we expect that the Kumori architecture will become more profitable in the future. This profit should be even more noticeable , with the increase in the demand for network capacity between datacenters.

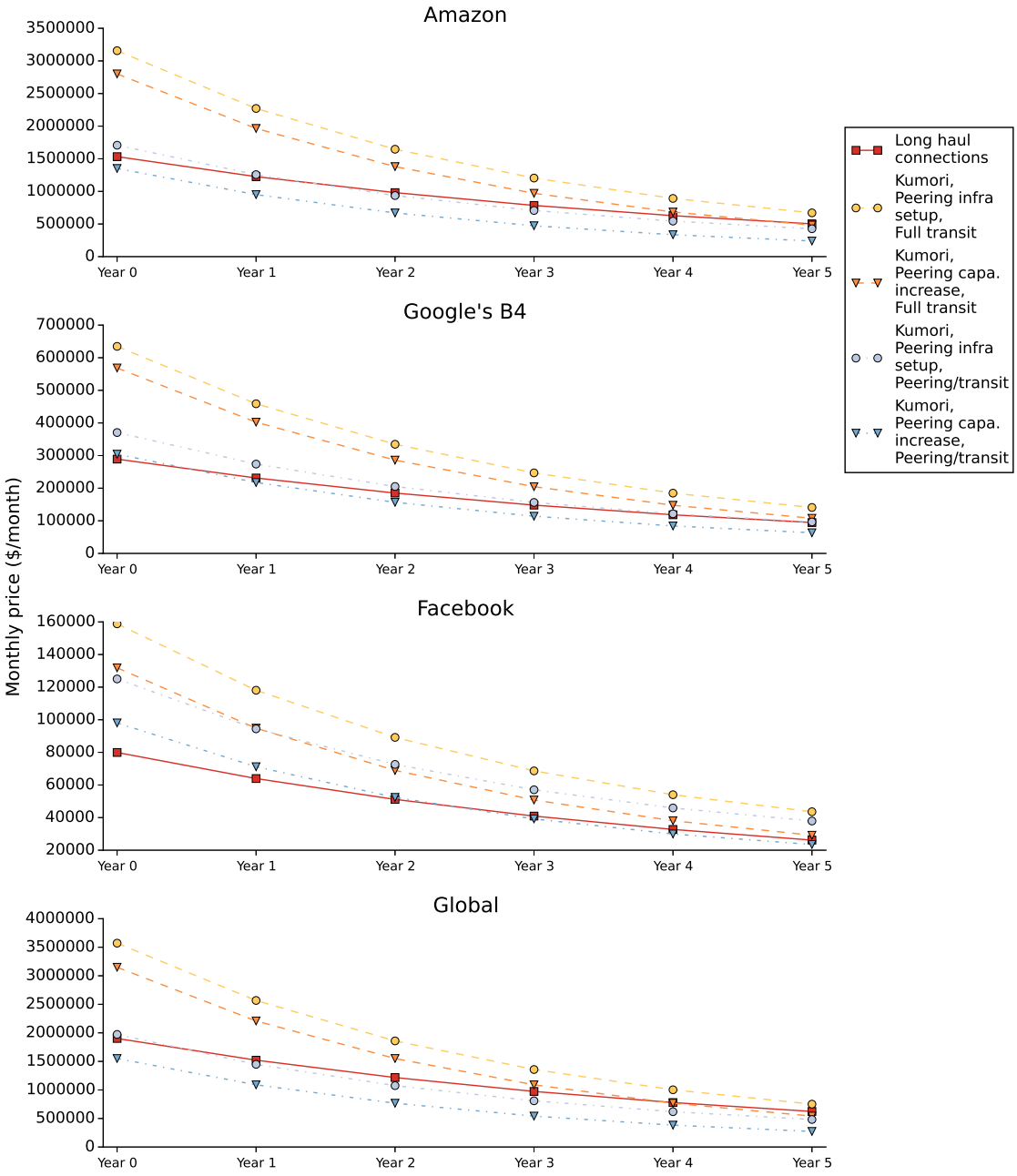


Figure 7.1 – Average number of path found among Amazon's PoP pairs depending on the routing inflection point choice policy

Chapter 8

Conclusion

Contents

8.1 Outcomes

The aim of this thesis has been to contribute to the vast and still open problem of Cloud resiliency. Major Cloud Services Providers (CSP) manage multiple datacenters distributed at the scale of a continent, or even the planet. Each of these datacenters may host several hundreds of thousands of servers and millions of virtual machines (VM) simultaneously. This thesis is focused on the resiliency of this type of very large Cloud infrastructures. The larger the size of the computing and networking infrastructures of a CSP, the higher the economy of scale provided by statistical multiplexing inherent to VM provisioning. Large datacenters are interconnected via IP networking facilities and long-haul fiber links. Both these types of resources are rented by the CSPs to telecoms operators or built on purpose in the case of very large CSPs. Continuity of service is a key performance indicator for a CSP. In this context, multiple factors may induce a deny of computing service. At the physical layer, a fiber cut, the failure a laser transmitter or of a photo-detector, etc. induce *de facto* an interruption of the computation of the impacted jobs. A job may also be interrupted due to logical layers malfunctioning (for instance in case of erroneous addressing or buffer overflow at an IP router). In addition to these external factors, a Cloud service disruption may also occur at the level of the backplane of the datacenters. In this thesis, we only focus on the resiliency of the inter-datacenter networking infrastructure. Our objective has been then to propose a solution to the problem of service discontinuity in case of physical or logical link failures thanks to the resiliency of inter-datacenters connectivity. According

to the state of the technology, existing approaches to setup long-haul IP networks connectivity are far too long to be commercially acceptable for Cloud's clients, and by consequence, for the CSPs. Indeed, such approaches induce installation delays from a few weeks to a few months. Besides, in those approaches, resiliency is a contractual service provided by the network connectivity providers. The efficiency of the operators to restore their infrastructure may vary considerably from one operator to another, and depends on Service Level Agreements (SLA) contracts. Such a dependency on the network connectivity providers is totally incompatible with the expectations of CSPs. In order to solve this problem, we have proposed in this thesis that the CSPs may get the possibility to control by themselves the resiliency of the inter-datacenter networking infrastructures. Our main objective has been then to reduce the delays required for the setup and provisioning of inter-datacenter connections.

Considering this set of network topology design constraints, we have introduced in this thesis a novel overlay architecture, Kumori. As mentioned previously, we have chosen the term "Kumori" in reference to the Japanese word meaning "Cloud", after a research internship in Tokyo. The Kumori architecture consists in a SDN-based network overlay. The Kumori overlay's coverage spans both the intra-datacenter and the inter-datacenter domains in order to steer network traffic flows from a given server located in a datacenter to any other server operated by the CSP. For its inter-datacenter part, the Kumori architecture consists in a set of routing inflection points. These inflection points are located at Internet exchange points that correspond to "operator-neutral" locations in the Internet. Such points benefit from the presence of a rich connectivity ecosystem. Inspired from SDN architectures, the operations of the Kumori overlay nodes are coordinated by a central controller. This controller is responsible for gathering measurements from the various overlay nodes. Those measurements are then used by the controller to detect link or node failures. Once such a failure is detected, the controller can instruct the overlay nodes to detour the concerned traffic.

We have evaluated the performance of the inter-datacenter part of the Kumori architecture in terms of path length, path diversity and number of required nodes in comparison with existing overlay designs. First, we compared the Kumori architecture with the Resilient Overlay Network (RON), a well-known edge-oriented overlay network. For that purpose, our metric has been based on the length of the necessary paths. This evaluation shows that the Kumori approach is more advantageous than RON for various types of CSPs. For large, global CSPs such as Amazon, Microsoft or Google, the lengths of the alternative paths provided by the Kumori approach and the RON approach are of the same order of magnitude. The number of nodes required in the overlay to access those paths is lower in Kumori than in RON. This result is particularly interesting because one of the major drawback of the RON architecture is its incapacity

to operate properly when more than fifty nodes are involved in the overlay. For smaller CSPs such as Atos or Dimension Data, the Kumori architecture provides significantly shorter alternative paths than RON. Thus, for those CSPs, the use of alternative paths provided by Kumori results in a lower performance degradation than with RON.

We then evaluated the resiliency provided by Kumori's inter-datacenter overlay. To do so, we compared the path diversity provided by Kumori to the Internet path diversity. To perform this second evaluation, we built a directed graph representing the Internet at the PoP level. This graph takes advantage of up-to-date datasets from the iPlane, DRAGON and PeeringDB projects. In this graph, the edges are directed and tagged to represent the complexity of the relationships between autonomous systems in the Internet. We used this graph to find and evaluate the number of edge-disjoint and node-disjoint paths for all the pairs of PoPs belonging to two CSPs, Amazon and Atos, representing the two CSP groups we identified in our first evaluation. The numerical results we have obtained outline that the Kumori architecture generally improves the number of diverse paths between two CSP PoPs. A detailed analysis underlines that the benefits of the Kumori's approach are more important for PoP pairs for which the topological diversity is limited. This observation stresses the potential of the Kumori overlay approach for smaller CSPs who are typically in this case. Besides, the comparison of the path diversity obtained for multiple Kumori setups with a varying number of routing inflection points stresses that Kumori's performance depends less on the number of routing inflection points used than on their placement at geographically diverse Internet exchange points.

At last, we evaluated Kumori from an economical perspective. We calculated the cost to pay to operate a Kumori overlay, and we compared it to the cost of operating a private protected connection mesh for several typical inter-datacenter topologies. This numerical evaluation highlights that the profitability of the Kumori architecture strongly depends on the ratio between the price of transit and the cost of long-haul links, since the major part of the cost of the Kumori approach is related to the architecture's connectivity. From the numerous numerical results we have obtained along this thesis, we can conclude that the Kumori architecture is economically interesting in Northern America and in Asia. At last, we also show that the price dynamics of both transit and long-haul private connections will increase the economical profitability of the Kumori architecture in the near future in more global regions.

8.2 Perspectives

In this thesis work, we have focused our efforts on the design and on the evaluation of Kumori's properties and performance on models of the Internet graph. In order to

pursue our work on Kumori, we intend to implement a prototype of this architecture. For that purpose, we shall use a SDN controller to implement the operational logic of each type of Kumori node as an SDN application running on a controller (for instance the Ryu SDN framework [Ryu]).

After this development work, we intend to test our prototype on an emulated network. The purpose of this test is to assess that we can effectively redirect network traffic and quickly react in case of a detected failure. We plan to test several options highlighted in Section 3.3. Indeed, we plan to evaluate the encapsulation mechanism we could use to redirect traffic among Kumori's nodes. Besides, we intend to test the load generated by the monitoring traffic sent by the Kumori nodes on the controller. The evaluation of the failure detection delay from those measurements could also deserve our attention. More precisely, we are planning to use either Mininet [Min] or GNS3 [GNS] to emulate an inter-datacenter Kumori overlay. In this test on an emulated environment, we need to model the traffic flowing between different datacenters managed by different CSPs. To the best of our knowledge, inter-datacenter network traffic has not been described with enough details in the literature. Thus, building such a traffic model would be very useful for our evaluation, and beyond for the research community working on inter-datacenter traffic optimization. Meanwhile, prior to such investigations, a permanent and attentive review of the results that could be published in this matter will have our priority.

In the Kumori architecture, we assume that it is feasible to control the Kumori nodes with a central controller from a rather long distance across the Internet. This hypothesis is rather strong if we consider that in the state of the art, SDN controllers are used to foster dataplane resiliency within a single network's domain. In the Kumori architecture, the latency and potential jitter on the link between the controller and the routing inflection points might have a strong impact on the architecture's capacity to react to failure events. This impact should be observed on a testbed deployed in the Internet. Besides, this testbed should be used to assess whether the headers and options used in the encapsulation method chosen to steer the network traffic among Kumori's nodes are filtered by middleboxes or proxy elements. These possible extensions in the continuation of our work could probably justify in themselves, another PhD.

In the design of our overlay, we make the assumption that increasing path diversity at the IP layer between datacenters enhances the resiliency of datacenter interconnections. Nevertheless, recent work such as [CCGF14] highlight the massive emergence of remote peering technologies. Those technologies allow ASes to connect to IXPs remotely using layer 2 technologies. Those connections are invisible from the graph representations we used to evaluate path diversity in the Internet in Chapter 6. According to [CCGF14], remote peering is used by as much as 20% of the members of some large IXPs. Given the characteristics of remote peering, we might have overlooked those peering links in the construction of our PoP-level topology. The impact

of remote peering on the path diversity provided by the Kumori overlay is uncertain, and should be studied in the future.

At last, we estimate that the PoP-level directed graph we have considered in this thesis to evaluate the Kumori architecture could be improved. Indeed, in our work, we have used two generic clustering methods. We know that the router connections at a POP follow specific architectural patterns. Thus, revisiting our router clustering method to use a pattern recognition algorithm could enhance the accuracy of the Internet model we have developed. In this matter, the benefits of the recent advances in matter of data analytics and Big Data could be applied. In the research community, one of the faiths of Internet modeling efforts is that the generated models are alas not updated once the research project is over. We estimate that either the algorithm implementation or an actualized PoP-level Internet model should be shared openly and regularly with the research community.

Part II

Manuscrit en français

Chapitre 1

Introduction

Contents

1.1 Contexte

Les technologies de l'information et de la communication (TIC) ont été fortement impactées par l'émergence du concept dit de *Cloud Computing*. Une définition de ce concept a été proposée il y a presque dix ans par Ian Foster [FZRL08] :

"*Un paradigme de calcul distribué à grande échelle qui vise à la réalisation d'économies d'échelle, dans lequel un pool virtualisé, abstrait et dynamique de ressources de calcul et de stockage est utilisé à la demande par des clients via l'Internet.*"

En d'autres termes, grâce au *Cloud Computing*, les entreprises ou les utilisateurs peuvent louer voire acheter de la puissance de calcul, de la capacité de stockage ou des services logiciels externalisés en fonction de leurs besoins. Avec le *Cloud Computing*, les utilisateurs finaux n'ont pas à investir dans leur propre infrastructure ou dans des licences logicielles pour répondre à leurs besoins informatiques. Depuis son émergence il y a dix ans, le *Cloud Computing* a fortement influencé l'industrie informatique. Ce modèle d'approvisionnement rapide et à la demande pour les services informatiques promet de couvrir les besoins des utilisateurs de manière élastique.

Le *Cloud Computing* défini par Ian Foster prend ses racines dans le concept d'informatique utilitaire, un terme inventé par John McCarthy en 1961. Il a gagné en

popularité avec le développement et la généralisation des applications Web accessibles via un navigateur. Grâce à ces applications Web, les logiciels sont vendus sous forme de services plutôt que téléchargés, installés et payés à la licence. Ce modèle est appelé *Software as a Service* (SaaS). Au milieu des années 2000, Amazon a appliqué un modèle similaire pour vendre des infrastructures ou des plates-formes informatiques via son offre Amazon Web Services. Amazon a aussi lancé l'offre Elastic Compute Cloud en 2006. Cette offre commerciale propose des services accessibles à tout utilisateur professionnel, universitaire ou privé connecté à Internet. Le concept de **Cloud public**, ou *Public Cloud* fait référence à ce mode d'approvisionnement de services Cloud très ouvert. Il suppose que, *a priori*, une même ressource matérielle ou logicielle peut être partagée dans le temps afin de traiter successivement des tâches de différents clients. Les principaux fournisseurs de services Cloud (ou Cloud Services Providers, CSP) tels que Google, Amazon, OVH, IBM, Microsoft *etc.* exploitent des datacenters de grande taille permettant de gérer sur un même site jusqu'à plusieurs centaines de milliers de serveurs. Des millions de machines virtuelles peuvent ainsi être activées ou désactivées simultanément sur ces serveurs.

Après une première période de scepticisme, le modèle du *Cloud Computing* est devenu courant. Alors qu'au début, le *Cloud Computing* était utilisé pour répondre à des besoins informatiques secondaires, les entreprises tendent à utiliser ces services pour leurs besoins critiques. Les exigences d'élasticité associées à la nécessité de garantir un certain niveau de confidentialité à l'utilisateur final ont conduit à l'émergence de nouvelles architectures de systèmes Cloud. Ces architectures sont présentées sous le nom de **Cloud privé**, ou *Private Cloud*. Ces nouvelles architectures s'appuient sur la virtualisation des ressources matérielles et logicielles des datacenters. Le Cloud privé et le Cloud public utilisent tous deux la flexibilité apportée par la virtualisation ainsi que des techniques d'orchestration sophistiquées pour répondre à des besoins dont l'évolution peut être très rapide. Dans le Cloud public, les ressources matérielles affectées aux utilisateurs finaux sont génériques dans la mesure où elles sont imposées par le CSP. En d'autres termes, le même ensemble de ressources matérielles et logicielles est proposé à tous clients d'un CSP donné. Au contraire, le Cloud privé permet non seulement d'isoler les ressources matérielles (et éventuellement logicielles) affectées à chaque client, mais aussi de personnaliser ces ressources en fonction des besoins. Étant donné que le Cloud privé utilise les mêmes technologies, les mêmes interfaces et les mêmes modèles de données que le Cloud public, une infrastructure cloud publique peut être utilisée pour décharger une partie du travail effectué par des Clouds privés en cas de charge importante. Ce fonctionnement en débordement est appelé **Cloud hybride**, ou *Hybrid Cloud*.

Au fur et à mesure que le *Cloud Computing* gagne en maturité, des standards émergent *de juro* ou *de facto*. Cette standardisation permet à plusieurs fournisseurs de

services Cloud (CSP) de proposer des interfaces et des méthodes similaires pour accéder à leurs services. Ces services standards peuvent alors être utilisés de façon interchangeable. Ainsi, les différents fournisseurs se font concurrence sur le prix, la disponibilité ou la qualité de leurs services (QoS). La QoS fournie par un CSP peut être mesurée en termes de rapport coût-efficacité. Pour un travail donné, cette efficacité peut être exprimée en fonction des délais de traitement maximums et moyens. On doit également tenir compte de la robustesse des ressources matérielles ou logicielles fournies par le CSP à l'utilisateur final.

Outre les grands CSPs qui exploitent d'énormes centres de données interconnectés à l'échelle mondiale (IBM, Google, Facebook, Amazon, Microsoft *etc.*), des centaines d'acteurs de petite ou de moyenne taille sont également apparus. En raison de la disparité d'échelle entre ces deux types d'acteurs, deux approches ont été observées ces dix dernières années afin de renforcer la position des "petits" CSPs. Ces deux approches sont connues sous le nom de *Cloud Brokering* ([DS14]) et de *Cloud Federation* ([RBL+09]). D'un côté, le *Cloud Brokering* peut être comparé aux alliances commerciales qui permettent à différentes compagnies aériennes indépendantes d'opérer des vols à escales de concert, et ainsi de vendre des billets pour des destinations très diverses. Ainsi, le *Cloud Brokering* trouve sa justification dans le fait que l'union fait la force. Par ailleurs, un Cloud fédéré (ou *Cloud Federation*) associe des CSPs de petite et moyenne taille qui peuvent soit fournir le même type de services mais qui décident de coopérer pour mettre en commun leurs ressources à destination de leurs clients, soit des services complémentaires. À cette fin, les partenaires d'une même fédération acceptent de partager leurs ressources matérielles et logicielles. Des modèles commerciaux spécifiques restent à concevoir pour le Cloud fédéré.

Malgré ces développements récents, la résilience du *Cloud Computing* reste à ce jour un problème ouvert par plusieurs aspects. Cette résilience est indépendante de la manière dont un CSP gère l'activation ou la désactivation des machines virtuelles (VM) qu'il déploie sur les machines physiques de son datacenter. Elle peut être assurée en renforçant deux aspects complémentaires de l'infrastructure. Pour le premier aspect, il s'agit de garantir la fiabilité des machines physiques du centre de données lui-même. Cette fiabilité dépend fortement de l'efficacité du processus de refroidissement des différentes baies dans lesquelles sont installées les machines physiques. Pour le second aspect, il s'agit de se prémunir contre une interruption de la connectivité entre les utilisateurs finaux et le centre de données, ainsi qu'entre les différents centres de données. Cette thèse porte sur ce deuxième aspect. Du point de vue du réseau, la résilience nécessite que les connexions entre les utilisateurs finaux et les centres de données hébergeant les infrastructures ou les applications utilisées par ces utilisateurs soient robustes. Ici, l'impératif de robustesse suppose que la connexion puisse survivre à une panne survenant entre les utilisateurs finaux et les centres de données. Ce type

de résilience repose sur l'absence de point de contention pour les connexions reliant les utilisateurs aux centres de données du CSP. La résilience de la connectivité est particulièrement importante dans le cas des services critiques tels que la banque ou la surveillance de sites de production industrielle.

Aujourd'hui, les CSPs majeurs assurent la résilience du réseau reliant leurs centres de données en déployant des liens privées entre ces centres [JKM+13]. Ces liens sont soit déployés spécifiquement par le CSP ou loués auprès de différents opérateurs. Ces liens sont doublés afin de mettre en œuvre une stratégie de basculement actif-passif. Cette stratégie présente plusieurs inconvénients. Tout d'abord, ce doublement des liens déployés peut être très coûteux pour les CSPs les plus petits. En effet, ce doublement nécessite un investissement initial important de la part du CSP afin de payer les frais d'installation du lien privé. En outre, la mise en place d'un tel lien privé entre des sites distants peut prendre plusieurs semaines ou plusieurs mois, ce qui est assez long pour certains CSP agiles. Par ailleurs, il faut signaler que, dans le contexte des Cloud fédérés, les CSPs de la fédération interagissent librement les uns avec les autres. Dans un tel contexte, il n'est pas économiquement viable d'établir une connexion privée entre chaque membre de la fédération. Une telle contrainte n'existe pas dans le contexte du *Cloud Brokering*.

1.2 Objectifs de la thèse

Dans cette thèse, notre objectif est de concevoir et d'évaluer la faisabilité d'une méthode alternative à la mise en place d'un réseau privé de liens redondés qui est utilisée par la plupart des CSPs pour assurer la résilience des connexions entre leurs datacenters. Cette solution alternative assurera la résilience, c'est-à-dire la résistance aux pannes de nœuds ou de liens, des connexions entre les datacenters des CSPs. Notre stratégie consistera à essayer d'offrir un large ensemble de chemins disjoints sur Internet sur lesquels le trafic réseau entre les datacenters pourra être dévié en cas de panne. Notre solution s'adressera plutôt à des CSPs d'importance restreinte. Il offrira à ces CSPs un intermédiaire entre l'utilisation de l'Internet "simple" et la mise en place d'un réseau privé de liens dédiés. Notre objectif est de permettre aux CSPs de contrôler la connectivité entre leurs datacenters. Ainsi, la solution que nous souhaitons concevoir doit être aussi indépendante que possible des différents opérateurs réseau, des fournisseurs de services Internet (FAI) et des accords de niveau de service (SLA) proposés par ces derniers.

Afin de satisfaire ces objectifs de conception, nous proposons "Kumori", une architecture de réseau *overlay* exploitant les principes du *Software-Defined Networking* (SDN). Nous avons choisi de nommer l'architecture ainsi d'après la traduction du mot

"nuage" en japonais pour indiquer qu'il cible les CSPs. L'architecture Kumori doit permettre à un CSP de contrôler le trafic réseau circulant entre ses serveurs à l'intérieur de chacun de ses datacenters et entre ses datacenters sur Internet. L'*overlay* Kumori diffère des réseaux *overlay* classiques conçus à des fins de résilience par deux caractéristiques importantes : tout d'abord, le routage au sein de l'architecture Kumori est contrôlé par un contrôleur logique centralisé inspiré du contrôleur des réseaux SDN ou du *Path Computation Element* des réseaux *Multiprotocol Label Switching* (MPLS). L'utilisation de cette entité centrale assure que les décisions de routage sont coordonnées au sein de l'*overlay* et qu'un état globalement optimal a été atteint. De plus, contrairement aux *overlay* classiques destinées à assurer la résilience, les nœuds utilisés par l'architecture Kumori pour contrôler le trafic réseau entre les différents datacenters sont situés à différents points d'échange Internet (IXP). Ainsi, les IXP sont des zones neutres au sein d'Internet. Ils offrent une connectivité d'une grande richesse comparativement à des nœuds périphériques sur Internet.

1.3 Structure du manuscrit

Le but de cette thèse est de décrire en détails l'architecture de Kumori, le réseau *overlay* que nous avons conçu pour faciliter l'interconnexion des datacenters des CSPs tout en garantissant la résilience de cette interconnexion en cas de pannes de liens ou de nœuds. Dans le reste de ce manuscrit, nous proposons une évaluation quantitative des performances que l'architecture Kumori permet d'atteindre. Nous évaluons également les caractéristiques économiques de l'architecture. Cette thèse est organisée comme suit.

Dans le présent chapitre (Chapitre 1), nous présentons le contexte de notre travail et introduisons plus en détail notre problématique ainsi que les exigences et contraintes de conception que nous souhaitons appliquer à l'architecture Kumori.

Dans le chapitre 2, nous décrivons la stratégie actuelle utilisée par les fournisseurs de services Cloud pour interconnecter leurs datacenters. Nous rappelons ensuite diverses applications du concept de *Software Defined Networking* (SDN) aux réseaux étendus (WAN) (Section 2.1). Dans la section 2.2, nous présentons et discutons les approches qui ont été précédemment proposées afin d'améliorer la résilience de l'Internet. La section 2.3 présente plusieurs représentations de l'Internet sous forme de graphes ainsi que différentes méthodes pour évaluer la diversité de chemin sur Internet à l'aide de ces représentations topologiques. Nous soulignons le manque actuel d'une représentation d'Internet dont la granularité serait au niveau des points de présence (PoP) des différents systèmes autonomes constituant Internet. Ce manque nous empêche d'évaluer correctement la diversité des chemins à ce niveau de granularité.

La section 2.4 met en évidence le riche écosystème de connectivité proposé par les points d'échange Internet (IXP). Nous analysons les avantages et les inconvénients de divers projets mettant en évidence les possibilités offertes par l'utilisation d'un contrôleur SDN aux IXPs. Nous soulignons dans la section 2.5 pourquoi les travaux précédents visant à améliorer la résilience de l'Internet ne répondent pas aux exigences de la solution alternative que nous proposons dans cette thèse.

Dans le chapitre 3, nous décrivons en détails l'architecture Kumori, un *overlay* supervisée par un contrôleur centralisé que les fournisseurs de services Cloud peuvent utiliser pour améliorer la résilience de leurs connexions inter-datacenter. La section 3.1 détaille les exigences et contraintes de cette architecture. La section 3.2 présente les éléments constitutifs de l'architecture Kumori et la manière dont ces éléments interagissent ensemble. La section 3.3 détaille les mécanismes qui peuvent être utilisés par les nœuds constituant l'*overlay* Kumori pour contrôler le trafic réseau d'un datacenter à un autre. La section 3.4 détaille les mécanismes permettant à l'architecture Kumori de détecter une défaillance et de la contourner. La section 3.5 résume les principales fonctionnalités de Kumori. Enfin, la section 3.6 présente les principaux indicateurs de performance selon lesquels nous évaluerons l'architecture Kumori dans les chapitres suivants.

Le chapitre 4 détaille une première évaluation quantitative des performances que l'architecture Kumori permet d'atteindre en comparaison avec les réseaux *overlay* classiques, tels que l'architecture RON (Resilient Overlay Network). La section 4.1 discute des paramètres que nous considérons dans notre comparaison de performance. Dans la section 4.2, nous présentons la méthodologie et l'ensemble de données que nous avons exploitées pour cette évaluation. La section 4.3 détaille les résultats de performance que nous avons obtenus. Nous observons que Kumori donne des résultats différents en fonction de la taille du CSP. Pour les CSPs de taille modeste, Kumori permet avec un nombre de nœuds équivalent d'accéder à un nombre similaire de chemins plus courts et plus diversifiés par rapport au chemin qui peut être atteint en utilisant un *overlay* classique. Pour les CSPs de taille plus importante, l'architecture Kumori réduit le nombre de nœuds requis pour accéder à un ensemble similaire de chemins diversifiés entre les datacenters du CSP.

Dans le chapitre 5, nous décrivons la méthode que nous utilisons pour construire une représentation graphique de l'Internet au niveau des points de présence géographiques des systèmes autonomes constituant Internet. La section 5.1 souligne les problèmes associés à la construction d'une telle topologie. La section 5.2 montre comment nous combinons plusieurs jeux de données pour constituer une représentation au niveau PoP de l'Internet. La section 5.3 présente les caractéristiques de la topologie résultante obtenue et donne des caractéristiques de haut niveau de ce graphe.

Le chapitre 6 détaille notre évaluation des bénéfices obtenus grâce à l'architecture Kumori en termes de diversité de chemins, et donc de résilience. Nous utilisons à cette fin le graphe représentant Internet construit précédemment. La section 6.1 précise la méthode utilisée dans notre évaluation, consistant en la comparaison du nombre de chemins diversifiés disponibles entre les paires de PoPs d'un CSP sur Internet de manière directe ou par l'intermédiaire de l'architecture Kumori. La section 6.2 décrit l'algorithme que nous avons utilisé pour évaluer la diversité des chemins dans les diverses configurations étudiées. La section 6.3 détaille notre méthodologie. La section 6.4 présente les principaux résultats obtenus. Nous y soulignons les avantages offerts par l'architecture Kumori en termes de diversité de chemins entre les paires de PoPs d'un même CSP. Cette amélioration est plus importante dans deux cas : pour les CSP de petite taille et pour les nœuds qui ne bénéficient pas d'une diversité de chemin topologique élevée.

Nous poursuivons notre évaluation de l'architecture Kumori dans le chapitre 7 où nous abordons les aspects économiques de cette architecture. La section 7.1 rappelle certains éléments sur l'économie des relations entre les opérateurs réseau et leurs clients dans le cœur de l'Internet. La section 7.2 détaille la structure des coûts des différentes méthodes d'interconnexion utilisées par les fournisseurs de services Cloud pour connecter leurs datacenters entre eux et au monde extérieur. La section 7.3 présente la structure de coût de l'architecture Kumori. La section 7.4 détaille notre comparaison des coûts opérationnels de l'architecture Kumori avec les coûts opérationnels d'un réseau d'interconnexion classique constitué d'un ensemble de liens privés et redondés entre les datacenters de différents CSPs. Notre comparaison met en évidence que la rentabilité de Kumori par rapport aux méthodes classiques dépend du rapport entre le coût du transit Internet et le coût des liens loués. Compte tenu de la dynamique des prix sur le marché et de l'augmentation du volume de trafic réseau entre datacenters, nous montrons que l'architecture Kumori deviendra de plus en plus attrayante dans les années à venir.

Le chapitre 8 conclut notre manuscrit. Nous résumons dans la section 8.1 les principales avancées que nous avons proposées dans cette thèse en matière de résilience des infrastructures Cloud. Enfin, la section 8.2 propose quelques perspectives pour la poursuite de notre travail.

1.4 Contributions

Au cours de cette thèse, j'ai eu l'occasion de présenter mon travail en plusieurs occasions sous la forme de communications publiques :

- Au Congrès DNAC en 2014, à l'occasion d'une présentation intitulée *"Vers un*

usage des SDN pour améliorer la connectivité inter-datacenter" ;

- Aux Journées RESCOM en 2015, au cours d'une présentation intitulée *"Évaluation de la diversité de chemins sur Internet : dúne granularité au niveau des AS à une vision au niveau des points de présence"*.

En outre, j'ai présenté trois articles de recherche dans des conférences à comité de lecture :

- *A SDN-based network architecture for cloud resiliency* [FG15], co-écrit avec Maurice Gagnaire, qui a été présenté à la *IEEE Consumer Communications and Networking conference* en 2015. Le travail présenté à cette occasion est décrit dans le chapitre 3 ;
- *Kumori : Steering Cloud traffic at IXPs to improve resiliency* [FPG16], co-écrit avec Cristel Pelsser et Maurice Gagnaire, qui a été présenté à la *12th IEEE International Conference on the Design of Reliable Communication Networks* en 2015. Cet article a reçu le "Best paper award" lors de cette conférence. Le travail présenté dans cet article est détaillé dans le chapitre 4.
- *"A Dynamic Offer/Answer Mechanism Encompassing TCP Variants in Heterogeneous Environments"* [FG14], co-écrit avec Maurice Gagnaire, qui a été présenté à la *International Conference on Advanced Networking Distributed Systems and Applications (INDS)* en 2014. Bien que les travaux présentés dans cet article s'attaquent à des problématiques du trafic inter-datacenter, ils ne sont pas directement applicables à la résilience du Cloud. C'est la raison pour laquelle nous avons choisi de ne pas faire référence à cette contribution dans ce document par souci de cohérence.

Chapitre 2

Description de la problématique et état de l'art

Contents

2.1 Description du domaine de l'étude

Le but de cette thèse est de concevoir une nouvelle architecture de réseau en superposition, ou *overlay*, permettant aux fournisseurs de services Cloud (CSP)

d'interconnecter leurs datacenters afin d'accroître la résilience de leurs infrastructures. Pour ce faire, il est nécessaire de mieux comprendre les avantages et les inconvénients des solutions qui sont aujourd'hui adoptées par les grands CSPs pour garantir cette résilience. La solution que nous proposons pour résoudre ce problème repose sur de multiples concepts inspirés par le *Software-Defined Networking* (SDN). Dans un premier temps, nous examinons brièvement les techniques de résilience existantes adoptées par les opérateurs réseau. Nous rappelons ensuite les principes de base du SDN appliqués aux réseaux étendus (WAN). Dans un deuxième temps, nous décrivons l'état de l'art en matière de résilience des réseaux. Ensuite, nous présentons des représentations et des méthodes existantes pour évaluer la diversité des chemins sur Internet. Enfin, nous précisons les questions restant ouvertes qui seront abordées tout au long de la thèse pour conclure ce chapitre.

2.1.1 Techniques existantes pour la résilience

Afin d'éviter que leurs services ne soient interrompus en cas de panne d'un centre de données entier (DC) ou si une catastrophe majeure touche une région entière, les CSPs déploient généralement leurs services dans plusieurs datacenters répartis dans le monde entier. Les services exécutés dans ces DCs distants sont synchronisés et sauvegardés à l'aide de liens réseau à haute capacité. De grands CSPs comme Amazon, Facebook ou Google construisent leur propre réseau à partir de liaisons par fibre optique qu'ils déploient ou louent aux opérateurs d'infrastructure pour interconnecter leurs DCs [JKM+13]. La figure 2.1 présente le réseau B4, le réseau inter-datacenter WAN de Google tel qu'il était en 2011. Cette infrastructure de liens longue distance privés s'étend sur trois continents pour relier les 12 datacenters globaux de Google. En raison de son coût intrinsèque, cette stratégie n'est accessible qu'aux quelques grands CSPs qui opèrent à l'échelle mondiale.

FIGURE 2.1 – Réseau B4 de Google : un réseau WAN inter-datacenter [JKM+13]

Pour leur part, les acteurs plus petits louent des liens privés dédiés ou des circuits MPLS auprès de grands fournisseurs de services Internet (ISP). Ainsi, ils créent souvent un maillage presque complet entre leurs DCs. Ce maillage complet est composé de liens redondés et en surcapacité fournis par plusieurs fournisseurs. La combinaison du surdimensionnement et de la stratégie multi-fournisseur est destinée à protéger les CSPs contre les défaillances d'équipements de réseau (liens et nœuds) entre les DCs.

Cette stratégie de connectivité utilisée par les CSPs pour assurer la résilience des communications entre leurs datacenters est difficile à appliquer pour sécuriser la connectivité des différents utilisateurs vers les datacenters. En effet, établir une connexion privée entre deux sites est une procédure longue et coûteuse. Les CSPs comme Amazon essaient de résoudre ce problème. Par exemple, l'offre AWS Direct Connect d'Amazon permet aux utilisateurs finaux de se connecter directement au réseau d'Amazon à partir d'un ensemble de points d'échange Internet ou IXPs [AWS]. Pourtant, dans ce cas, les clients d'Amazon doivent soit être présents à l'un des IXPs où Amazon est présent, soit compter sur un ISP pour atteindre ces IXPs avec une qualité de service (QoS) et une résilience suffisantes.

En outre, dans la plupart des cas, la résilience des connexions privées utilisées par les CSPs de taille modeste est assurée par les fournisseurs de services Internet (les opérateurs télécoms) qui sont souvent directement concernés par les pannes de liens. Une coupure de fibre qui se produit sur la couche physique implique en général des perturbations multiples au niveau de la couche application. La qualité du service, mesurée par le délai moyen et le temps maximum de transit des paquets entre les nœuds, ainsi que le délai de récupération moyen et maximal du flux de données (au niveau TCP) après une panne font l'objet d'un contrat entre les CSPs et leurs fournisseurs de services Internet. Ainsi, en cas de défaillance physique (optique, électrique ou radio), les CSPs s'appuient sur la capacité de leurs fournisseurs à respecter leurs engagements contractuels. Nous estimons que cette situation n'est pas satisfaisante pour les CSPs qui cherchent à contrôler au mieux leur propre infrastructure. La solution que nous proposons devra permettre aux CSPs de garder autant que possible le contrôle de la résilience de leurs connexions.

2.1.2 Utilisation des SDN pour les réseaux WAN

Au début des années 2010, alors que le *Cloud Computing* a gagné en popularité et en maturité, le monde du réseau a également assisté à l'émergence d'un concept majeur : le *Software-Defined Networking* (SDN). Le SDN prend ses racines dans les travaux de Martin Casado, Scott Shenker et Nick McKeown [MAB+08]. Il s'agit d'un concept technologique dans lequel une entité logiquement centrale, le **contrôleur**, gère la

politique de routage des paquets au sein du réseau contrôlé en transmettant ses instructions à l'ensemble des équipements du réseau. Dans le SDN, le comportement du réseau est programmable dynamiquement. Le contrôleur masque donc la configuration du réseau aux applications. À cet égard, le SDN peut être considéré comme un facilitateur de la virtualisation des réseaux. Le SDN a d'abord été présenté dans [MAB+08] comme un moyen de permettre aux chercheurs d'utiliser les réseaux de campus universitaires pour la recherche sur les nouvelles architecture réseau, dite *clean slate*. Aujourd'hui, le SDN est utilisé dans de nombreux autres contextes, allant de la mise en place de réseaux expérimentaux à l'opération de réseaux de production fortement chargés.

Depuis quelques mois, les idées associées au SDN sont appliquées à l'optimisation des réseaux étendus (ou WAN). C'est ce qui est appelé le *Software-defined WAN* (SD-WAN). le SD-WAN est utilisé par les entreprises ou les CSPs pour contrôler la façon dont leur trafic réseau est routé entre différents sites locaux ou branches de l'entreprise. Le SD-WAN est parfois présenté comme un intermédiaire entre l'utilisation de circuits MPLS ou de liens dédiées privées entre les branches locales, et l'utilisation de réseaux privés virtuels sur Internet entre ces branches. Dans le SD-WAN, la résilience et la qualité de service sont traditionnellement fournies par surdimensionnement, et par l'utilisation de liens Internet sur lesquels le trafic est routé. Dans ce contexte, le SD-WAN permet aux administrateurs réseau de discriminer le type de trafic qui circule sur les circuits MPLS ou par Internet.

Le SDN peut également être utilisé dans un contexte WAN pour diriger le chemin emprunté par les flux réseaux entre différents sites dans un réseau de liens dédiées. Par exemple, Jain *et al.* montrent l'utilisation du SDN pour contrôler le routage des flux réseaux entre différents datacenters connectés par l'intermédiaire du réseau B4, le réseau WAN de production de Google qui est composé de liens privés dédiés [JKM+13]. Ici, l'utilisation de SDN couplée à la capacité d'influencer le démarrage de certaines opérations sollicitant le réseau permet à Google d'utiliser certains liens dans son infrastructure WAN à près de 100% de leur capacité, alors que l'utilisation moyenne de ces liens atteint 70%. De tels taux d'utilisation sont bien supérieurs aux taux d'utilisation moyens observés dans le cadre de l'usage de méthodes classiques d'interconnexion dans lesquelles les liaisons WAN privées sont doublées. Dans le cas classique, par conception, chaque lien ne doit jamais être utilisé à plus de 50% de sa capacité.

L'exemple de Google montre qu'utiliser un contrôleur logiquement centralisé pour coordonner la façon dont les datacenters sont interconnectés sur de longues distances est un objectif réalisable. Dans cette thèse, notre objectif est d'utiliser un contrôleur centralisé pour coordonner la politique de routage et ainsi assurer la résilience des connexions inter-datacenter.

2.2 La résilience des réseaux : État de l'art

La résilience des réseaux peut être abordée selon les points de vue de plusieurs acteurs concernés : opérateur réseau, client, autorité régalienne... Néanmoins, la plupart des travaux effectués sur ce sujet adoptent systématiquement le point de vue des opérateurs réseau. Dans la suite de cette section, nous mettons en lumière certains projets de recherche qui peuvent être utilisés par les utilisateurs du réseau pour améliorer la résilience de leurs connexions. Plusieurs approches peuvent être adoptées. Dans un premier temps, nous examinons comment les réseaux superposés ou *overlay* peuvent être utilisés pour améliorer la résilience. Ensuite, nous analysons les avantages du multi-attachement ou *multihoming* (*i.e.* le fait que les datacenters ou les réseaux d'entreprise soient connectés simultanément à plusieurs opérateurs) ainsi que de l'usage simultané de multiples chemins ou *multipath*. Nous présentons par la suite plusieurs projets utilisant un contrôleur centralisé pour renforcer la QoS et la résilience. Enfin, nous décrivons brièvement des travaux antérieurs utilisant la relaxation de contraintes de routage sur Internet pour faciliter le contournement de pannes majeures.

2.2.1 Les réseaux superposés ou *overlay*

Le projet Detour [SAA+99] utilise un réseau *overlay* pour acheminer le trafic de données entre différents nœuds en utilisant des liens alternatifs au liens directs sur Internet avec une meilleure QoS. Dans leur article, Savage *et al.* observent que, statistiquement, pour 30 à 80% des cas de panne observés sur Internet, il existe pour chaque chemin direct entre deux nœuds sur Internet un chemin alternatif avec de meilleures caractéristiques en termes de bande passante, de perte de paquets ou de latence (RTT). A partir de cette observation, les auteurs suggèrent d'utiliser un réseau *overlay* dans lequel les nœuds sont connectés entre eux par des tunnels. Andersen *et al.* ont ensuite présenté le projet Resilient Overlay Network (RON) [ABKM01], qui consiste en une architecture de réseau *overlay* construite afin d'assurer la résilience des liens entre les membres du réseau. Dans ce projet, l'*overlay* est composée de nœuds qui mesurent activement les caractéristiques (bande passante disponible, latence, gigue) des liens qui les relient aux autres nœuds de l'*overlay*. Cette mesure active permet de réagir très rapidement en cas de panne. Détecter rapidement des pannes impose de poser des limites sur la taille, le nombre de nœuds participants à de tels *overlays*. Dans leur article, Andersen *et al.* montrent qu'au-delà de 50 nœuds, l'*overlay* ne peut plus être utilisée efficacement sans sacrifier la réactivité aux pannes.

Gummadi *et al.* [GMG+04] ont aussi abordé la question de la résilience des réseaux. Dans leur article, les auteurs soulignent que le routage par la source à un seul

saut peut être utilisé pour acheminer efficacement le trafic par delà la plupart des pannes affectant un chemin sur Internet entre deux nœuds distants. Un nœud aléatoire dans un *overlay* est utilisé pour détourner le trafic autour d'une panne de lien détectée. Le routage par la source est utilisé pour contrôler la façon dont le trafic est acheminé via ce nœud. Contrairement à RON, la surveillance permanente des liens n'est pas utilisée, mais l'approche proposée par Gummadi *et al.* devrait permettre d'utiliser des chemins alternatifs dans des délais plus courts que ceux observés avec les autres approches en cas de panne. Néanmoins, cette stratégie simple ne permet pas d'éviter les pannes affectant le dernier lien utilisé par une connexion entre deux points si le nœud de destination n'est pas attaché à plusieurs réseaux opérateurs.

Dans les projets que nous avons présentés jusqu'ici, les nœuds constituant le réseau *overlay* collaborent de façon décentralisée. Dans [HHL+09], Ho *et al.* présentent une approche dans laquelle l'utilisation d'un contrôleur centralisé permet de découvrir plus de chemins alternatifs au sein d'un *overlay* qu'en utilisant une approche décentralisée. En outre, dans leur projet, les auteurs soulignent les avantages de la prise en compte de la topologie de réseau sous-jacente dans la constitution d'un réseau *overlay* résilient. Des idées similaires ont été exploitées dans le projet Resilient Overlay for Mission Critical Applications (ROMCA) [ZP09]. Dans cette architecture, un nœud central est utilisé par les autres nœuds pour faciliter la découverte de nœuds voisins. Un protocole de routage décentralisé est utilisé pour découvrir et choisir les liens à utiliser entre les nœuds de l'*overlay*. La diversité des chemins au sein de l'*overlay* est régulièrement mesurée à l'aide de sondages par *traceroute*. Les stratégies de protection des liens et d'allocation des ressources aux différentes applications sont déterminées en utilisant le protocole RSVP-TE, utilisé au sein des réseaux MPLS.

2.2.2 Le multi-attachement ou *multihoming*

Outre les réseaux *overlay*, le multi-attachement ou *multihoming* peut être utilisé pour améliorer la résilience du réseau, comme le démontrent Akella *et al.* [AMS+03]. Dans leur article, les auteurs ont montré que se connecter à 4 opérateurs Internet suffit à une entreprise pour obtenir une résilience efficace à condition que les opérateurs choisis aient peu de chevauchement entre les chemins qu'ils proposent vers différents points d'Internet. Ce résultat permet aux entreprises et aux utilisateurs des services d'opérateurs Internet d'envisager de ne pas recourir à des contrats d'assurance de qualité de service stricts auprès de leurs fournisseurs de connectivité réseau tout en maintenant une bonne résilience. Les auteurs mettent en garde contre un risque lié à un choix trop dynamique de l'opérateur à utiliser pour se connecter à Internet, qui peut induire une instabilité de routage trop importante.

D'autres projets ont combiné le *multihoming* avec les approches présentées dans la

section **??**. Par exemple, dans [HWJ08], Han *et al.* ont conçu un réseau *overlay* dans lequel le *multihoming* est également utilisé afin d'améliorer les propriétés de résilience obtenues par les projets Detour [SAA+99] ou RON [ABKM01]. Dans le cadre de ces projets mêlant *overlay* et *multihoming*, les nœuds sont placés dans le réseau en tenant compte de la topologie du réseau. Les informations portant sur la topologie du réseau sous-jacent sont également utilisées dans la construction des différents chemins à utiliser pour contourner les pannes. L'utilisation du *multihoming* permet à un réseau *overlay* de contourner les pannes touchants les derniers liens d'un chemin donné. L'applicabilité de l'approche présentée dans [HWJ08] semble néanmoins compliquée dans un environnement où le réseau n'est pas sous le contrôle de l'opérateur de l'*overlay* attendu que le placement des nœuds est crucial.

2.2.3 L'usage simultané de multiples chemins ou *multipath*

L'utilisation d'un *overlay* et du *multihoming* augmente la résilience en permettant aux flux de données d'être redirigé sur un chemin alternatif en cas de découverte d'une panne sur un chemin sur Internet. Une autre approche peut consister à combiner plusieurs chemins ou liens dans une connexion virtuelle pour éviter les effets d'une panne sur un lien unique. Avec l'approche SmartTunnel [LZQL07], Li *et al.* ont proposé d'utiliser des tunnels logiques combinant plusieurs chemins pour assurer la résilience. SmartTunnel combine la correction d'erreur directe (FEC) avec l'usage simultané de plusieurs chemins pour faire rapidement face aux pertes de paquets que la correction d'erreur ne peut pas corriger. Les multiples chemins constituant un tunnel sont choisis pour être topologiquement divers.

2.2.4 La centralisation du contrôle

Le *multipath* est également utilisé dans le projet CORONET (Controller based Robust Network) [KST+12]. Le but de ce projet financé par l'agence DARPA est de concevoir une architecture de réseau évolutive et tolérante aux pannes supportant le *multipath*. Dans cette architecture réseau, un contrôleur SDN centralisé orchestre la manière dont le trafic est acheminé au sein du réseau. Même si ce projet vise à être appliqué dans le domaine des réseaux optiques, les concepts adoptés pour assurer la résilience peuvent être appliqués de manière assez générique. Un seul plan de contrôle est utilisé quel que soit le plan de données. Comme pour notre architecture, CORONET est conçu pour être utilisé dans le contexte d'application Cloud. Une différence notable avec notre approche est que CORONET n'est pas une solution indépendante de l'opérateur : en effet, CORONET utilise des informations sur l'état du réseau provenant des routeurs du réseau sous-jacent, en particulier les routeurs de réseau optique.

Plusieurs approches pour assurer la résilience des réseaux MPLS ont été étudiées et proposées dans cadre de drafts ou de RFC. Ces travaux ont été décrits dans deux articles. Dans [RI07], Raj *et al.* présentent les différentes approches utilisées par la communauté MPLS pour rerouter le trafic réseau très rapidement *(Fast Reroute)*. Le *Fast Reroute* vise à accélérer la récupération du réseau après la défaillance d'un lien ou d'un nœud afin d'atteindre l'objectif de temps de récupération de 50 ms spécifié par l'UIT-T pour la couche physique. Les approches présentées dans cet article utilisent des mécanismes des réseaux MPLS tels que les protocoles RSVP-TE ou LSP pour mettre en place, annoncer et réserver des ressources pour des chemins alternatifs à utiliser en cas de rupture de lien ou de panne d'un nœud. Les concepts présentés dans cet article ne peuvent pas être pleinement appliqués dans notre architecture Kumori car il est nécessaire de contrôler le réseau pour déployer la plupart de ces techniques de reroutage rapide. Dans [PCG+13], Paolucci *et al.* passent en revue les travaux de recherche effectués sur le *Path Computation Element* (PCE), un élément central des réseaux MPLS destinés à précalculer des chemins en fonction de critères donnés. Le PCE peut être utilisé par les opérateurs de réseau pour améliorer la résilience de leur réseau dans un contexte inter-domaine, *i.e.* dans un contexte où des réseaux MPLS opérés par différents opérateurs au sein d'ASs différents doivent coopérer. Le PCE consiste en un élément centralisé du réseau qui contrôle la politique de routage globale. Il est alors possible d'exploiter cette propriété du PCE pour assurer la résilience des réseaux MPLS. Les approches présentées dans [PCG+13] ne sont pas applicables à notre contexte car les administrateurs des différents réseaux MPLS doivent collaborer pour fournir des informations sur leurs réseaux respectifs. Ce n'est pas une option pour notre architecture puisque nous voulons permettre aux CSPs d'être indépendants des fournisseurs de connectivité réseau.

Depuis l'émergence du concept de SDN, plusieurs projets se sont attaqués à la résilience de ces réseaux, tant pour assurer la résilience du contrôleur que pour assurer celle du plan de données. Dans ce cas, l'objectif est d'atteindre une rapidité de restauration en cas de panne similaire à celle des réseaux opérateurs, soit typiquement moins de 50 millisecondes selon les normes UIT-T. Dans la plupart des projets visant à améliorer la résilience des contrôleurs SDN, chaque contrôleur SDN coopère avec d'autres contrôleurs situés dans son voisinage. En cas de défaillance d'un contrôleur, un contrôleur voisin peut prendre en charge les équipements habituellement gérés par le contrôleur défaillant. Dans le projet **DISCO** [PBL14], les réseaux SDN sont divisés en régions. Ainsi, chaque région est constituée d'un ensemble d'équipements réseau (routeurs ou switches) pilotés par un contrôleur unique. Les contrôleurs des différentes régions d'un réseau s'échangent régulièrement des messages pour synchroniser leur fonctionnement et mettre à jour leurs connaissances respectives des règles de routage qu'ils utilisent. En cas de panne du contrôleur d'une région, la gestion des équipements pilotés par ce contrôleur est redistribué aux autres contrôleurs situés dans les régions

alentours. Dans le projet **Orion** [FBG+14], les contrôleurs gérant un réseau SDN sont organisés selon un arbre hiérarchique. Ainsi, plusieurs contrôleurs à un niveau n de cet arbre sont supervisés par un contrôleur maître fonctionnant au niveau logique n + 1. Idéalement, un contrôleur maître au niveau n + 1 est situé au centre géographique du cluster des contrôleurs qu'il protège. Dans un même cluster, les contrôleurs communiquent avec les autres membres au moyen de messages réguliers. En cas de défaillance d'un contrôleur au niveau logique n, le contrôleur maître qui fonctionne au niveau logique (n + 1) de l'arbre désigne un contrôleur parmi les contrôleurs de niveau n pour prendre en charge les équipements qui étaient gérés par le contrôleur défaillant.

Dans le SDN, le plan de données doit être surveillé par le contrôleur central pour détecter les défaillances qui pourraient subvenir. Des techniques de surveillance passive et active peuvent être utilisées à cette fin. Le projet OpenNetMon [vADK14] et le projet Payless [CBAB14] proposent tous deux une méthode pour surveiller passivement les réseaux SDN. Dans ces projets, des messages OpenFlow sont utilisés par le contrôleur pour recueillir des statistiques sur les flux du réseau passant par les différents équipements du plan de données. Ces deux projets utilisent le fait que, dans OpenFlow, les règles envoyées par le contrôleur aux équipement de commutation et de routage pour préciser les règles de routage ont une durée de validité variable. Lorsqu'une règle expire, les équipements réseau envoient au contrôleur un message *FlowRemoved*. Ce message contient un ensemble de statistiques sur les paquets qui ont été routés au moyen de cette règle. Dans les projets OpenNetMon et Payless, la précision de la surveillance dépend de la durée de validité des règles fournies par le contrôleur. Aussi, une durée de validité courte sollicite le contrôleur de manière importante. Afin de réduire cette contrainte sur le contrôleur, les deux projets adoptent un mécanisme adaptatif pour allonger ou raccourcir les intervalles de surveillance selon que la situation est stable ou fluctuante. Alors que la surveillance passive a l'avantage de n'utiliser que les échanges du contrôleur avec les équipements réseau qu'il gère, ce mécanisme de supervision peut être un peu trop lent à détecter une panne soudaine. Afin de faciliter une détection plus rapide des défaillances dans les réseaux SDN, d'autres projets complètent cette surveillance passive par un sondage actif.
Dans [SSC+11] et dans [AAK14], une méthode de surveillance, la *Bidirectional Forwarding Detection* (BFD) décrite dans [KW10a] et dans [KW10b], est utilisée pour détecter les pannes de liens entre deux nœuds. Dans [SSC+11], la BFD est combinée à une stratégie de protection de chemin pré-calculée semblable à ce qui peut être fait dans le cadre du *Fast Reroute* des réseaux MPLS pour assurer une récupération rapide en cas de panne.

Ces applications des principes de résilience aux réseaux SDN sont très intéressantes pour notre travail. Pourtant, nous devons déterminer si ces approches sont adaptées à un réseau *overlay* englobant des sites potentiellement très éloignés.

En effet, dans un *overlay* global, deux points posent un problème : la latence entre les nœuds de l'*overlay* et le contrôleur, ainsi que le fait que les nœuds de l'overlay utilisent des tunnels et non des liens classiques plus facilement contrôlables.

2.2.5 Assouplir les règles de routage pour améliorer la résilience des réseaux

Plusieurs projets de recherche évaluent les avantages de nouvelles règles de routage sur Internet pour mieux faire face aux pannes. Dans [WZMS07], Wu *et al.* caractérisent la capacité d'Internet à résister à un ensemble de scénarios de panne spécifiques. Dans cet article, Le réseau Internet est représenté sous la forme d'un graphe dirigé de Systèmes Autonomes (AS) dans lequel un modèle de politique de routage classique, le modèle *valley-free* est utilisé. Dans leurs travaux, les auteurs soulignent la fragilité d'Internet dans certains cas. Ils présentent quelques assouplissements possibles des politiques de routage afin de remédier à certaines de ces faiblesses. Plus tard, dans [HCC+12], Hu *et al.* envisagent également de relâcher les politiques de routage pour permettre de faire face à des pannes liées à des catastrophes naturelles. Ils reconnaissent le rôle des points d'échange Internet dans la connectivité du réseau. Ils suggèrent de relâcher les politiques de routage à des IXP spécifiques et de mettre en place des accords de peering spécifiques pendant une période limitée entre certaines ASs afin de faciliter le processus de restauration suite aux pannes.

Cet état de l'art souligne les avantages potentiels que peut apporter une utilisation des réseaux *overlay* pour améliorer la résilience des connexions entre datacenters. Plutôt que de réparer les pannes, les *overlays* s'appuient sur leur capacité à détourner le trafic autour de problèmes potentiels pour assurer la joignabilité d'un nœud. Pourtant, dans ces *overlays*, la coordination entre nœuds est rarement centralisée, ce qui peut conduire à une utilisation sous-optimale des ressources disponibles. Par ailleurs, l'utilisation de concepts du SDN pour favoriser la résilience du réseau a également été abordée dans des travaux antérieurs. Néanmoins, dans ces projets, le contrôleur et les équipements qu'il contrôle sont situés dans le même réseau, et opérés par le même opérateur. L'utilisation de telles approches dans le contexte d'un overlay WAN pourrait être problématique.

2.3 Evaluer la diversité des chemins dans un graphe représentant Internet

Au-delà de la conception de l'architecture de réseau Kumori que nous proposons, une grande partie de notre travail a consisté à évaluer les avantages offerts par cette

architecture par rapport aux solutions alternatives présentées dans 2.2. Afin d'effectuer cette comparaison, nous avons regardé quels étaient les représentations d'Internet sous forme de graphe disponibles dans la littérature. Nous avons également regardé quelques-unes des méthodes proposées ces dernières années permettant d'évaluer la diversité des chemins sur Internet.

2.3.1 La topologie d'Internet et ses représentations

Après la forte croissance du réseau Internet au cours des années 1990, il est devenu plus approprié de considérer Internet comme un écosystème évolutif que comme une construction statique suivant des règles de conception simples [CMM99]. Plusieurs projets ont alors émergé afin de mesurer et caractériser la structure d'Internet. Les travaux du *Center for Applied Internet Data Analysis* (CAIDA) sont particulièrement marquants dans ce domaine. Il s'agit d'un organisme collaboratif (universités, opérateurs et institutions gouvernementales) qui favorise une meilleure collaboration entre tous ces acteurs pour assurer une croissance cohérente de l'infrastructure Internet. À cette fin, CAIDA propose un outil de mesure Internet de grande envergure connu sous le nom de **Archipelago** (Ark) [Hyu06]. Grâce à un ensemble de mesures actives, Archipelago aide les membres de CAIDA à avoir une bonne connaissance de l'état actuel de la topologie Internet. L'outil Archipelago est souvent utilisé dans des projets de recherche visant à fournir une vue à jour de l'infrastructure Internet. Ces représentations adopte la granularité des Systèmes Autonomes (AS). Outre Archipelago, d'autres initiatives de recherche telles que iPlane ou DIMES poursuivent un même objectif. Le projet iPlane [MIP+06] réalise des mesures quotidiennes des flux de trafic transitant par Internet. À cette fin, iPlane procède à une collecte quotidienne de mesures de trafic effectuées par des sondes Traceroute ou Paris Traceroute. Par conséquent, iPlane présente une vision quotidienne de l'état de la topologie d'Internet. Cette vision est parcellaire, et adopte le niveau de granularité des routeurs, bien plus détaillé qu'une granularité au niveau des AS. Par ailleurs, iPlane propose une représentation de l'Internet au niveau du point de présence (PoP) des opérateurs. Cette représentation est construite à partir des mesures Traceroute collectées par iPlane. IPlane regroupe les différentes adresses IP apparaissant dans les mesures Traceroute à l'aide des entrées DNS associées à ces adresses IP. DIMES [SS05] est un autre effort de la communauté de recherche pour étudier la structure et la topologie d'Internet. Ce projet de recherche utilise des mesures *crowdsourcées* effectuées par des agents logiciels exécutés bénévolement. Les agents DIMES effectuent des mesures Traceroute et des pings réguliers entre eux. Les résultats sont collectés par le projet DIMES pour constituer un ensemble de données similaire à iPlane. A partir du jeu de données de DIMES, Feldman [FS08] propose une méthode pour générer automatiquement des cartes la topologie Internet au niveau des PoPs. Cette méthode

est basée sur une analyse du délai observé dans les mesures Traceroute effectuées par les volontaires de DIMES entre deux adresses IP. Feldman applique sa méthode pour déterminer la structure du niveau de PoP des 100 plus grandes ASs et observer leur évolution semaine après semaine. Depuis 2010, un autre consortium connu sous le nom de RIPE NCC fournit un nouvel outil de sondage : RIPE ATLAS [RIP10]. Il se compose d'environ 9000 sondes hébergées bénévolement et rattachées à plusieurs réseaux autour du globe.

Les représentations d'Internet au niveau des routeurs ou des AS proposées par la communauté à ce jour présentent des lacunes. Les graphes représentant l'Internet au niveau AS représentent chaque AS en tant que nœud unique. Procéder ainsi revient à masquer la diversité de ces AS en termes de taille et d'étendue géographique. D'autre part, les graphes représentant Internet au niveau des routeurs mettent sur un même plan des liens très courts entre routeurs situés sur un même site et des liens à plus longue distance entre les points de présence géographiques des ASs (PoP). Puisque les ASs adoptent une architecture de connectivité hautement redondante au sein de leurs PoPs pour assurer la résilience au sein de ces PoPs, il n'est pas nécessaire de représenter les liens les plus courts pour avoir une vue pertinente de la diversité de chemin au sein des ASs. Enfin, les représentations Internet au niveau PoP existantes révélées dans notre étude de l'état de l'art ont toutes deux des lacunes. La représentation au niveau PoP proposée par iPlane est beaucoup trop détaillée pour être considérée comme une cartographie correcte : une comparaison des données d'iPlane avec des réseaux dont la topologie est connue (Géant, Amazon, IIJ) le souligne. En outre, même si la méthode utilisée par Feldman est intéressante, les données du projet DIMES ne sont plus disponibles ni mises à jour, et les dernières données sont trop anciennes pour donner une vue précise d'Internet aujourd'hui. Ainsi, dans notre travail, nous visons à construire un graphe représentant Internet au niveau PoP qui soit à jour.

Sur Internet, les relations inter-AS ne sont pas symétriques. Cela est lié à la nature commerciale des relations de peering ou de transit entre ces systèmes autonomes (AS). Gao et Rexford [GR00] ont décrit l'effet de cette asymétrie sous la forme d'un modèle de routage simple connu sous le nom de *valley-free routing*. Le modèle *valley-free* utilise une caractérisation des liens entre AS sur Internet : client-fournisseur, pair-à-pair ou fournisseur-client. Ce modèle a été remis en question car certaines annonces de routes sur Internet violant ses principes ont été observées [QMM07]. Ce modèle de routage est également critiqué car il ne rend pas correctement compte de la complexité des relations entre ASs dont le type de relation peut changer en fonction du temps ou de la région géographique où les AS sont mis en relation [RWM+11]. Néanmoins, ce modèle reste applicable dans une majeure partie des cas. Pour appliquer le modèle de routage *valley-free*, il est nécessaire d'inférer le type de la relation qui lie les ASs entre eux. Plusieurs algorithmes ont été proposés pour inférer

cette relation, basée sur la position relative des ASs dans les chemins [Gao01], sur les vues partielles du graphe Internet au niveau AS observé depuis différents points d'Internet [SARK02], [OPW+10] ou sur des heuristiques appliquées à un graphe au niveau AS simplifié [DBEH+07]. Compte tenu de l'importance des relations inter-AS dans le routage Internet, il est nécessaire de prendre ces informations en compte pour pouvoir déterminer quels sont les chemins routables sur Internet.

2.3.2 Caractérisation et mesure de la résilience sur Internet

L'un des objectifs initiaux du réseau Internet est de rendre le réseau résistant aux pannes de nœuds et de liens. Il est largement admis que la probabilité que des événements multiples de ce type se produisent en même temps est négligeable, sauf dans le cas d'une catastrophe naturelle (par exemple un tremblement de terre). Dans le contexte d'un environnement Cloud, sujet de cette thèse, pour assurer la résilience de connexions sur Internet il faut rediriger efficacement les flux concernés par une panne sur une route alternative non-affectée. Une telle déviation doit être réalisée dans un délai qui reste compatible avec la qualité de service attendue par les utilisateurs finaux. Dans la conception d'Internet, ce besoin a conduit à la décentralisation de l'intelligence de routage vers les bords du réseau. Au fur et à mesure de la croissance d'Internet, certains chercheurs ont travaillé sur la caractérisation de la résilience d'Internet. Dans [TMSV03a] et dans [TMSV03b], Texeira *et al.* ont traité cette question. Ils caractérisent la résilience du réseau d'un fournisseur d'accès Internet (FAI), Sprint ainsi que des topologies de réseau de FAI générées en utilisant Rocketfuel [SMW02]. Les topologies étudiées sont des graphes au niveau PoP, un PoP étant un emplacement géographique où un fournisseur d'accès Internet déploie un ensemble de routeurs. Ces travaux introduisent également la notion de chemins arc-disjoints et nœud-disjoints. Les performances des réseaux des FAI en termes de récupération après une panne sont quantifiées au moyen de la fonction de distribution cumulative (CDF) (ou fonction de répartition) du nombre de chemins alternatifs qui peuvent être trouvés entre les PoP. Bien que ce travail souligne l'importance d'examiner les réseaux au niveau des PoP pour déterminer leurs propriétés en termes de résilience, il est limité à l'étude de réseaux d'opérateurs, et n'envisagent pas les réseaux inter-AS. Plus tard, Rohrer *et al.* ([RS11], [RJS14]) ont proposé une autre méthode pour évaluer la diversité des chemins : le score de diversité. Selon eux, les mesures classiques de la théorie des graphes telles que le diamètre, le degrés des nœuds ou la centralité des nœuds ne peuvent pas décrire avec précision les propriétés de résilience d'un graphe. Pour contrer les limites de ces métriques classiques, ils définissent une mesure de diversité entre deux chemins qui est utilisée pour calculer un score de diversité. Ce score de diversité est compris entre 0 et 1. Il peut être utilisé pour caractériser la diversité de deux chemins, d'un ensemble de chemins entre deux nœuds ou de tous les chemins

d'un graphe entier.

2.3.3 Recherche de chemins routables au sein du graphe Internet

Afin d'évaluer la diversité des chemins entre deux points d'Internet, il est nécessaire de pouvoir trouver tous les chemins conformes à la politique de routage *valley-free* dans un graphe dirigé représentant Internet. En effet, comme il a été mis en évidence par Erlebach *et al.* [EHM+06], travailler sur une représentation Internet non dirigée ne permet pas de rendre compte de la complexité du routage BGP entre ASs. Dans cet article, les auteurs démontrent également que la recherche de chemins *valley-free* à nœuds disjoints dans un graphe Internet orienté au niveau AS est un problème NP-dur. Grâce à une transformation du graphe, les auteurs ont réussi à réduire la complexité du problème et à trouver une solution en temps polynomial à ce problème. Klöti *et al.* [KKAD15] suggèrent une autre méthode de transformation de graphe pour trouver des chemins AS *valley-free* sur Internet. Malgré l'intérêt des résultats présentés dans ces deux travaux, il est difficile d'appliquer ces transformations de graphe à un graphe Internet au niveau PoP. En effet, dans les graphes de niveau PoP, la présence de relations intra-AS rend difficile une transposition de ces transformations.

2.4 Les points d'échange Internet

Pendant longtemps, le rôle et l'impact des points d'échange Internet (IXP) sur Internet étaient peu connus. Les IXPs sont des points où les ASs acceptent d'interagir pour échanger du trafic sur la base d'accords de peering s'ils y trouvent un avantage mutuel. Le peering est en général associé à un modèle économique de compensation, où les opérateurs compensent financièrement le déséquilibre dans le trafic échangé. Au cours des dernières années, certaines études se sont concentrées sur la localisation des IXPs dans le graphe Internet [AKW09] ou sur la détermination des membres de ces IXPs [KAK+16]. Dans [ACF+12], Ager *et al.* prennent l'exemple d'un grand IXP européen pour montrer que le rôle des points d'échange Internet est sous-estimé. Ils montrent que les ASs font plus de peering à ces IXPs que les serveurs de route des IXPs ne tendent à le montrer. En outre, les IXPs et la proximité qu'ils facilitent entre ASs jouent un rôle important dans l'aplatissement d'Internet montré dans [DD10], [GILO11] ou [GALM08]. En effet, ces articles montrent qu'Internet est une construction moins pyramidale que ce qui était auparavant admis. Ils montrent que les réseaux de distribution de contenu (CDN) ou les grands fournisseurs de contenu sont présents aux différents IXPs pour échanger directement leur trafic avec les fournisseurs de services Internet régionaux, en contournant les grands opérateurs de niveau 1 ou (*tier 1*).

Du point de vue de la résilience, le rôle des IXP pour accroître la diversité des chemins a été souligné dans [HCC+12] ou dans [CHW+11]. Dans ces articles, les auteurs suggèrent qu'une relaxation temporaire de la politique de routage *valley-free* aux points d'échange Internet peut aider certains réseaux dans la récupération de leur connectivité après une catastrophe naturelle ou la défaillance d'un nœud. Les deux études utilisent un modèle asymétrique d'Internet, mais dans le graphe qu'elles considèrent, Internet est modélisé au niveau AS.

En outre, l'idée d'utiliser les emplacements neutres que constituent les IXPs pour contrôler le routage du trafic entre deux points d'Internet a déjà été exprimée. L'utilisation de l'infrastructure des IXPs pour déployer de nouveaux services est rendue possible par des projets tels que le *Software-Defined Internet Exchange* (SDX) [GVS+14]. SDX permet de combiner l'opération d'un peering traditionnel utilisant le protocole *Border Gateway Protocol* (BGP) avec l'utilisation d'un contrôleur SDN pour prendre en charge des cas de peering élaborés tels que le peering spécifique à une application, l'ingénierie de trafic entrant ou la redirection de trafic via des middleboxes. Dans ce cadre, SDX permet à plusieurs acteurs tels que les FAI, les CSP ou les opérateurs de transit de contrôler finement la manière dont leur trafic réseau est géré au sein des IXPs. En s'appuyant sur SDX, Kotronis *et al.* suggèrent dans [KKR+15] et dans [KKR+16] d'utiliser des points de contrôle et de surveillance situés aux IXP pour contrôler la façon dont le trafic est acheminé d'un nœud à un autre dans Internet et ainsi assurer la résilience ou la qualité du service d'un trafic réseau donné. Dans ces travaux, les auteurs font l'hypothèse que lorsqu'un AS est présent à deux IXPs, alors un "*pathlet*" existe au sein de cet AS entre ces deux IXPs. Ce *pathlet* peut avoir des propriétés de QoS spécifiques garanties par l'opérateur. L'utilisation de *pathlets* entre IXPs est une différence majeure avec la solution que nous avons proposée dans Kumori où nous ne souhaitons pas utiliser de liens à QoS garantie par un quelconque opérateur au sein de notre *overlay*.

D'un point de vue économique, la neutralité des IXP vis-à-vis des grands fournisseurs d'accès Internet et la possibilité d'utiliser des innovations technologiques comme SDX peuvent être utilisées pour introduire un nouvel acteur dans l'écosystème de connectivité Internet : l'*Overlay Services Provider* (OSP). Ce type d'acteur a été décrit et présenté dans [ZDA07], où Zhu *et al.* montrent la possibilité économique pour un OSP de fournir un meilleur service de connectivité que les ISP traditionnels en termes de QoS. Ce nouvel acteur pourrait fournir une meilleure QoS en positionnant des routeurs à différents IXPs et en sélectionnant dynamiquement le meilleur opérateur entre ces routeurs. A ce titre, les travaux de Zhu *et al.* ouvrent des perspectives quant au potentiel de l'architecture Kumori au-delà de ce travail de thèse.

2.5 Problèmes ouverts

L'état de l'art que nous avons présenté montre un décalage entre les objectifs que nous poursuivons dans la conception de l'architecture Kumori et les possibilités offertes par les travaux de recherche disponibles.

Tout d'abord, les solutions traditionnelles pour assurer la résilience des réseaux par la protection des chemins et les garanties de niveau de qualité de service nécessitent très souvent un contrôle sur les éléments de routage constituant le réseau. Aussi, ils ne peuvent être utilisés que par des opérateurs opérant des infrastructures réseaux inter-datacenter à grande échelle pour leur propre compte. Ces solutions sont souvent proposées aux CSPs pour les aider à sécuriser la connectivité de leur infrastructure. Dans un tel cas, les CSPs n'ont pas de prise sur l'opération effective des solutions utilisées pour assurer la résilience. Ils ne peuvent que s'assurer que les pénalités liées à un non-respect des obligations contractuelles de QoS soient suffisamment dissuasives. À notre avis, ce manque de contrôle opérationnel par le CSP est un inconvénient majeur des solutions de résilience traditionnelles.

Les CSPs peuvent avoir un meilleur contrôle sur le chemin emprunté entre les nœuds constituant leur infrastructure en utilisant des réseaux *overlay*, des stratégies de *multihoming* ou en utilisant des chemins multiples. Ces concepts souffrent de certains manques. Tout d'abord, ils sont déployés au bord du réseau, souvent aux datacenters des CSPs. Ces datacenters peuvent ne pas bénéficier d'un écosystème d'interconnectivité riche à partir duquel ils pourraient choisir leur fournisseur de réseau, selon l'emplacement du datacenter. Cette connectivité limitée des nœuds participants à un *overlay* limite la richesse des chemins qui peuvent être pris entre les nœuds de l'*overlay*. Dans Kumori, nous aimerions utiliser des nœuds superposés placés au niveau des IXPs afin que ces nœuds puissent bénéficier d'un grand choix d'opérateur. Nous nous attendons à ce qu'il en résulte une plus grande diversité de chemins, et donc une meilleure résilience.

Dans la plupart des réseaux *overlay* que nous avons présentés ci-dessus, la logique de routage est distribuée. Il peut en résulter une difficulté à atteindre des optimums d'utilisation globaux pour la connectivité des nœuds d'un réseau *overlay*, alors qu'un contrôle centralisé pourrait aider à remédier à cette inefficacité. Comme le montrent les travaux récents portant sur les réseaux SDN, cette centralisation du contrôle du réseau permet la mise en œuvre de cas d'usage intéressants que nous aimerions appliquer grâce à notre architecture Kumori.

Chapitre 3

L'architecture Kumori

Contents

3.1 Objectifs

Dans ce travail de thèse, notre objectif principal est de remplacer le maillage de lien privée utilisé le plus souvent par les fournisseurs de services Cloud (CSP) pour interconnecter leurs datacenters (qui a été présenté dans la section 2.1.1) par une architecture de réseau *overlay*. Cette architecture *overlay* tire parti du fait que la plupart des datacenters sont rattachés à plusieurs opérateurs pour accéder à Internet. Nous pensons qu'en utilisant une telle approche, le coût de mise en place ou d'opération de connexions résiliantes entre datacenters sera réduit. En outre, l'utilisation de cette architecture devrait accélérer le rattachement d'un nouveau datacenter à un réseau inter-datacenter existant.

En utilisant notre architecture, nous voulons atteindre le même niveau de résilience que le niveau atteint en utilisant un maillage de liens privés redondés, voire même un niveau supérieur. Nous ciblons un rétablissement de la connectivité d'un datacenter par

reroutage plus rapide que ce qui peut être atteint par l'utilisation de BGP. En effet, BGP permet de réaliser une telle opération en environ une minute. Notre objectif idéal serait d'approcher le niveau de performance du *Fast Reroute* dans MPLS. Néanmoins, nous ciblons plutôt un rétablissement en cas de panne en quelques secondes.

Dans les méthodes utilisées aujourd'hui pour assurer la résilience des connexions des datacenters, l'opérateur de réseau qui fournit la connectivité aux datacenters est seul à contrôler les actions mises en œuvre en cas de panne. En pratique, un CSP doit compter sur le fait que l'opérateur dont il est client respectera son contrat d'assurance de qualité de service (SLA) sans disposer d'aucun levier technique pour réagir à une panne observée. Avec notre architecture, nous voulons donner aux CSP le contrôle de leur connectivité réseau. Ainsi, notre architecture doit être aussi indépendante que possible des opérateurs de réseau. Cela se traduit par des exigences portant sur l'échange d'informations entre les CSP et les fournisseurs de services Internet (ISP) ou sur le placement des nœuds constituant notre *overlay*.

Notre dernier objectif est de permettre l'application de stratégies de résilience par flux ou par application déployée dans les datacenters. En effet, tous les types de trafic issus des datacenters n'ont pas une importance équivalente. Les CSP peuvent souhaiter appliquer des stratégies de résilience différentes en fonction de la priorité des différents flux, des exigences de certaines applications ou de leur perte de revenus en cas de panne. La conception de ces politiques différenciées dépasse le cadre de cette thèse, mais nous devons fournir des méthodes pour permettre leur mise en œuvre. Cette volonté de contrôler la résilience de chaque flux réseau s'accompagne d'un souhait de contrôler ces flux jusqu'au serveurs récepteurs finaux de ces flux au sein des datacenters. Cela implique de pouvoir contrôler ces flux à l'intérieur des centres de données et en dehors de ces datacenters sur Internet.

Pour résumer, nous essayons d'atteindre les objectifs suivants dans la conception de notre architecture Kumori :

- Donner au CSP le contrôle opérationnel de leur connectivité ;
- Fournir une solution à moindre coût par rapport à un maillage de liens privées ou de circuits MPLS entre datacenters ;
- Permettre de rattacher un nouveau datacenter plus rapidement qu'en utilisant les stratégies de résilience traditionnelles ;
- Contrôler la résilience du trafic réseau de façon dynamique et par flux ;
- Contrôler le trafic réseau de bout en bout, d'un serveur à un autre, à l'intérieur et à l'extérieur des datacenters du CSP.

3.2 Présentation de l'architecture Kumori

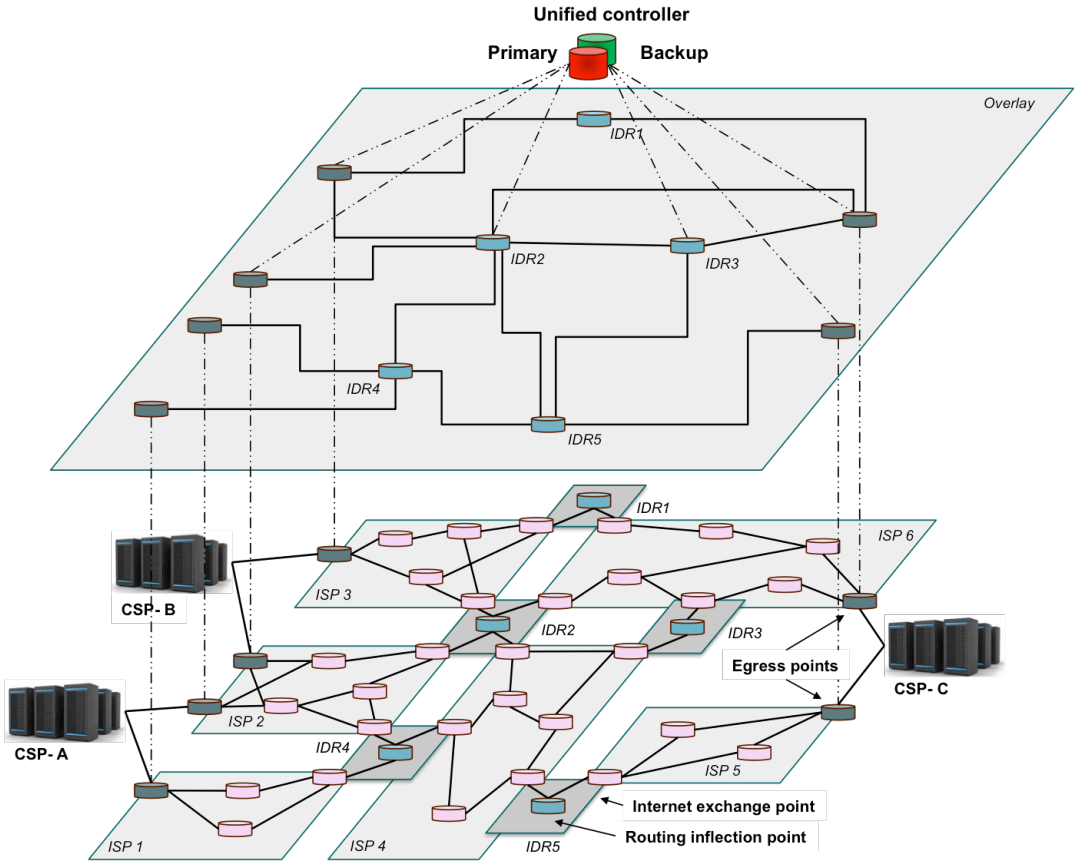


FIGURE 3.1 – Schéma de l'architecture Kumori dans le cadre de l'interconnexion de 3 centres de données. Le schéma représente les points de sortie, les points d'inflexion de routage et le contrôleur.

Dans la conception de Kumori, nous avons traduit ces objectifs en principes de conception détaillés. Tout d'abord, nous voulons que Kumori soit utilisé par les CSPs indépendamment des FAI qui leur fournissent leur connectivité. Cette exigence nous a conduit à l'utilisation d'une couche *overlay* réseau pour contrôler le trafic en dehors des datacenters du CSP. En outre, nous souhaitons utiliser des emplacements relativement neutres sur Internet pour placer les nœuds de l'*overlay*. Pour effectuer un contrôle dynamique des routes empruntées par les flux de réseau dans l'*overlay*, nous avons décidé d'utiliser un contrôleur centralisé pour gérer les différents nœuds. De cette façon, nous évitons d'avoir à concevoir ou à réutiliser un protocole de contrôle distribué pour lequel le temps de convergence pourrait être problématique, et nous centralisons les décisions de routage pour améliorer la capacité de contrôle du CSP. Enfin, notre souhait de contrôler le routage du trafic complètement d'un serveur à un autre nous a conduit à concevoir Kumori comme une architecture dotée de deux facettes. Certains éléments sont utilisés au sein des datacenters du CSP tandis qu'un *overlay* est utilisé

entre ces datacenters sur Internet. Dans les sections qui suivent, nous présentons chaque aspect de l'architecture Kumori.

3.2.1 À l'intérieur des centres de données

À l'intérieur de chaque datacenter, nous contrôlons le chemin emprunté depuis les différents serveurs jusqu'au points de sortie en utilisant soit une approche SDN soit un routage par segment (*Segment Routing*), en fonction des équipements déployés dans chaque datacenter. Comme présenté dans 2.1.2, le SDN est une technologie permettant à un contrôleur centralisé d'influencer ou de mettre à jour les politiques de routage appliquées par un ensemble d'équipements réseau via une interface de programmation (API) ouverte. Pour utiliser le SDN au sein d'un datacenter, les équipements qui y sont déployés doivent permettre ce contrôle. Si tel est le cas, Kumori peut utiliser les API pour contrôler le trafic réseau des différents serveurs du datacenter vers un point de sortie. Le SDN est une technologie relativement nouvelle, et elle est loin d'être déployée dans la majorité des datacenters. En outre, les équipements déployés aujourd'hui ne peuvent pas être mis à jour facilement pour supporter le SDN car la possibilité d'influencer le plan de contrôle d'un équipement en temps réel à partir d'une interface externe n'était pas prévue à la conception de ces équipements.

Afin de contrôler le trafic réseau au sein des datacenters qui ne sont pas prêts à utiliser le SDN, nous souhaitons utiliser le *Segment Routing* ou routage par segment (SR). Le SR est actuellement en cours de normalisation par l'IETF. Il s'agit d'une mise en œuvre des principes du routage par la source. Avec le SR, un nœud d'origine peut spécifier quels nœuds ou liens un paquet doit traverser à l'aide d'identifiants appelés identifiants de segment. Les routeurs échangent des informations sur les segments qu'ils gèrent en utilisant un protocole de routage à état de liens tel que OSPF. Dans notre architecture, le SR peut être utilisé à la place du SDN pour des raisons de facilité de déploiement. En effet, l'utilisation du SDN nécessite le remplacement des équipements des datacenters du CSP, alors qu'il est envisagé de pouvoir déployer le SR sur des routeurs déjà déployés au moyen d'une mise à jour du firmware de ces équipements. Cette capacité de déploiement couplée à la capacité de contrôler le trajet d'un paquet à partir de la source de ce paquet est l'une des raisons pour lesquelles nous considérons le SR comme une technologie appropriée pour accompagner le déploiement du SDN au sein des centres de données. Dans le SR, la liste des segments à utiliser pour un flux de réseau donné est donnée par un élément de calcul de chemin ou *Path Computation Element* (PCE) qu'on peut rencontrer dans les réseaux MPLS. Dans notre architecture, ce rôle est confié au contrôleur.

3.2.2 Entre les centres de données

En dehors de chaque datacenter, Kumori est constituée d'un ensemble de nœuds formant un *overlay* qui est contrôlé par un contrôleur centralisé. Les nœuds de l'*overlay* sont situés à différents IXPs. Ils peuvent également être installés au sein des réseaux de grands opérateurs Internet ou situés dans des nœuds de CDN (*Content Delivery Network*) pourvu que les accords commerciaux soient appropriés. Les nœuds sont déployés par le CSP, donc ne sont pas sous le contrôle d'un de ses fournisseurs de connectivité. Ils sont utilisés pour rediriger le trafic réseau entre deux datacenters autour d'une panne de lien ou suivant un itinéraire plus performant selon certaines métriques de QoS. Ils reçoivent les règles de routage du trafic du contrôleur centralisé. Ils effectuent des mesures régulières pour évaluer l'état des liens qui les relient aux autres nœuds de l'*overlay*. Le contrôleur centralisé est régulièrement informé du résultat de ces mesures. Ainsi, il peut déterminer la politique de routage la plus appropriée au sein de l'*overlay*.

Les datacenters sont connectés aux nœuds de *overlay* par un ensemble de points de sortie, les *egress points*. Chaque point de sortie est relié à un FAI qui connecte le datacenter à Internet, et à un seul. Ainsi, les nœuds de sortie sont de préférence situés à proximité de l'équipement gérant la connexion à ce FAI. Dans notre architecture, nous souhaitons que les CSP utilisent au moins deux FAI pour se connecter à Internet. En effet, comme nous l'avons vu dans [AMS+03] ou [HWJ08], cette stratégie de multi-attachement est un moyen d'assurer la résilience de la connectivité de bout en bout entre deux points d'Internet.

3.2.3 Description détaillée des éléments de l'architecture

Les points d'inflexion de routage (ou *routing inflection points*)

Les points d'inflexion de routage jouent un rôle essentiel dans notre architecture. Ils sont utilisés pour contrôler la façon dont un flux de réseau donné est acheminé via Internet entre différents centres de données. Ils peuvent détourner le trafic réseau d'une route utilisée par défaut et conforme à la politique de routage *valley-free*. Ils sont coordonnés par un contrôleur centralisé.

Dans Kumori, nous voulons que ces nœuds puissent contrôler le routage du trafic de données tout en permettant au CSP de garder un contrôle total sur ces nœuds. Pour cela, nos points d'inflexion de routage sont situés au sein de différents IXPs. En effet, les IXP sont des sites neutres où il est possible de bénéficier d'une connectivité très riche du fait des nombreux AS qui y sont présents. Nous pouvons donc y placer une plate-forme matérielle et négocier des accords de peering avec d'autres ASs. En outre,

les points d'inflexion de routage peuvent être également placés sur des sites de peering privées opérés par de grands FAI ou au sein même du réseau de grands FAI qui mettent à disposition de certains acteurs des sites où ils peuvent placer des équipements ou serveurs de cache spécifiques. Néanmoins, nous souhaitons que le CSP soit toujours en mesure de contrôler pleinement les opérations de ses points d'inflexion de routage.

Les points d'inflexion de routage effectuent des mesures régulières sur les liens qui les relient pour détecter proactivement les pannes. Nous souhaitons utiliser des méthodes non intrusives pour tester la disponibilité des liens de l'*overlay*. Les points d'inflexion de routage informent régulièrement le contrôleur central des résultats de ces tests. Ensuite, cette information peut être utilisée par le contrôleur pour apporter les modifications appropriées à la politique de routage globale.

Entre les points d'inflexion de routage, nous voulons éviter d'utiliser un mécanisme d'encapsulation ou de *tunneling*. Au lieu de cela, nous voulons que les nœuds de l'*overlay* utilisent un en-tête SR dans le cas où le trafic serait convoyé en IPv6 ou une règle de filtrage pour identifier les paquets appartenant à un flux de réseau donné et réécrire les entêtes IPv4 dans le cas où IPv6 ne pourrait pas être utilisé. Il est possible que certains opérateurs procèdent à un filtrage de l'en-tête SR au moyen d'une middlebox. Dans le cas où nous détecterions un tel filtrage, nous aurons besoin d'utiliser une méthode d'encapsulation.

Sous plusieurs aspects, les points d'inflexion de routage se comportent de la même manière que des équipements SDN. En effet, les équipements SDN peuvent compter les paquets appartenant à un flux de réseau donné, faire correspondre les paquets à un ensemble de règles de filtrage ou réécrire des paquets à la volée. En outre, les informations que les nœuds de l'*overlay* doivent communiquer au contrôleur centralisé peuvent facilement être convoyées par des messages OpenFlow [Ope13]. La principale différence entre notre architecture et un réseau SDN plus classique tient dans le fait que les nœuds de l'*overlay* ne sont pas directement connectés entre eux, et qu'ils sont gérés par un contrôleur situé à grande distance.

Les points de sortie des centres de données (ou *Egress node*)

Dans l'architecture Kumori, les points de sortie sont responsables du passage du trafic réseau des datacenters vers un fournisseur d'accès Internet donné et *vice versa*. Nous choisissons d'associer chaque point de sortie à un seul FAI pour éviter de créer des points de contention ou de faiblesse.

Au-delà de leur rôle de sélection du FAI à utiliser pour atteindre Internet, les points de sortie sont utilisés pour enlever l'entête SR utilisée à l'intérieur des datacenters et

relayer les paquets au sein de l'*overlay*. Pour ce faire, les points de sortie écoutent les annonces de segment faites à l'aide du protocole de routage interne aux datacenters. Ensuite, ils fournissent ces informations au contrôleur centralisé. Les points de sortie annoncent également des segments pour être joints correctement depuis l'intérieur du datacenter. De plus, ces segments annoncés peuvent également être associés à une politique de routage spécifique qui sera appliquée dans l'*overlay*.

Enfin, les points de sortie surveillent régulièrement les liens réseaux utilisés au sein des datacenters pour évaluer leur disponibilité. Les mesures sont prises depuis un ensemble de serveurs vers les points de sortie. Étant donné que les équipements utilisés à l'intérieur du datacenter peuvent ne pas avoir les mêmes fonctionnalités de surveillance que les commutateurs SDN, ils ne peuvent pas être utilisés pour tester chaque lien réseau. Pourtant, cette limitation n'est pas problématique car les administrateurs des datacenters ont souvent d'autres outils pour assurer la stabilité de leur réseau.

Le Contrôleur

Le contrôleur centralisé est le cerveau de l'architecture Kumori. Il contrôle la politique de routage au sein de notre architecture, à l'intérieur de chaque datacenters et dans l'*overlay*. Le contrôleur est responsable de la récupération des mesures effectuées par les nœuds de l'*overlay* afin de maintenir la politique de routage dans le temps et de s'assurer que les objectifs de résilience sont atteints. À cette fin, il est nécessaire que le contrôleur ait une vue à la fois sur le réseau interne des datacenters et dans l'*overlay* entre les datacenters, et qu'il contrôle la politique de routage dans ces deux zones.

Notre contrôleur centralisé hérite des propriétés et des fonctionnalités des contrôleurs SDN et du PCE des réseaux MPLS. À l'intérieur des datacenters, il fournit aux serveurs une liste de segments à utiliser pour acheminer le trafic vers les points de sortie compte tenu des exigences de QoS. Entre les datacenters, il met à jour régulièrement les règles de routage utilisées par les points d'inflexion de routage qui traduisent la politique globale de gestion du trafic et de résilience.

Comme cet élément central est critique dans l'architecture Kumori, il peut être considéré comme un point de faiblesse ou, à tout le moins, comme un goulot d'étranglement dans notre système. Il peut en résulter des problèmes de résilience. La résilience du contrôleur peut être obtenue en utilisant des méthodes de redondance classiques de l'industrie informatique. Plusieurs projets étudient ces questions dans la communauté de recherche sur le SDN. Ils adoptent soit des approches fédérales, soit des méthodes hiérarchiques.

Le projet *Distributed multi-domain SDN controllers* (DISCO) [PBL14] est un exemple

de distribution de contrôleur utilisant un modèle fédéral. Dans ce projet, les contrôleurs sont responsables de zones du réseau, consistant en un ensemble d'équipements réseau qui sont géographiquement proches l'un de l'autre, ainsi que du contrôleur de zone. Les contrôleurs se synchronisent entre eux en échangeant des messages de routage. DISCO propose une solution aux problèmes de résilience du contrôleur en configurant, pour chaque équipement de réseau, un contrôleur de sauvegarde en plus du contrôleur de zone principal. Ce contrôleur de sauvegarde est le contrôleur gérant une zone adjacente. Il est utilisé lorsque le contrôleur principal devient inaccessible.

À la différence du projet DISCO, le projet Orion [FBG+14] adopte une approche hiérarchique pour gérer les problèmes de résilience des contrôleurs SDN. Dans ce projet, les contrôleurs sont organisés selon un arbre hiérarchique. Les décisions portant sur la politique de routage globale sont prises au sommet de l'arbre, et les contrôleurs situés dans les branches de l'arbre définissent et fournissent aux équipements réseau des règles de routage plus fines.

Au cours de notre travail, nous n'avons pas eu le temps d'évaluer ces deux approches dans notre contexte. Ainsi, nous choisissons de laisser cette question pour des travaux futurs qui feront suite à notre projet.

3.3 Contrôle du trafic dans l'architecture Kumori

Dans Kumori, nous souhaitons utiliser un mécanisme approprié pour contrôler la manière dont les flux réseau sont acheminés au sein de l'*overlay*. De manière classique, des méthodes de tunneling sont utilisées à cette fin. En effet, le tunneling est largement utilisé dans le cadre de la virtualisation des réseaux, et plusieurs protocoles ont été développés pour encapsuler le trafic et le faire passer par des tunnels : le protocole *Virtual Local Area Network* (VLAN) [Jef14], le protocole *Virtual eXtensible Local Area Network* (VXLAN) [MDD+14], Le protocole *Locator/ID Separation* (LISP) [FFML13], le protocole *Stateless Transport Tunneling* pour la virtualisation réseau (STT), le protocole *Generic Routing Encapsulation* (GRE) , Le protocole *Generic Network Virtualization Encapsulation* (Geneve) [GGS16], ou le protocole *Generic UDP Encapsulation* (GUE) [HYZ16]. Ces protocoles diffèrent par l'en-tête d'encapsulation qu'ils utilisent, par le type de trafic qu'ils encapsulent et par leur maturité. Dans Kumori, nous aimerions encapsuler le trafic au niveau IP afin de construire un réseau virtualisé IP étendu. Ainsi, nous privilégierons des protocoles tels que GRE, GUE ou Geneve qui encapsulent le trafic IP dans des trames UDP.

Outre l'encapsulation du trafic d'un point de l'architecture Kumori à un autre, nous avons besoin d'une méthode pour permettre aux nœuds de l'architecture de déterminer comment router le trafic au sein de l'*overlay*. Plusieurs mécanismes peuvent être

utilisés à cette fin.

Tout d'abord, chaque nœud de l'*overlay* Kumori peut contacter le contrôleur chaque fois qu'il reçoit un flux de trafic qu'il ne connait pas encore. Cette méthode est classique dans les réseaux SDN où les équipements reçoivent du contrôleur des règles de routage pour les flux qu'ils doivent gérer. Cette méthode présente un inconvénient : lors de l'établissement du flux, la communication avec le contrôleur introduit un délai qui peut être très important pour un système de grande taille. L'impact de ce délai peut être limité en pré-installant des règles de routage de manière proactive dans les nœuds de l'architecture Kumori.

Pour éviter aux nœuds de l'architecture Kumori d'avoir à contacter le contrôleur centralisé lors de l'arrivée d'un flux réseau inconnu, nous pouvons utiliser les principes du routage par la source que nous avons présenté dans la section 3.2.1. Pour ce faire, nous pouvons utiliser l'en-tête SR dans les paquets circulant au sein de l'*overlay* Kumori. Ce cas d'usage de l'entête a été prévu par les concepteurs du *Segment Routing* dans [PFF+16]. Pour contrôler le routage du trafic dans l'*overlay*, le premier nœud (entrant) demande au contrôleur une liste de segments à utiliser pour un flux réseau donné. La liste des segments à utiliser pour acheminer le trafic est encodée sous la forme d'en-têtes SR dans les paquets envoyés. Le comportement des points d'inflexion de routage dépend alors des segments qu'ils reçoivent plutôt que d'une règle de routage reçue du contrôleur. L'inconvénient majeur de cette méthode est que l'en-tête SR ne peut être utilisé que sur un réseau MPLS ou sur un réseau IPv6. Malheureusement, nous ne sommes pas sûrs de la possibilité d'utiliser IPv6 entre les différents points d'inflexion de routage, même si le trafic IPv6 global est en croissance.

Récemment, Yong et Hao [YH16] ont proposé un mécanisme alternatif pour assembler, tisser des tunnels entre eux et créer un chemin entre deux nœuds au sein d'un *overlay*. Dans ce mécanisme, les tunnels sont identifiés par des identifiants uniques. Chaque nœud de l'*overlay* gère une table où les relations entre un tunnel entrant et les tunnels à utiliser pour relayer le trafic sont renseignées. Pour acheminer un flux réseau vers un nœud de sortie dans l'*overlay*, le nœud d'entrée encapsule le trafic vers le prochain nœud souhaité et indique les identifiants de tunnel suivant. A chaque saut, le trafic est décapsulé, l'identifiant du tunnel suivant est récupéré et le trafic est réencapsulé pour atteindre le saut suivant dans l'*overlay*. Ce mécanisme est un intermédiaire entre le contrôle total du trafic opéré dans le SDN et le routage par la source opéré dans le *Segment Routing*. En utilisant cette méthode, le nœud d'entrée peut complètement contrôler le chemin emprunté par les flux réseau au sein de l'*overlay*. Nous pensons que ce mécanisme peut être un mécanisme intéressant pour acheminer le trafic au sein d'un *overlay* relativement petit car il répond aux inconvénients des mécanismes de contrôle du trafic réseau utilisé dans le SDN et dans le SR. Le mécanisme de tissage de tunnels décrit par Yong et Hao nécessite que le

protocole de tunneling permette d'ajouter une option dans l'en-tête d'encapsulation. Comme les en-têtes Geneve [GGS16] ou GUE [HYZ16] peuvent être étendus en utilisant des champs facultatifs utilisant un formalisme du type type-longueur-valeur (TLV), ces protocoles d'encapsulation pourront être utilisés dans l'architecture Kumori.

3.4 Amélioration de la résilience avec l'architecture Kumori

Le principal cas d'usage de Kumori est d'assurer la résilience des connexions entre datacenters. À cette fin, nous devons être en mesure d'acheminer le trafic entre ces datacenters en utilisant plusieurs itinéraires disjoints, tout en permettant une détection rapide des pannes et une redirection rapide du trafic sur un autre chemin.

Notre architecture vise à assurer qu'à tout moment, plusieurs chemins peuvent être utilisés pour acheminer les flux de réseau entre deux points situés dans des datacenters distants appartenant à un CSP donné. Les datacenters sont connectés à Internet par l'intermédiaire de plusieurs FAI. Cette stratégie de multi-attachement permet d'éviter que le raccordement à un opérateur Internet constitue un point de faiblesse dans l'architecture. Au sein des datacenters, les serveurs et les points de sortie sont reliés par l'intermédiaire de plusieurs chemins. Dans l'*overlay*, notre objectif est de disposer de suffisamment de nœuds pour s'assurer que les datacenters soient connectés par plusieurs chemins disjoints. Pour s'assurer de la disjonction de ces chemins, nous cherchons à prendre en compte la topologie du réseau Internet dans le choix des nœuds de l'*overlay* et de leur placement, comme il est suggéré dans [HWJ08]. Dans le chapitre 6, nous allons présenter une méthode destinée à évaluer la diversité des chemins entre les PoPs des opérateurs que Kumori permet d'utiliser, ainsi que le bénéfice fourni par l'architecture par rapport au routage du trafic inter-datacenter sur Internet.

Les points de sortie et les points d'inflexion de routage de l'architecture mesurent régulièrement la qualité des liens et les paramètres des flux réseau qui les traversent. Au cours des dernières années, plusieurs projets ont étudié la surveillance des liens dans les réseaux SDN. Par exemple, le projet OpenNetMon [vADK14] présente une méthode de surveillance de l'état des liens utilisant les échanges de messages OpenFlow entre un contrôleur et les équipements qu'il gère. Dans cette méthode, Van Adrichem *et al.* utilisent le fait que les règles de routage de flux ont une durée de validité limitée pour permettre au contrôleur SDN de récupérer des statistiques sur les flux réseau. Les équipements SDN envoient ces statistiques au contrôleur par l'intermédiaire de messages OpenFlow de type *FlowRemoved*. OpenNetMon propose également un schéma de synchronisation adaptatif pour effectuer des évaluations précises des mesures de débit ou de gigue tout en évitant de générer trop d'échanges

OpenFlow lorsque la situation est stable. Dans le projet Payless, Chowdhury *et al.* utilisent également le protocole OpenFlow pour échanger des statistiques de trafic réseau avec une fréquence variable. Dans l'architecture Kumori, l'utilisation d'une méthode de surveillance du réseau similaire permettra au contrôleur centralisé de détecter des anomalies ou des pannes du réseau. Dans les projets OpenNetMon et Payless, la capacité à détecter les pannes rapidement tout en utilisant une stratégie de surveillance passive découle de la fréquence des échanges entre le contrôleur et les nœuds de l'*overlay*. Dans ces deux projets, un mécanisme d'adaptation permet d'atteindre un équilibre entre le souhait de réagir rapidement à différentes pannes et la nécessité d'éviter de surcharger le contrôleur ou de mal détecter une panne en raison d'un artefact de mesure.

Pour vérifier la réalité d'une pane détectée de manière passive ou pour raccourcir le temps de détection de ces pannes, d'autres projets préfèrent utiliser un mécanisme de mesure actif. Dans [SSC+11] et dans [AAK14], la surveillance passive utilisant les messages OpenFlow est complétée par l'utilisation du protocole *Bidirectional Forwarding Detection* (BFD) décrite dans les RFC 5880 [KW10a] et 5883 [KW10b]. Le protocole BFD est utilisé pour détecter les pannes entre deux nœuds d'un réseau donné. Il est conçu pour entrainer une faible surcharge tout en détectant les pannes rapidement. Dans [SSC+11], le protocole BFD est combiné avec une stratégie de protection de chemin pré-calculée pour pouvoir assurer un temps de récupération compatible avec les exigences des opérateurs au sein du réseau d'un fournisseur de services Internet. Dans Kumori, nous utiliserons BFD pour confirmer les pannes détectées en utilisant la surveillance passive même si nous ne ciblons pas un temps de récupération de moins de 50 ms.

Dans Kumori, la résilience est obtenue en fournissant un ensemble de chemins disjoints pour réacheminer le trafic en cas de panne. Le reroutage du trafic réseau autour d'une panne de nœud ou de lien est effectué dans notre architecture à l'aide du routage par segment à l'intérieur des datacenters et de l'*overlay* entre ces datacenters. À l'intérieur des datacenters, les serveurs sont informés d'une panne par le contrôleur centralisé qui calcule un itinéraire alternatif et qui leur fournit une autre liste de segments pour acheminer leur trafic. Entre les datacenters, les points d'inflexion de routage reçoivent un nouvel ensemble de règles de routage. En utilisant ces nouvelles règles, les nœuds de l'*overlay* peuvent réagir aux pannes détectées tout en préservant la connectivité des datacenters entre eux.

Le contrôleur centralisé joue un rôle essentiel pour assurer la résilience au sein de notre architecture. En effet, il est chargé de détecter les pannes réseau qui se seraient produites grâce aux mesures qu'il reçoit des différents éléments de l'architecture. Il dispose d'une vue globale de l'*overlay* et du réseau des datacenters. Ainsi, il est capable de calculer des routes alternatives et d'apporter les modifications appropriées

aux règles de reroutage appliquées au sein de l'*overlay*. Ce calcul de route alternative peut être effectué à la volée, ou à l'avance en considérant un ensemble de scénarios de panne potentiels. À cet égard, nous souhaitons réutiliser les concepts et les travaux réalisés par la communauté MPLS autour du *Fast Reroute* présentés notamment dans [RI07].

3.5 Résumé

Dans ce chapitre, nous avons présenté l'architecture Kumori, un *overlay* réseau contrôlé de manière centralisée et utilisable à la fois au sein des datacenters et entre ces datacenters sur Internet. À l'intérieur du datacenter, l'*overlay* est utilisée pour contrôler le routage du trafic réseau depuis un serveur vers le point de sortie le plus approprié. Les points de sortie sont les interfaces entre le datacenter et chaque fournisseur de connectivité utilisé pour accéder à Internet. Entre les datacenters d'un CSP, Kumori est constitué d'un ensemble de points d'inflexion de routage placés à différents IXPs. Les points d'inflexion de routage bénéficient d'une connectivité riche. Ainsi, ils peuvent utiliser un chemin approprié pour relayer le trafic vers un datacenter de CSP tout en évitant des pannes éventuellement détectées. Au sein de l'*overlay*, les décisions de routage du trafic sont prises par un contrôleur centralisé. Ce contrôleur collecte des informations sur l'état du réseau venant des points de sortie des datacenters et des points d'inflexion de routage pour détecter les pannes. Lorsqu'un tel problème est détecté, le contrôleur est chargé de coordonner la politique de routage au sein de l'*overlay* et d'envoyer des règles aux différents nœuds afin d'appliquer cette politique.

3.6 Les indicateurs de performance de l'architecture Kumori

Dans Kumori, nous avons tiré profit des enseignements de plusieurs initiatives de recherche antérieures présentées dans l'état de l'art technique. Pourtant, nous ne sommes pas sûrs que cette solution apporte un bénéfice notable aux CSPs prêts à l'utiliser pour améliorer la résilience des connexions entre leurs datacenters. Ainsi, nous devons évaluer les propriétés de Kumori et la capacité de cette architecture à s'intégrer à Internet aujourd'hui.

Tout d'abord, nous aimerions déterminer comment Kumori se comporte en comparaison avec des *overlays* classiques telles que RON [ABKM01]. Pour ce faire, nous souhaitons évaluer la **longueur** des chemins de contournement fournis soit par Kumori soit par RON entre les nœuds appartenant aux datacenters d'un CSP. Nous

voulons également déterminer si cette performance observée est similaire quel que soit le CSP utilisant l'*overlay* ou si certains facteurs ont un impact sur cette performance.

En outre, nous voulons nous assurer que Kumori améliore la résilience du réseau inter-datacenter pour les CSPs. À cet égard, nous voulons évaluer la **diversité de chemins** obtenue sur Internet et par l'intermédiaire de Kumori entre les différents nœuds d'un CSP. Ceci nous permettra de mesurer le gain en résilience fourni par notre architecture. Nous souhaitons effectuer cette évaluation pour plusieurs CSPs afin de déterminer si les bénéfices de Kumori sont homogènes ou si ils sont différents selon les CSPs.

Le coût d'opération et de maintenance d'un *overlay* dépend du nombre de nœuds nécessaires à l'opération de cet *overlay*. En évaluant la performance des chemins fournis et les gains en termes de diversité de chemins fournis par Kumori, nous déterminerons aussi le nombre de nœuds optimal pour obtenir une performance optimale de l'architecture Kumori.

Enfin, une nouvelle architecture *overlay* telle que Kumori doit être économiquement viable pour ses utilisateurs afin qu'elle puisse être déployée sur le terrain. Ainsi, nous évaluerons la rentabilité de Kumori en comparant son **coût d'opération** au coût d'une solution classique de connectivité entre datacenters.

Chapitre 4

Une première évaluation des bénéfices apportés par l'architecture Kumori

Contents

4.1 Indicateurs de performance utilisés

Dans notre évaluation, nous comparons Kumori à d'autres *overlays* réseau visant à renforcer la résilience des connexions sur Internet. La principale différence entre Kumori et ces autres *overlays* est liée à l'emplacement des nœuds de l'*overlay* : Dans Kumori, les points d'inflexion de routage sont situés à différents IXP alors que dans la plupart des autres *overlays*, les nœuds sont situés en bordure du réseau. Dans cette évaluation, l'*overlay* RON est utilisé comme point de comparaison pour notre architecture Kumori.

Dans cette comparaison, nous aimerions d'abord déterminer si Kumori et RON peuvent fournir des chemins alternatifs avec des caractéristiques de performance similaires. Pour cela, nous utilisons la latence sur ces chemins comme indicateur. Nous

comparons la latence des chemins les plus courts accessibles en utilisant chacune des deux architectures. En outre, nous voulons évaluer le coût de déploiement et d'exploitation de Kumori par rapport à RON. À cette fin, nous faisons l'hypothèse que ce coût dépend du nombre de nœuds participant à l'*overlay*. Nous comparons le nombre de nœuds nécessaires dans Kumori et dans RON pour accéder au même nombre de chemins.

4.2 Méthode d'évaluation

Dans cette première évaluation, nous avons choisi d'utiliser les données extraites de la base de données iPlane [MIP+06] le 15 février 2015. Cet ensemble de données présente deux avantages. Tout d'abord, contrairement aux simulations, il représente des liens réels qui ont été observés sur Internet ce jour-là. Même si l'ensemble de données ne représente pas l'ensemble d'Internet, il est plutôt significatif pour ce qui concerne le cœur d'Internet. De plus, à la différence de données extraites des serveurs de route BGP, iPlane peut montrer des liens de transit qui sont utilisés pour acheminer un trafic réseau réel alors qu'ils restent invisibles dans les tables BGP des serveurs de route. Ces deux avantages font d'iPlane un jeu de données intéressant à utiliser. Cependant, certains travaux sont nécessaires pour l'adapter à notre étude.

Les données iPlane brutes sont rendues accessibles sous la forme d'archives de mesures *traceroute* effectuées quotidiennement ainsi que de quelques fichiers de données synthétisées. Ces données synthétisées sont produites en assemblant toutes les données portant sur les liens entre routeurs observés un jour donné dans différentes mesures *traceroute* et en calculant une moyenne des latences qui ont été mesurées entre ces routeurs. L'une de ces synthèses de données donne la latence des liens entre les routeurs révélés dans les mesures iPlane, les taux de perte sur ces liens, l'association entre les adresses IP observées, les routeurs et les systèmes autonomes (AS) auxquels les routeurs peuvent être associés. Les mesures effectuées par iPlane sont réalisées tous les jours depuis le lancement du projet. Dans notre évaluation, nous utilisons des données synthétisant les mesures effectuées le 15 février 2015. En utilisant ces données, nous avons construit un graphe représentant les liens entre les routeurs qui ont pu être observés ce jour-là. Nous avons utilisé le langage de programmation Python pour analyser les fichiers de données iPlane et la bibliothèque logicielle igraph pour construire le graphe et le manipuler. Le graphe que nous avons obtenu est pondéré et non dirigé. Il comprend 190.028 nœuds représentant les routeurs et 916.390 arcs représentant les liens observés entre les routeurs. Nous avons choisi d'utiliser la latence moyenne observée du lien comme poids associé à chaque arc.

Dans les données iPlane, l'évaluation de la latence entre les routeurs pose deux

problèmes. Tout d'abord, si un échange de données a été observé sur un lien, mais que la latence n'a pas pu être évaluée avec précision, un coût négatif de -9999 est donné au lien. Or, la bibliothèque igraph ne permet pas d'associer un poids négatif à un arc. Pour résoudre ce problème, nous avons donné à chaque arc dont la latence calculée est négative un poids égal au double du poids maximum observé dans l'ensemble des données. D'autre part, certains liens ont une latence égale à 0. En effet, dans iPlane, les mesures sont arrondies à la milliseconde, et pour des liens très rapides, un coût nul est donné. Pourtant, ce coût nul ne tient pas compte du coût de commutation au sein des routeurs. Pour prendre en compte ce coût de commutation, nous donnons à chaque lien avec un coût zéro un coût minimal égal au double de la latence mesurée en faisant un *ping* sur l'interface interne d'un serveur Linux sous Ubuntu 14.04 LTS, soit 0,5 ms.

Une fois notre graphe pondéré non dirigé obtenu, nous avons besoin de repérer les routeurs appartenant aux différents CSPs considérés dans notre étude, ainsi que les routeurs participants à un IXP afin de rechercher et d'évaluer la latence des chemins entre les différents nœuds des CSPs.

Afin d'identifier les nœuds appartenant aux CSPs, nous avons choisi les principaux fournisseurs de services Cloud mondiaux à partir de données collectées par Gartner auxquelles nous avons ajouté des acteurs régionaux intéressants. Après cette première étape d'identification, nous obtenons une liste de 13 CSPs. Ensuite, nous avons recherchés quels étaient les ASs utilisés par ces CSPs en utilisant les outils d'analyse BGP de Hurricane Electric [HE-]. Nous avons ainsi obtenu une liste de 133 ASs intéressants. Enfin, nous avons cherché dans l'ensemble de données iPlane les nœuds appartenant à chacun de ces ASs.

L'identification des nœuds appartenant aux différents IXPs a été plus complexe. En effet, il n'existe pas de base de données centralisée fiable des IXP existants et, par conséquent, il n'y a pas de données sur les préfixes IP ou les ASs utilisés par les IXPs. Néanmoins, des données partielles peuvent être obtenues auprès de PeeringDB [Pee], une base de données où les administrateurs de différents réseaux fournissent de manière volontaire des informations sur leur politique de peering ; ou auprès de Packet Clearing House [**?**], un institut de recherche à but non-lucratif qui exploite des plateformes de mesure au sein de plusieurs IXPs à travers le monde. Nous avons d'abord nettoyé les données que nous avons trouvées dans ces deux bases de données et nous les avons associées afin de trouver les classes d'adresses IP utilisées par les différents IXPs. Ensuite, nous avons trouvé les nœuds iPlane qui étaient présents à un IXP en utilisant ces plages d'adresses IP.

Suite à cette phase d'identification, nous avons trouvé 1 604 nœuds appartenant à un des 13 CSPs considérés et 2 177 nœuds présents à un IXP sur les 190 028 nœuds du graphe. Dans la phase suivante, nous cherchons les plus courts chemins reliant ces nœuds identifiés pour comparer RON à notre architecture Kumori.

Pour évaluer correctement la capacité de RON et de notre architecture à acheminer le trafic entre les routeurs appartenant aux CSP, nous avons supprimé tous les arcs reliant deux nœuds appartenant au même CSP dans notre graphe. En supprimant ces arcs, nous nous assurons de comparer notre architecture à RON plutôt qu'à la stratégie d'interconnexion des CSPs. Ensuite, nous avons considéré les chemins les plus courts entre toutes les paires de routeurs des CSPs, entre les paires de routeurs des IXPs et entre les routeurs des CSPs et ceux des IXPs. Avec l'ensemble de ces chemins, nous avons comparé les chemins obtenus en utilisant l'architecture RON et les chemins utilisant les routeurs situés aux différents IXPs.

4.3 Résultats

Nous avons comparé l'architecture Kumori avec RON en deux temps : Tout d'abord, nous avons considéré un CSP imaginaire fédérant tous les nœuds que nous avons identifiés dans notre évaluation. Ensuite, nous avons analysé les résultats pour chaque CSP.

4.3.1 Résultats généraux

Dans cette section, nous examinons d'abord les résultats obtenus en considérant l'ensemble des nœuds appartenant à un CSP quel qu'il soit.

Tout d'abord, nos mesures montrent que les chemins accessibles en utilisant l'architecture Kumori ont une latence plus faible ou égale à celle des chemins fournis par un *overlay* tel que RON pour 1,255,950 paires de nœuds sur les 1 285 606 paires possibles. Cela représente 97,5% des cas. Si nous considérons une amélioration stricte des performances en termes de latence, notre architecture permet d'obtenir des chemins plus courts que l'architecture RON pour 73.511 paires de nœuds de CSP sur les 1.285.606 paires possibles. Cela représente une amélioration stricte de la performance dans 3.1% des cas. Ce résultat montre que notre architecture donne accès à des liens dont la latence est similaire à ceux que l'architecture RON propose dans la grande majorité des cas, mais ne démontre pas une amélioration drastique de la performance la plupart du temps.

4.3.2 Résultats détaillés

Après une première évaluation générale, nous comparons le nombre de nœuds nécessaires dans l'architecture Kumori et dans le réseau *overlay* RON pour acheminer le trafic de données entre les paires de nœuds des CSPs. La figure 4.1 présente un

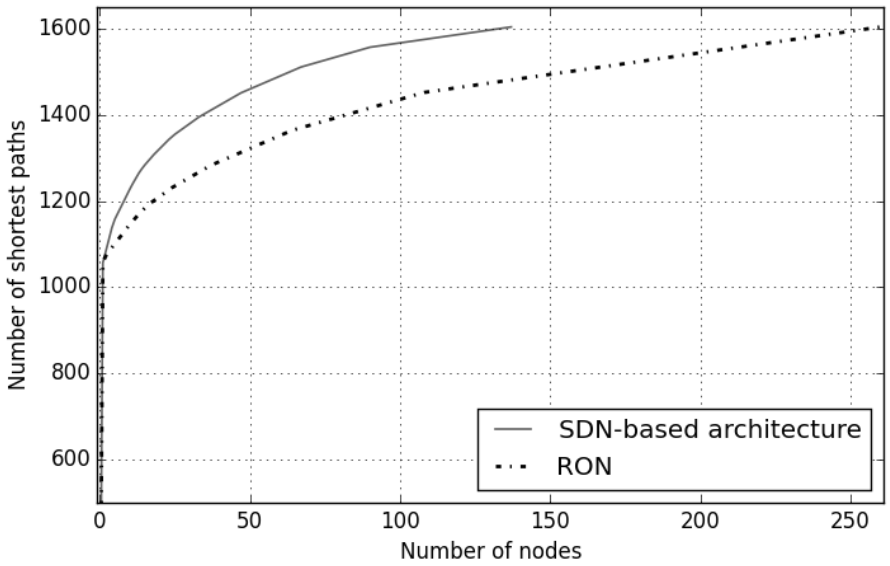


FIGURE 4.1 – Fonction de répartition du nombre de nœuds nécessaires pour accéder à un nombre maximal de chemins les plus courts entre les routeurs des CSPs.

graphe de la distribution cumulative du nombre de nœuds requis dans notre architecture et dans l'architecture RON. Cette figure montre que notre architecture recquiert 47% de nœuds en moins que l'architecture RON pour proposer un chemin alternatif à l'ensemble des paires de neouds des CSPs. Si l'on considère 80 % des paires de nœuds des CSPs, seuls 15 nœuds sont nécessaires dans Kumori pour proposer un chemin alternatif à ces paires tandis que 38 nœuds sont nécessaires dans l'architecture RON. Attendu que le coût d'exploitation d'un réseau *overlay* dépend du nombre de nœuds à opérer dans cet *overlay*, ce résultat montre que l'architecture Kumori est moins coûteuse à déployer et à exploiter qu'un *overlay* RON.

4.3.3 Analyse des résultats par CSP

Dans les résultats obtenus dans la section 4.3.1, nous avons considéré que les PoP appartenant aux 13 grands CSPs que nous avons sélectionnés appartenaient à un seul grand CSP. Pourtant, les CSP que nous avons choisi d'étudier ne sont pas homogènes, puisque le plus grand CSP en termes de nombre de routeurs dans notre échantillon compte environ 200 fois plus de nœuds que le plus petit CSP. Dans cette section, nous étudions les gains fournis par notre architecture pour chaque CSP de notre échantillon. Nous avons également inclus deux réseaux de recherche Cloud à fins de comparaison : WIDE et Géant.

Dans cette étude, nous comparons l'architectures Kumori avec un *overlay* RON, et nous utilisons deux métriques pour comparer ces architectures. Nous avons représenté les résultats que nous avons obtenus dans la figure 4.2. Sur cette figure, chaque point

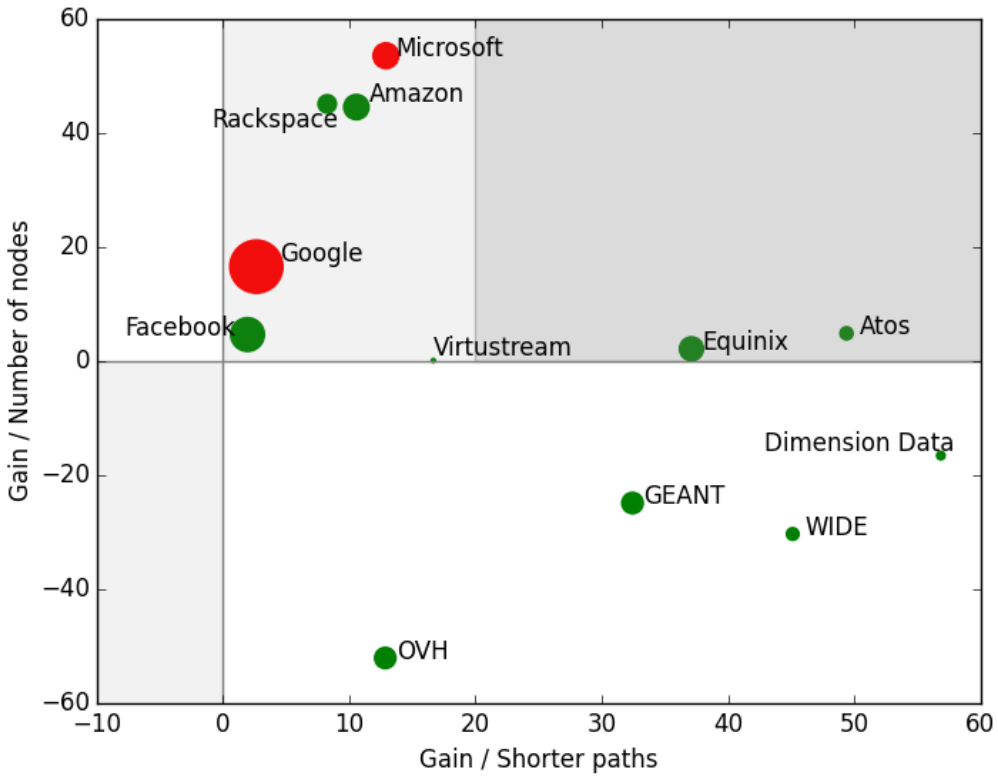


FIGURE 4.2 – Bénéfices de l'architecture Kumori par rapport à l'architecture RON en termes de longueurs des chemins les plus courts et de nombre de nœuds nécessaires pour y avoir accès.

représente un CSP spécifique. Le diamètre du point est proportionnel au nombre de routeurs associé au CSP dans notre graphe. Nous choisissons de colorer les points en rouge si l'architecture RON ne peut pas être utilisée pour des raisons de passage à l'échelle *i.e.* lorsque le nombre de nœuds nécessaires pour pouvoir proposer un chemin alternatif à toutes les paires de nœuds du CSP est supérieur à 50. Cette limite de taille a été précisée dans [ABKM01].

Premièrement, pour comparer les performances des deux architectures *overlay* que nous comparons, nous évaluons la proportion des paires de nœuds pour lesquelles l'architecture Kumori permet d'utiliser un chemin strictement plus court que l'architecture RON. Cette mesure est l'abscisse x des points représentant les divers CSPs sur la figure 4.2. Ensuite, pour comparer le coût de fonctionnement des deux architectures, nous avons comparé le nombre de nœuds nécessaires dans Kumori pour proposer un chemin alternatif à toutes les paires de nœuds d'un CSP avec le nombre de nœuds nécessaires dans l'architecture RON pour obtenir le même résultat. Nous calculons la différence entre les deux nombres et divisons cette différence par le nombre de nœuds utilisés dans RON. La proportion qui en résulte est l'ordonnée y des points représentant les différents CSP sur la figure 4.2.

Sur la figure 4.2, nous pouvons observer deux groupes de points distincts : un premier à droite de la figure, et un autre au milieu et en haut de la figure. Le premier groupe de points correspond aux CSPs les plus petits de notre échantillon ainsi que les deux initiatives de recherche Cloud, WIDE et Géant. Pour ces CSP, l'*overlay* Kumori permet d'utiliser des chemins plus courts que RON tout en nécessitant jusqu'à 20% plus de nœuds dans l'*overlay*. Au contraire, l'autre groupe correspond aux CSPs les plus importants de notre échantillon. Pour ces CSPs, notre architecture permet d'accéder à des chemins alternatifs similaires à ceux que propose RON en utilisant un nombre inférieur de nœuds dans l'*overlay*.

Nous expliquons ces résultats par la différence de stratégie utilisée par les différents CSP pour assurer leur connectivité. Les CSPs les plus importants ont optimisé leur architecture réseau pour se rapprocher des infrastructures de leurs clients finaux. Amazon propose par exemple à ses clients de se connecter directement à son réseau à partir de plusieurs points de présence grâce à son offre Direct Connect. Nos résultats montrent que la stratégie adoptée par ces grands CSP se traduit par une relative proximité des routeurs des CSPs avec les différents IXPs. Au contraire, les CSP de taille plus modeste comptent souvent sur un ou plusieurs opérateurs réseau pour accéder à Internet et joindre leurs clients. Par conséquent, leurs routeurs sont relativement plus éloignés des IXP que des routeurs appartenant aux CSPs de taille plus importante.

Dans cette comparaison entre Kumori et RON, nous pouvons constater que, dans tous les cas, notre architecture offre un bénéfice par rapport à l'architecture RON Pour au moins une des deux métriques considérées. Les avantages que procure notre

architecture sont très différents selon la stratégie de connectivité adoptée par le CSP. Ce résultat renforce notre souhait d'implémenter et de déployer notre *overlay* pour l'évaluer en conditions réelles.

4.4 Conclusion

Nos résultats montrent que les bénéfices que les CSPs peuvent obtenir en utilisant l'architecture Kumori dépendent de leur taille. Pour les CSPs les plus modestes tels que Dimension Data, les chemins rendus accessibles via notre architecture sont plus courts que ceux que peut proposer RON dans au moins 32% des cas. Pour les grands CSPs tels qu'Amazon, notre architecture donne accès à des chemins alternatifs similaires à ceux de l'architecture RON en utilisant jusqu'à 53% de nœuds en moins dans l'*overlay*. Ainsi, Kumori propose une solution à un problème d'adaptabilité à l'échelle majeur de l'architecture RON. Nous expliquons ces résultats par la différence entre les stratégies de connectivité adoptées par les CSPs. Les grands CSPs sont mieux interconnectés aux IXPs pour optimiser leurs coûts réseau tandis que les CSPs les plus petits dépendent encore de grands FAI pour connecter leurs datacenters à Internet.

Bien que cette première évaluation permette d'entrevoir un potentiel prometteur pour l'architecture Kumori, elle ne nous permet pas de mesurer correctement le gain en résilience fourni par l'architecture Kumori. Pourtant, l'amélioration de la résilience des connexions inter-datacenter est un des objectifs clés de Kumori. En outre, le graphe sur lequel nous avons effectué notre évaluation n'est pas dirigé, alors que la nature commerciale des relations entre les systèmes autonomes (AS) sur Internet se traduit par la nécessité de représenter Internet comme un graphe dirigé. Nous verrons dans les prochains chapitres de cette thèse comment nous avons réussi à dépasser ces limitations afin d'évaluer correctement le gain en termes de résilience fourni par l'architecture Kumori.

Chapitre 5

Construction d'une représentation d'Internet au niveau des points de présence des systèmes autonomes

Contents

5.1 Description du problème

La première évaluation de l'architecture Kumori présentée dans le chapitre 4 a souligné le potentiel de cette architecture. Les bénéfices associés à Kumori dépendent des caractéristiques topologiques du réseau des CSPs et de leur connectivité à Internet. Ainsi, nous avons observé que pour certains CSPs, l'architecture Kumori donne accès à des chemins alternatifs plus courts qu'une architecture *overlay* classique. Pour d'autres CSPs, Kumori donne accès à des chemins alternatifs similaires avec un nombre de nœuds réduit par rapport aux *overlays* classiques. Au-delà de ces améliorations, notre objectif est également d'évaluer les avantages proposés par l'architecture Kumori en termes de résilience des connexions entre les datacenters des

différents CSPs sur Internet. Dans notre approche, la résilience est obtenue en permettant aux nœuds d'un CSP de bénéficier d'une grande variété de chemins pour joindre leur destination. Aussi, offrir une plus grande diversité de chemin tend à augmenter la résilience du réseau en cas de pannes de liens ou de nœuds. En cas de panne réseau, intuitivement, plus les chemins alternatifs sont nombreux et diversifiés, plus il sera possible de contourner une panne de lien ou même d'un nœud donné.

Des travaux antérieurs ont étudié la diversité des chemins sur Internet au niveau AS. D'autres ont observé la diversité de chemins au niveau PoP au sein des réseaux de différents FAI. Dans notre étude, nous souhaitons évaluer la diversité des chemins entre les nœuds de différents CSPs au niveau des PoPs sur Internet. Cet objectif se justifie par le fait que les différents AS peuvent être très différents en termes de taille, de présence régionale ou de politiques de peering. Aussi, considérer les chemins sur Internet au niveau AS revient à mettre sur le même plan un AS représentant le réseau d'un ISP régional et un AS représentant le réseau d'un opérateur Tier-1. Ceci est très simpliste car au sein de ces deux ASs, la diversité de chemins au niveau PoP peut être très différente.

Afin d'évaluer la diversité des chemins au niveau PoP entre les nœuds de différents CSPs, nous avons besoin d'un graphe Internet dirigé représentant le lien entre les PoPs des opérateurs. Attendu que nous n'avons pas pu trouver un tel graphe actualisé dans notre état de l'art, nous avons décidé de construire un tel graphe. Étant donné l'importance des points d'échange Internet (IXP) qui accueuillent une part croissante du trafic inter-AS total, nous devons les prendre en compte. Dans le graphe que nous construisons, nous déterminons d'abord l'ensemble des participants à chaque IXP pour ensuite localiser ces IXPs au sein du graphe. Nous opérons un regroupement des routeurs des différents AS afin de reconstituer leurs différents PoPs et reconstruire les liens qui les relient entre eux et aux PoPs d'autres ASs.

5.2 Construction d'une topologie au niveau PoP

Pour construire un graphe topologique représentant Internet au niveau des PoPs des différents AS, nous utilisons quatre sources de données :

- Le graphe Internet iPlane, dont la granularité se place au niveau des routeurs [MIP+06],
- Un graphe Internet dirigé au niveau AS récupéré auprès du projet DRAGON [SVLR14],
- Les données portant sur la participation des ASs aux différents IXPs fournies par PeeringDB [Pee],

- Les données de géolocalisation d'adresses IP fournies par la base de données GeoIP2 de MaxMind [Max].

Le processus de construction de notre graphe de niveau PoP se décompose en quatre étapes. Tout d'abord, nous nettoyons les données que nous avons extraites d'iPlane pour réparer les associations malencontreuses entre adresses IP et AS (*cf.* section 5.2.1). Ensuite, en utilisant les données de participation des AS aux différents IXPs, nous déterminons quel routeurs de la topologie iPlane est présent à un IXP donné (*cf.* section 5.2.2). Dans une troisième étape, nous regroupons les routeurs ensemble pour reconstituer les PoPs de chaque AS (*cf.* section 5.2.3). Enfin, nous associons les résultats obtenus lors des trois étapes précédentes pour construire le graphe Internet dirigé au niveau des PoP que nous utiliserons dans notre évaluation de la diversité des chemins(*cf.* section 5.2.4).

5.2.1 Nettoyage des sources de données

Le projet iPlane [MIP+06] produit des données obtenues à partir de mesures traceroute réalisées quotidiennement entre plusieurs points d'observation. Ces relevés de traceroutes sont filtrés et traités afin de produire une synthèse des relations entre différents routeurs sur Internet. Les routeurs identifiés dans les données du projet iPlane ont plusieurs interfaces dont les adresses apparaissent dans les relevés. La synthèse quotidienne faite par iPlane donne la latence moyenne sur des liens reliant les routeurs entre eux, ainsi que le taux de perte de paquets entre ces routeurs. De plus, tous les deux mois, le projet iPlane produit un fichier d'association IP-AS pour permettre d'associer chaque routeur à l'AS auquel il appartient. À partir de ces données, notre objectif initial a été de construire un graphe non dirigé représentant Internet au niveau des routeurs. Afin d'obtenir une couverture suffisante d'Internet tout en conservant des données relativement fraîches, nous avons construit ce graphe à partir des synthèses de mesures traceroute effectuées entre le 13 juin et le 14 juillet 2015.

Au cours de notre construction de ce graphe, nous avons remarqué des incohérences dans les données produites par iPlane. Par exemple, nous avons remarqué que les routeurs appartenant à AS 3303 représentaient 8% de l'ensemble des routeurs apparaissant dans les données d'iPlane, ce qui ne semblait pas cohérent avec l'importance de l'opérateur utilisant ce numéro d'AS. En outre, les associations de certaines plages d'adresses IP à des AS se sont révélées ambigües. Afin de pouvoir reconstituer une association correcte entre routeur et AS, nous avons pré-traité les données iPlane en effectuant certaines corrections à l'aide d'informations fournies par les outils BGP proposés par Hurricane Electric [HE-].

Après cette première étape, nous avons obtenu un graphe d'Internet au niveau routeur comptant **417,638 sommets** représentant les différents routeurs et **7,687,300**

arcs représentant les liens apparaissant dans les mesures iPlane entre ces routeurs. En outre, nous avons également obtenu une association appropriée entre chaque routeur et l'AS auquel il appartient. La figure 5.1 représente de manière schématique le graphe obtenu à ce stade pour un petit nombre d'ASs fictifs.

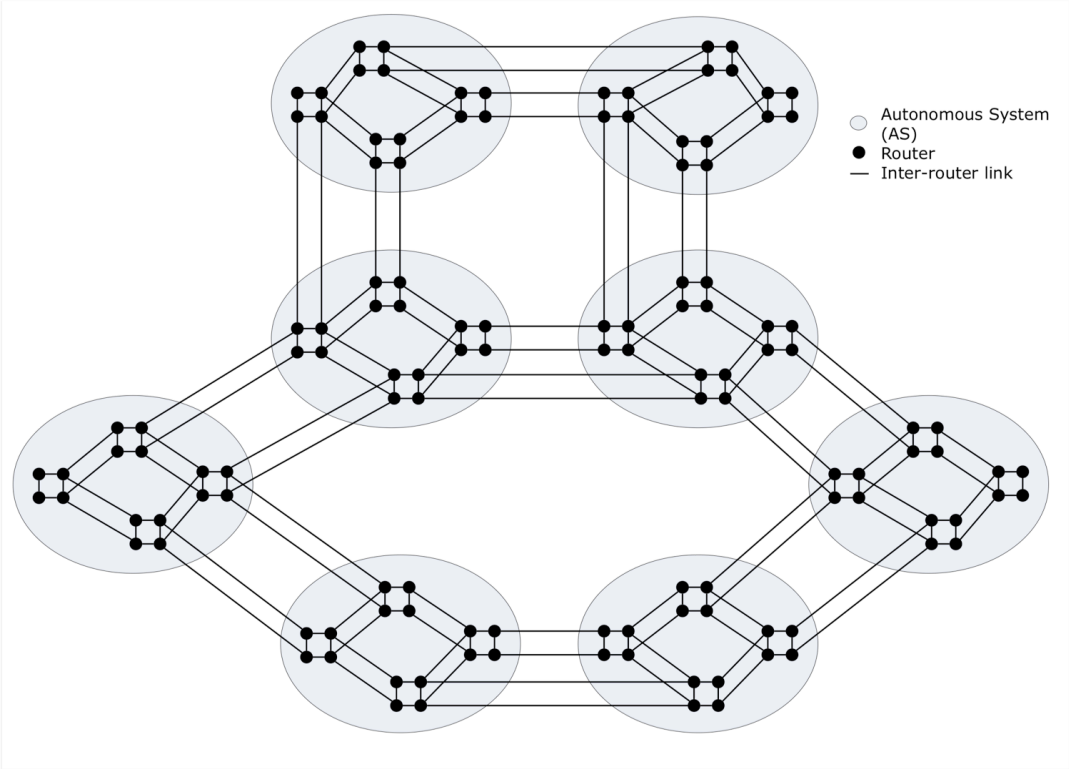


FIGURE 5.1 – Représentation schématique de la topologie obtenue après le nettoyage des données sources.

5.2.2 Reconstitution de l'attachement des routeurs aux différents points d'échange Internet

Dans [ACF+12], Ager *et al.* soulignent que le rôle des IXPs dans l'Internet aujourd'hui n'est pas correctement pris en compte. Ils ont souligné que le nombre de liens de peering établis à ces IXP est beaucoup plus important que ce que les différents serveurs de route BGP publics peuvent suggérer. Nous aimerions prendre en compte l'importance des IXP dans notre topologie Internet au niveau PoP. Pour ce faire, nous devons localiser les IXPs dans le graphe Internet au niveau routeur que nous avons construit lors de la première étape.

À cette fin, nous utilisons d'abord les informations fournies par la base PeeringDB [Pee]. Pour la plupart des IXP, cette base de données fournit des informations portant sur les plages d'adresses IP utilisées par les IXPs pour donner une adresse aux équipements connectés. Ainsi, dans un premier temps, nous avons

analysé l'ensemble des données iPlane pour déterminer quels routeurs utilisent une adresse IP appartenant à la plage d'adresses utilisée par un IXP. Ensuite, en comparant la liste des routeurs présents à chaque IXP et la liste des ASs membres de ces mêmes IXPs, nous avons observé que nous n'avions qu'une association partielle des routeurs aux IXPs. En effet, nous n'avons pu associer un routeur à un IXP que pour seulement 3,5% des AS membres d'un IXP tel que déclaré dans la base de données PeeringDB.

Cette observation nous a mené à appliquer deux autres méthodes pour associer des routeurs aux IXPs. Tout d'abord, pour chaque IXP, nous avons examiné la liste des ASs membres pour déterminer ceux pour lesquels nous ne pouvions pas trouver un routeur participant. Nous avons alors considéré tous les routeurs de ces ASs pour trouver ceux qui sont connectés à un routeur participant à l'IXP. Cette procédure nous a permis d'associer plus de routeurs à chaque IXP : à la fin de cette opération, nous avons pu trouver un routeur présent à un IXP pour 45,4% des AS membres d'un IXP tel que déclaré dans la base de données PeeringDB.

À ce stade, pour plusieurs IXPs, il reste des ASs membres de certains IXPs pour lesquels nous n'avons pu trouver de routeur associé. Pour résoudre ce problème, nous utilisons une dernière technique pour associer les routeurs aux différents IXPs. Dans cette dernière étape, nous utilisons les informations de géolocalisation fournies par la base de données GeoIP2 de Maxmind [Max]. En utilisant les informations portant sur la plage d'adresses IP utilisée par l'IXP déclarée dans la base de données PeeringDB, nous déterminons l'emplacement géographique de chaque IXP. Pour les ASs pour lesquels nous n'avons pu associer un routeur à un IXP auquel ils participent, nous avons cherché à déterminer quel routeur était le plus proche de l'IXP. Les routeurs situés à plus de 30 kilomètres d'un IXP ne sont pas considérés comme participants à cet IXP.

A la fin de ce processus en trois temps, nous obtenons une liste de routeurs connectés à chaque IXP. Dans cette liste, nous sommes en mesure de reconstituer 62,1% des associations entre les AS et les IXPs auxquels ils participent telles que déclarées dans la base PeeringDB. Le caractère partiel de cette information s'explique par le fait que nous avons essayé de reconstruire les associations routeurs - IXP pour tous les IXPs listés dans la base PeeringDB, quel que soient leur importance ou leur localisation. Une étude similaire faite par Lodhi *et al.* [LLD+14] décrit des résultats comparables. Cette étude démontre que les informations portant sur l'adhésion des ASs aux différents IXPs sont moins précises pour les petits IXPs ou pour les IXPs situés dans les régions en développement. La figure 5.2 représente de manière schématique le graphe obtenu après le travail d'association des routeurs avec les IXPs que nous avons effectué.

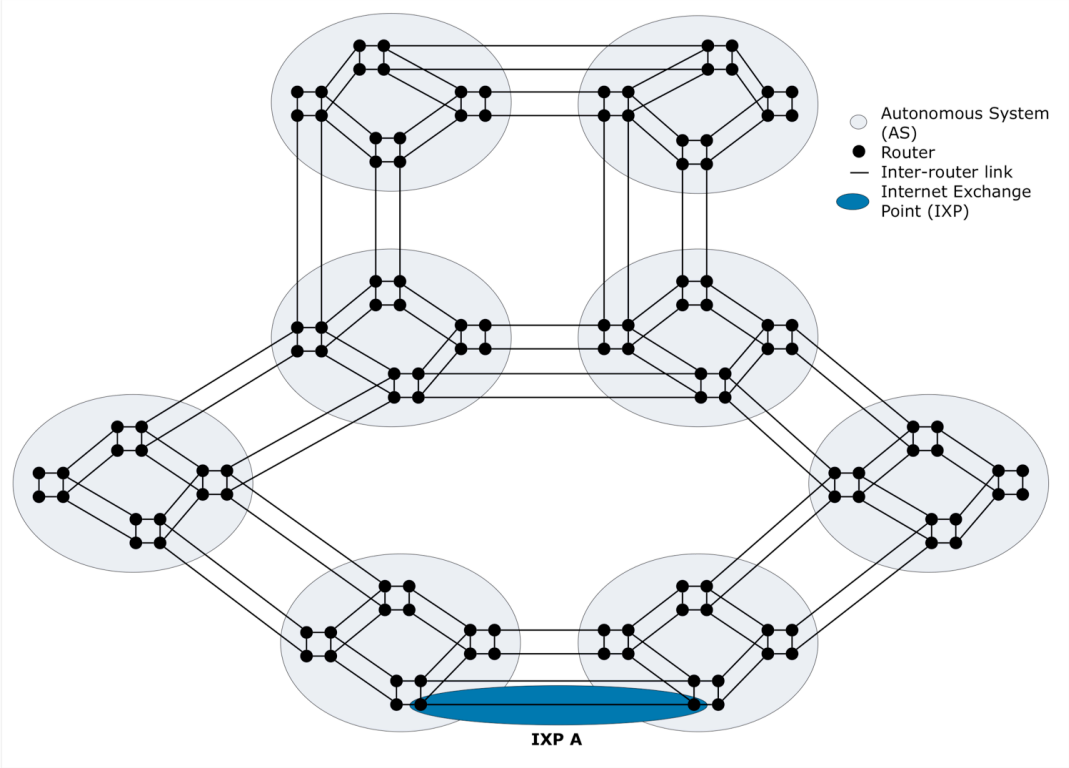


FIGURE 5.2 – Représentation schématique de la topologie obtenue après la reconstitution de l'attachement des routeurs aux différents points d'échange Internet.

5.2.3 Reconstitution des PoPs par regroupement de routeurs

A ce stade, nous avons un graphe représentant Internet au niveau des routeurs, avec une liste de routeurs participants aux différents IXPs. Cette représentation montre le détail de tous les liens inter-routeurs révélés par les données du projet iPlane. Nous avons observé pour certains systèmes autonomes (Sprint, Amazon, IIJ ou Géant) que le nombre de routeurs dans notre topologie est beaucoup plus important que le nombre de points de présence géographiques de ces ASs. En effet, les ASs déploient souvent plusieurs routeurs à chacun de leur PoP pour assurer la résilience de ces PoPs en cas de panne. À plus grande échelle, une panne de routeur au sein d'un PoP n'a quasiment aucun impact sur la résilience globale d'un AS, sauf si cette panne est corrélée avec d'autres pannes affectant le même PoP. Aussi, considérer la topologie Internet au niveau des PoPs est suffisant pour caractériser la diversité de chemin intra-AS sur de grandes échelles. La complexité de l'algorithme que nous allons utiliser pour analyser la diversité de chemins sur Internet croît avec le nombre d'arcs et de sommets dans le graphe. Travailler au niveau des PoPs permet de réduire le temps d'exécution de l'algorithme de découverte de chemins divers sans impact significatif sur notre évaluation de la résilience.

Pour reconstituer les PoPs des différents opérateurs, nous regroupons les routeurs

en fonction de leur proximité relative. Plusieurs techniques existent dans la théorie des graphes pour effectuer un tel regroupement des sommets d'un graphe. Afin de choisir la méthode la plus appropriée, nous comparons les algorithmes de clustering développés et utilisables par l'intermédiaire de la bibliothèque Python igraph [igr]. Nous utilisons ces algorithmes pour regrouper les routeurs appartenant à IIJ, Amazon et Géant dans notre topologie, et nous comparons le nombre de regroupements obtenus avec le nombre de points de présence géographiques de ces systèmes autonomes. La comparaison que nous réalisons montre que deux algorithmes donnent des résultats proches de ce qui est observé en réalité : l'algorithme Infomap [RB07] et l'algorithme de découverte de communautés par marche aléatoire [PL05]. Les bons résultats obtenus en utilisant ces deux algorithmes se justifient par leurs principes de conception. L'algorithme Infomap regroupe les sommets de manière à optimiser l'équation de *mapping* du graphe. Cette équation est une construction mathématique destinée à rendre compte de la structure sous-jacente du graphe. Dans notre graphe, cette structure sous-jacente est sous-tendue par l'architecture réseau des points de présence des opérateurs. L'algorithme de découverte de communautés par marche aléatoire est basé sur le fait qu'un parcours aléatoire du graphe a tendance à passer plus de temps au sein des communautés les plus denses. Dans notre graphe, ces communautés sont constituées par les routeurs qui sont conjointement présents à chaque PoP. Étant donné la capacité des deux algorithmes à effectuer un regroupement de bonne qualité des routeurs d'IIJ, d'Amazon et de Géant, nous avons décidé de les utiliser pour regrouper les routeurs de chaque AS de notre topologie. Dans le cas d'un résultat divergent, nous choisissons de conserver la représentation la plus détaillée pour chaque AS, donc de garder un nombre de PoPs plus important.

Après cette étape, nous obtenons un graphe représentant Internet au niveau des PoPs des différents ASs. Ce graphe est constitué de **148,926 sommets** représentant les PoPs et de **1,041,271 arcs** représentant les liens entre ces PoPs. La figure 5.3 représente de manière schématique le graphe obtenu après le regroupement des routeurs que nous avons réalisé pour chaque AS.

5.2.4 Reconstitution des politiques de routage BGP

Dans la dernière étape de notre processus de construction de graphe, nous devons déterminer quels sont les types de relation entre PoPs qui sont représentées par les différents arcs de notre graphe afin de prendre en compte l'asymétrie des relations entre ASs sur Internet. La topologie réalisée par le projet DRAGON [SVLR14] nous donne les informations nécessaires concernant les arcs reliant les PoPs qui se trouvent dans des ASs différents. Dans DRAGON, un arc entre deux ASs peut être marqué comme représentant une relation *client-fournisseur* ($C2P$), *fournisseur-client* ($P2C$),

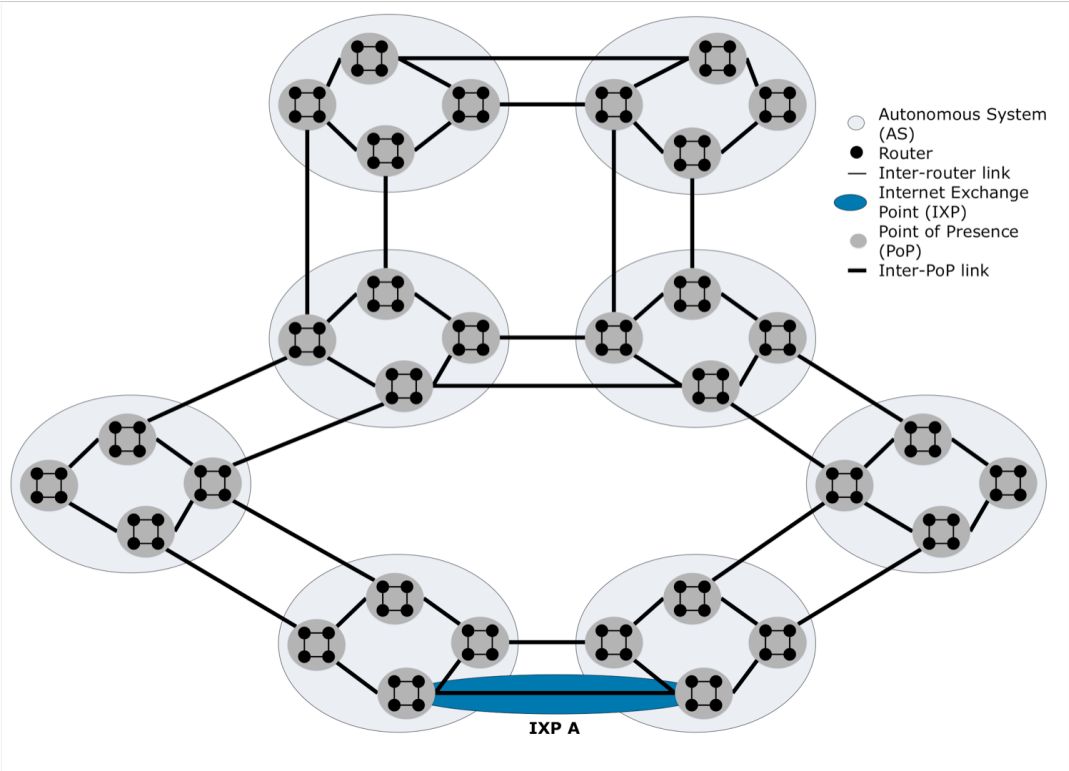


FIGURE 5.3 – Représentation schématique de la topologie obtenue après la reconstitution des PoPs par regroupement de routeurs.

pair à pair ($P2P$) ou *inconnu* (UN). Nous appliquons le même marquage dans notre graphe aux liens entre PoPs appartenant à des ASs différents. A ce stade, les arcs reliant les sommets situés au sein d'un même AS restent sans marquage. Nous choisissons de les marquer comme étant *internes* (IN). Ces arcs seront traités différemment (*cf.* section **??**). La figure 5.4 représente de manière schématique le graphe obtenu après le marquage des liens entre PoPs permettant de rendre compte de la politique de routage BGP utilisée au sein du graphe. Ce graphe sera utilisé dans l'évaluation de la diversité des chemins présentée dans le chapitre 6.

5.3 Conclusion

À la fin du processus décrit dans ce chapitre, nous obtenons une représentation Internet sous forme d'un graphe de PoPs. Dans ce graphe, les arcs reliant les PoPs sont marqués selon le type de relation entre AS qu'ils représentent. Les IXP sont représentés par des sommets logiques connectés aux PoPs où se trouvent les routeurs membres de ces IXPs. Compte tenu que les données à partir desquelles ce graphique a été construit sont récentes, il s'agit d'une représentation à jour d'Internet aujourd'hui.

Le graphe résultant obtenu est constitué de 148 926 sommets et de 1 041 271 arcs

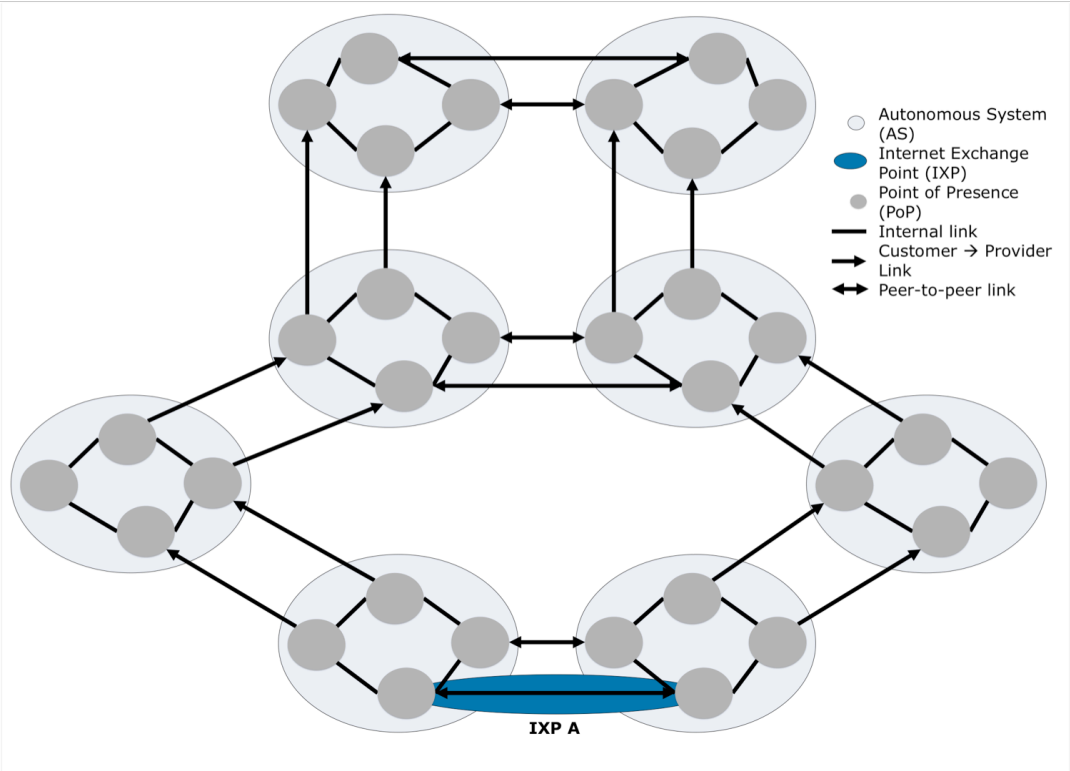


FIGURE 5.4 – Représentation schématique de la topologie obtenue après la reconstitution des politiques de routage BGP.

étiquetés. Ce graphe est assez lâche, comme l'indique sa densité qui est égale à $4,69.10^{-5}$. Le degré moyen dans ce graphe est égal à 13,98 et son diamètre est égal à 12.

Travailler avec une représentation d'Internet au niveau des PoP pour mesurer la diversité de chemin est une opération complexe compte tenu de la grande taille du graphe. Pourtant, la manipulation de ce graphe est plus aisée que de manipuler un graphe Internet de niveau routeur tel que la topologie iPlane. D'autre part, l'évaluation de la diversité des chemins à ce niveau de granularité est suffisante pour rendre compte de la diversité de chemins au sein des systèmes autonomes sur Internet. En effet, les représentations d'Internet au niveau AS tendent à masquer les différences d'échelle et d'étendue géographique entre ASs.

Chapitre 6

Évaluation de la diversité de chemin

Contents

6.1 Présentation de la méthode d'évaluation

Dans ce chapitre, nous présentons notre évaluation des bénéfices en termes de résilience de l'architecture Kumori. Avec cette architecture, nous voulons améliorer la

résilience des connexions entre centres de données en augmentant le nombre de chemins routables disjoints que nous pouvons emprunter pour permettre de rerouter le trafic efficacement en cas de panne d'un nœud ou d'un lien sur le chemin emprunté par défaut. Nous cherchons donc à comparer le nombre de chemins routables disjoints que l'architecture Kumori permet d'utiliser au nombre de chemins disjoints directs empruntables sur Internet entre différents datacenters d'un CSP.

Dans cette évaluation, nous utilisons la représentation Internet de niveau PoP présentée dans le chapitre 5. Nous avons conçu et développé un algorithme pour trouver les chemins disjoints respectant la politique de routage de Gao-Rexford entre deux PoPs dans ce graphe. Avec cet algorithme, nous recherchons l'ensemble des chemins disjoints entre les paires de PoPs de deux CSPs dans quatre situations différentes :

- **Chemins à arcs disjoints directs** : Ces chemins entre deux PoPs ne partagent aucun arc, mais ils peuvent partager un ou plusieurs nœuds. Nous les désignons comme étant directs car l'architecture Kumori n'est pas utilisée pour utiliser ces chemins. Le nombre de chemins à arcs disjoints directs entre deux PoP est le nombre de chemins disponibles entre ces deux PoP sur Internet.
- **Chemins à nœuds disjoints directs** : Ces chemins entre deux PoPs ne partagent aucun nœud. Ainsi, par la construction, ils ne partagent aucun arc non plus. Nous les décrivons comme direct car nous n'utilisons pas Kumori pour accéder à ces chemins.

Le nombre de chemins à arcs disjoints ou à nœuds disjoints directs entre deux PoP est le nombre de chemins qui peuvent être emprunté sur Internet entre ces deux PoPs.

- **Chemins Kumori à arcs disjoints** : Ces chemins sont des chemins à arcs disjoints entre deux PoPs qui traversent un nœud Kumori situé à un IXP donné.
- **Chemins Kumori à nœuds disjoints** : Ces chemins sont des chemins à nœuds disjoints entre deux PoPs qui traversent un nœud Kumori situé à un IXP donné.

Le nombre de chemins Kumori disjoints empruntables entre deux PoPs est comparé au nombre de chemins disjoints directs entre ces mêmes nœuds pour évaluer les avantages associés à l'utilisation de Kumori en termes de résilience.

Nous recherchons le nombre de chemins disjoints parmi les paires de PoPs de deux fournisseurs de services Cloud, Amazon et Atos. Nous évaluons les avantages de Kumori en termes de résilience en observant l'augmentation du nombre de chemins disjoints disponibles pour les paires PoP de ces deux CSP. Ensuite, nous évaluons l'impact de certains facteurs de conception et de déploiement de l'architecture Kumori tels que le nombre de nœuds dans l'*overlay* ou le choix des IXPs où les nœuds sont situés sur les performances en termes de résilience.

6.2 Description de l'algorithme de recherche de chemins disjoints

Tout d'abord, nous devons déterminer combien de chemins disjoints peuvent être empruntés entre tous les PoPs des CSPs que nous considérons dans notre étude ; soit de manière directe, soit via un IXP pour évaluer les bénéfices apportés par Kumori. Nous utilisons le graphe représentant Internet que nous avons créé et présenté dans le chapitre 5 pour cette évaluation.

6.2.1 Un premier algorithme de recherche de chemins *valley-free*

Comme nous l'avons mentionné dans la section 2.3.3, le problème de la découverte de chemins *valley-free* dans notre graphe orientée de niveau PoP est un problème complexe. Au niveau AS, ce problème est NP-dur. Dans des travaux antérieurs présentés dans [EHM+06] ou dans [KKAD15], des méthodes de transformation de graphes ont été suggérées pour transformer un graphe orienté de niveau AS et faciliter ainsi la découverte de chemins *valley-free* dans ce graphe. Ces transformations utilisent le fait que, dans un tel graphe, la plupart des relations sont asymétriques et que les chemins routables sur Internet ne peuvent traverser qu'une seule relation inter-AS de type $p2p$. Dans notre graphe Internet de niveau PoP, nous considérons que toutes les relations intra-AS sont symétriques et nous ne limitons pas le nombre d'arcs intra-AS qu'un chemin routable peut emprunter. A ce titre, nous ne pouvons pas transposer facilement les méthodes de transformation proposées dans [EHM+06] et dans [KKAD15] sur notre graphe.

L'impossibilité d'appliquer des simplifications à notre graphe orienté de niveau PoP représentant Internet nous a amené à adopter une autre méthode pour découvrir les chemins routables au sein de ce graphe. Dans une première approche, nous avons essayé de rechercher les chemin *valley-free* dans notre graphe sans appliquer ni transformation ni simplification. Nous avons utilisé un algorithme de recherche en profondeur en raison de son empreinte mémoire plus faible que celle d'un algorithme de recherche en largeur. Dans cette recherche de chemin, nous avons appliqué la contrainte liée à l'application de la politique de routage *valley-free* en gardant en mémoire le type du dernier lien inter-AS parcouru. Si le dernier arc inter-AS a été marqué comme étant de type *client-fournisseur*, tous les arcs inter-AS peuvent être parcouru dans la suite de notre recherche de chemin. Par contre, si le dernier arc inter-AS parcouru était marqué comme étant de type *pair à pair* ou *fournisseur-client*, seuls les arcs inter-AS de type *fournisseur-client* peuvent être parcourus. Dans tous les cas, les arcs de type *interne* peuvent toujours être parcourus.

Nous avons choisi d'ignorer les arcs de type *unknown* dans notre recherche de chemin car nous ne pouvions pas appliquer de manière claire la politique de routage *valley-free* en parcourant ces liens. Les détails de cet algorithme initial sont présentés dans l'encart 3.

Algorithm 3 Algorithme *Depth-first search* initial utilisé pour la recherche de chemins *valley-free*

Require : $start \leftarrow$ nœud de départ,
$destination \leftarrow$ Destination à atteindre,
$graph \leftarrow$ Graphe représentant Internet,
$node \leftarrow$ nœud courant,
$in \leftarrow$ Variable d'état décrivant le type du dernier lien inter-AS franchi,
$explored \leftarrow$ Liste des nœud franchis,
$queue \leftarrow$ File d'attente contenant les tuples $(node, in, explored)$.
Ensure : $node = destination$
$incoming \leftarrow c2p$
$node \leftarrow start$
$queue \leftarrow INSERT((start, c2p, explored), queue)$
while $queue$ is not $empty$ **do**
$(node, incoming, explored) \leftarrow POP(queue)$
if $node = destination$ **then**
return $explored$
end if
$children \leftarrow CHILDREN(node, graph)$
for $child$ in $children$ **do**
$edge \leftarrow EDGE(node, child, graph)$
$next_{in} \leftarrow TYPE(edge)$
if $child$ in $explored$ **then**
$continue$
end if
if $next_{in}$ is $internal$ **then**
$explored \leftarrow explored + node$
$queue \leftarrow (child, in, explored)$
end if
if in is $c2p$ or $next_{in}$ is $p2c$ **then**
$explored \leftarrow explored + node$
$queue \leftarrow (child, next_{in}, explored)$
end if
end for
end while

Nous avons implémenté ce premier algorithme de recherche de chemin en Python. Cela nous a permis de jeter les bases de cet algorithme de découverte de chemins *valley-free*, mais nous avons surtout constaté qu'étant donné la complexité de notre graphe de niveau PoP, cet algorithme ne permettrait pas d'obtenir des résultats en un temps raisonnable : Ainsi, la découverte de l'ensemble des chemins à arcs disjoints entre deux des PoPs de notre graphe a pris plusieurs semaines sur une machine

récente dotée de 64 Go de mémoire vive et de 8 cœurs de calcul.

6.2.2 Optimisation de l'algorithme

Après notre première implémentation, nous avons cherché par quelles méthodes nous pourrions accélérer notre recherche de chemin. Attendu que la complexité de notre algorithme de recherche de chemin croît avec le nombre d'arcs et de sommets dans le graphe, nous avons d'abord cherché à supprimer des sommets afin de réduire la taille du graphe de niveau PoP. Nous avons donc utilisé le graphe représentant Internet au niveau des ASs et nous avons recherché dans ce graphe les chemins *valley-free* entre les PoPs des CSPs, ainsi qu'entre ces PoPs et les différents IXPs.

Nous avons en outre cherché à améliorer notre algorithme. Considérant que les chemins disjoints *valley-free* constituent un sous-ensemble de l'ensemble des chemins disjoints dans un graphe orienté, nous avons cherché des solutions possibles à ce problème dans la littérature. Nous avons constaté qu'il est possible de démontrer que limiter la longueur maximale du chemin réduit considérablement la complexité de l'algorithme de découverte de chemins disjoints. Poser une telle limite est logique car, comme l'expliquent Kühne *et al.* [KA12], le nombre moyen d'AS sur un chemin entre deux points d'Internet est relativement stable, et on peut considérer que le trafic Internet entre deux nœuds traverse en moyenne 4,3 ASs, en dépit de l'augmentation observée du nombre d'ASs sur Internet. En effet, il a été montré par Dhamdhere *et al.* [DD10] qu'Internet est un réseau relativement aplati. Attendu ces constatations, nous avons limité le nombre d'ASs traversés par les chemins que nous cherchions à 4. Nous avons ensuite exécuté notre algorithme en limitant le nombre de PoPs composant les chemins et en augmentant progressivement cette limite. Nous présentons l'impact de cette limite sur le nombre de chemins divers trouvé dans les sections 6.4.2 et **??**.

Enfin, nous avons apporté une dernière amélioration à notre algorithme en calculant la longueur du chemin le plus court entre chaque sommet dans notre graphe de niveau PoP. Pour simplifier ce calcul, nous avons supprimé les contraintes associées à la politique de routage *valley-free*. Nous obtenons la longueur de chemin la plus courte pour chaque paire PoP. Nous utilisons ensuite cette information portant sur la longueur du chemin le plus court à deux fins. Tout d'abord, cette longueur de chemin le plus court vers un PoP de destination peut nous aider à abandonner l'exploration de certains chemins. Dans notre algorithme de recherche de chemin, à chaque sommet, on soustrait la longueur du chemin le plus court vers la destination à la limite de longueur du chemin. Si le résultat est négatif, on ne peut atteindre la destination en utilisant un chemin suffisamment court. Alors, nous abandonnons l'exploration de ce chemin. En outre, nous avons utilisé les mesures de longueur du chemin le plus court à chaque étape pour classer les successeurs possibles d'un nœud donné avant de les mettre

dans la file d'attente des nœuds à explorer. Cette modification a été inspirée par l'algorithme A* [HNR68]. Nous utilisons la distance parcourue calculée la plus courte comme une métrique indiquant le candidat à visiter en priorité pour atteindre la destination le plus rapidement possible. La version finale de notre algorithme est présentée dans l'encart 4.

6.2.3 Exécution de l'algorithme

Nous avons implémenté notre algorithme optimisé en C++ en utilisant Cython [BBC+11], un compilateur convertissant des programmes Python en C ou en C++. Nous avons choisi d'effectuer cette compilation en C++ pour des raisons de performances et d'empreinte mémoire. Nous avons alors exécuté notre algorithme de recherche de chemin disjoint sur un cluster de calcul composé de 9 serveurs de calcul équipés chacune de 2 processeurs Intel Xeon cadencés à 2,6 GHz. L'exécution de la recherche de chemins disjoints a été répartie entre différents nœuds du cluster en utilisant SLURM [YJG03], un gestionnaire de tâches utilisé dans de nombreux clusters de calcul HPC (High Performance Computing) utilisant Linux ou Unix.

6.3 Méthode d'évaluation

Nous présentons comment nous utilisons les chemins trouvés avec l'algorithme décrit dans les sections précédentes pour évaluer la diversité de chemins entre les PoPs d'un CSP donné et comparer plusieurs méthodes permettant d'accéder à des chemins disjoints. Dans cette évaluation de la diversité, nous utilisons deux métriques élaborées dans des travaux antérieurs. En premier lieu, nous utilisons la distribution cumulative du nombre de chemins divers trouvés entre plusieurs paires de nœuds, qui a été présentée par Texeira *et al.* [TMSV03b]. Dans notre travail, nous avons l'intention d'appliquer une méthode similaire au graphe Internet de niveau PoP que nous avons présenté dans le chapitre 5 et ainsi aller au-delà d'une qualification de la diversité réduite à l'échelle du réseau d'ASs. En outre, nous utilisons le score de diversité introduit par Rohrer *et al.* [RS11]. Ce score de diversité est une mesure comprise entre 0 et 1 qui peut être utilisée pour caractériser la disjonction de deux chemins, d'un ensemble de chemins entre deux nœuds ou des chemins au sein d'un graphe entier. Nous utiliserons cette mesure dans notre travail pour fournir un score synthétique évaluant la diversité des chemins entre les PoPs d'un CSP donné.

Algorithm 4 Depth-first valley-free path search with path length limit

Require : $start \leftarrow$ nœud de départ,
$destination \leftarrow$ Destination à atteindre,
$graph \leftarrow$ Graphe représentant Internet,
$node \leftarrow$ nœud courant,
$in \leftarrow$ Variable d'état décrivant le type du dernier lien inter-AS franchi,
$explored \leftarrow$ Liste des nœud franchis,
$metrics \leftarrow$ Dictionnaire associant à chaque nœud la longueur du chemin le plus court vers la destination,
$queue \leftarrow$ File d'attente contenant les tuples $(node, in, explored)$.
Ensure : $node = destination$
$incoming \leftarrow c2p$
$node \leftarrow start$
$queue \leftarrow INSERT((start, c2p, explored), queue)$
$list_{temp} \leftarrow$ possible candidates list
while $queue$ is not $empty$ **do**
  $(node, incoming, explored) \leftarrow POP(queue)$
  if $node = destination$ **then**
    return $explored$
  end if
  if $length(explored) > limit - metrics[node]$ **then**
    return $failure$
  end if
  $children \leftarrow CHILDREN(node, graph)$
  for $child$ in $children$ **do**
    $edge \leftarrow EDGE(node, child, graph)$
    $next_{in} \leftarrow TYPE(edge)$
    if $child$ in $explored$ **then**
      $continue$
    end if
    if $next_{in}$ is $internal$ **then**
      $explored \leftarrow explored + node$
      $list_{temp} \leftarrow (child, in, explored)$
    end if
    if in is $c2p$ or $next_{in}$ is $p2c$ **then**
      $explored \leftarrow explored + node$
      $list_{temp} \leftarrow (child, next_{in}, explored)$
    end if
  end for
  $list_{temp} \leftarrow SORT(list_{temp}, metrics)$
  $queue \leftarrow list_{temp}$
end while

6.3.1 Distribution cumulative du nombre de chemins divers

ă La première métrique que nous utilisons est la distribution cumulative du nombre de chemins divers introduite par Texeira *et al.* dans [TMSV03a] et dans [TMSV03b]. Dans

notre évaluation, nous l'adaptons afin de comparer la distribution cumulative du nombre de chemins à arcs disjoints et la distribution cumulative du nombre de chemins à nœuds disjoints à la distribution cumulative du nombre de chemins maximal. La distribution cumulative du nombre de chemins maximal est calculée en déterminant, pour chaque paire de PoPs que nous considérons, le minimum entre le degré sortant du sommet source et le degré entrant du sommet de destination. Le nombre résultant est le nombre maximum de chemins disjoints qui peuvent être trouvés entre les deux sommets représentant la paire de PoPs. Cette diversité maximale est la limite supérieure du nombre de chemins à arcs disjoints et à sommets disjoints que nous pourrons trouver. Plus nous approchons de la diversité maximale, plus un système donné est capable de donner accès à une part importante des chemins disjoints existants entre deux nœuds donnés.

6.3.2 Score de diversité

La deuxième métrique que nous utilisons pour caractériser la diversité de chemin est le score de diversité présenté par Rohrer *et al.* [RJS14]. Dans ces travaux, les auteurs introduisent un ensemble de mesures pour caractériser la disjonction de chemins. Un chemin P entre une source s et une destination d est défini comme un vecteur contenant tous les arcs L et tous les sommets intermédiaires N composant ce chemin. Cela s'exprime par l'équation :

$$P = L \cup N \tag{6.1}$$

En utilisant cette définition, Rohrer *et al.* définissent le score de diversité de deux chemins donnés P_a et P_b. Ce score de diversité est donné par la formule :

$$D(P_a, P_b) = 1 - \frac{|P_a \cap P_b|}{|P_a|} \tag{6.2}$$

Où $|P_a| \leq |P_b|$. Ce score de diversité a une valeur comprise entre 0, qui signifie que les chemins sont identiques, et 1, qui indique que les chemins sont complètement disjoints.

À partir de ce score de diversité entre deux chemins donnés, Rohrer *et al.* calculent un score de diversité global pour un ensemble de chemins $\{P_0 ... P_k\}$, le score de diversité de chemins effectif ou EPD. Cet EPD est donné par la formule :

$$EPD = 1 - e^{-\lambda k_{sd}} \tag{6.3}$$

Où k_{sd} est une somme de scores de diversité donnée par la formule :

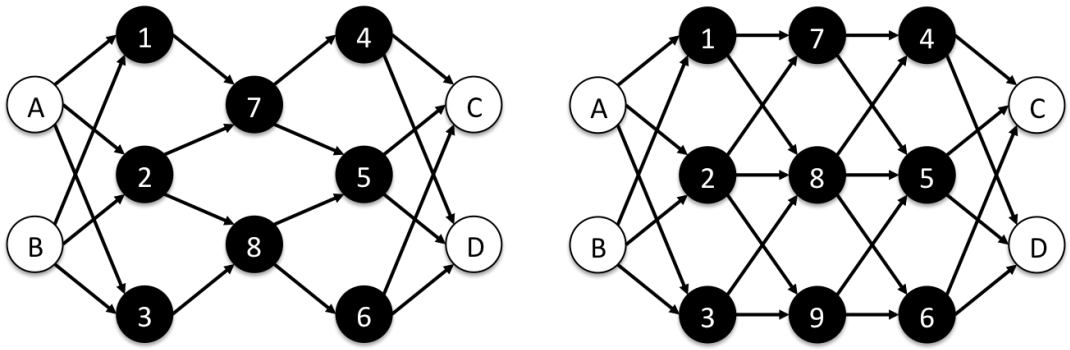
$$k_{sd} = \sum_{i=1}^{k} D_{min}(P_i) \tag{6.4}$$

Où $D_{min}(P_i)$ est, pour un chemin P_i, le score de diversité minimum obtenu en calculant le score de diversité de ce chemin avec tous les autres chemins de l'ensemble $\{P_0...P_k\}$ et où λ est une constante qui exprime la nécessité de fournir des chemins suffisamment disjoints entre eux. Dans notre évaluation, nous choisissons de fixer λ à 1. Le score EPD d'un ensemble de chemins est également compris entre 0 et 1. Le score EPD d'un ensemble contenant moins de 2 chemins est égal à 0.

Le score de diversité effectif d'un graphe est donné par la moyenne des scores EPD des ensembles de chemins disjoints entre chaque $source$ et chaque $destination$ au sein du graphe. Dans notre évaluation de la diversité, l'EPD arc-disjoint est la moyenne des scores EPD de l'ensembles des chemins à arcs disjoints entre les paires de nœuds que nous considérons, alors que l'EPD nœud-disjoint est la moyenne des scores EPD de l'ensembles des chemins à sommets disjoints parmi ces paires.

6.3.3 Calcul des métriques pour deux exemples de graphes

Pour permettre une compréhension intuitive de la capacité des métriques présentées préalablement à rendre compte de la diversité des chemins dans un graphe, nous comparons les deux graphes exemples. Ces deux graphes sont présentés dans la figure 6.1.



(a) Graph 1 (b) Graph 2

FIGURE 6.1 – Deux graphes exemples

Nous observons que le premier graphe présente une diversité de chemin à sommets disjoints plus petite que le deuxième graphe car il y a un goulot d'étranglement au milieu du premier graphe.

Pour le premier graphe, la distribution cumulative du nombre de chemins divers est représentée sur la figure 6.2. Sur cette figure, nous voyons que la la distribution cumulative du nombre de chemins divers à sommets disjoints est inférieure à la distribution cumulative du nombre de chemins divers maximale et à la distribution cumulative du nombre de chemins divers à arcs disjoints. Cela rend compte de l'impossibilité de trouver trois chemins à sommets disjoints entre A ou B et C ou D étant

donné que les nœuds 7 et 8 sont un point de contention. Pour le deuxième graphe, les trois distributions cumulatives sont superposées.

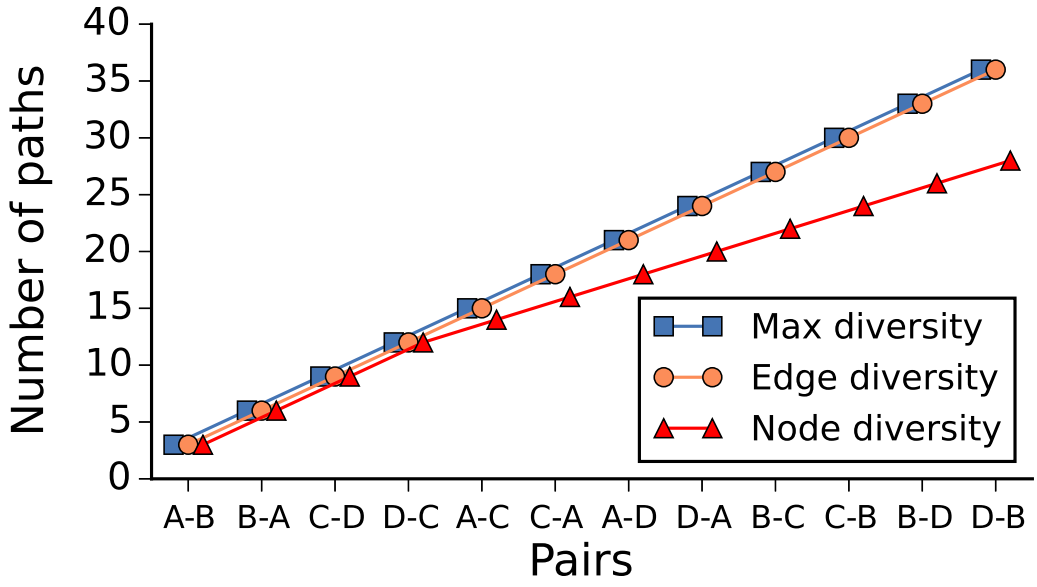


FIGURE 6.2 – Distribution cumulative du nombre de chemins entre A, B, C et D pour le graphe 1.

Examinons maintenant les scores de diversité. Nous considérons le premier graphe pour montrer comment le score de diversité est calculé pour une paire donnée, $A \to C$. Pour cette paire, on peut trouver trois trajets disjoints entre A et C :
$P_1 =< A \to 1 \to 7 \to 4 \to C >$, $P_2 =< A \to 2 \to 7 \to 5 \to C >$ et
$P_3 =< A \to 3 \to 8 \to 6 \to C >$. En calculant la diversité relative des deux premiers chemins, nous trouvons :

$$D(P_1, P_2) = 1 - \frac{|P_1 \cap P_2|}{|P_1|} = 1 - \frac{1}{7} = \frac{6}{7} \tag{6.5}$$

La diversité effective du chemin $EPD_{A,C}$ de la paire $A \to C$ peut être calculée comme suit :

$$\lambda k_{sd} = \sum_{i=1}^{k} D_{min}(P_i) = 2 * \frac{6}{7} + 1 = \frac{19}{7} \tag{6.6}$$

$$EPD_{A,C} = 1 - e^{-\lambda k_{sd}} = 1 - e^{-\frac{19}{7}} = 0.93338 \tag{6.7}$$

Le tableau 6.1 présente les résultats obtenus en réalisant ce calcul pour l'ensemble des paires de nœuds des deux graphes. Ces résultats peuvent nous aider à interpréter les résultats que nous obtiendrions sur un graphe inconnu. En premier lieu, les scores de diversité du graphe 2 sont plus élevés que pour le graphe 1, ce qui traduit le gain de diversité fourni par l'addition du nœud 9 dans le graphe 2. De plus, si l'on observe le

score $EPD_{A,C}$ nœud-disjoint, on remarque que le score a augmenté dans le graphe 2, alors que le score de diversité de deux chemins nœud-disjoint est toujours égal à 1. Cette augmentation est liée au fait que dans le graphe 2, on peut trouver trois chemins à sommets disjoints entre A et C alors qu'on ne peut trouver que 2 tels chemins dans le graphe 1.

TABLE 6.1 – Scores de diversité pour les graphes 1 et 2

Métrique	Graphe 1	Graphe 2
$EPD_{A,C}$ **arc-disjoint**	0.93338	0.95021
$EPD_{A,C}$ **nœud-disjoint**	0.86466	0.95021
EPD **arc-disjoint**	0.93899	0.95021
EPD **nœud-disjoint**	0.89318	0.95021

6.4 Amélioration de la diversité de chemins par l'utilisation de l'architecture Kumori

6.4.1 Diversité de chemins sur Internet pour deux CSPs

Nous utilisons la méthode présentée dans les sections précédentes pour mesurer la diversité des chemins Internet parmi les PoPs appartenant à deux CSPs, Amazon et Atos. Amazon est le plus grand fournisseur de services Cloud en termes de parts de marché [**?**], tandis qu'Atos est l'un des plus importants CSPs européens. Ces deux CSPs utilisent de stratégies différentes pour connecter leurs infrastructures à Internet. Amazon est doté de routeurs connectés à 52 IXPs différents, alors qu'Atos n'est présent qu'à un seul IXP. Dans le cadre de son offre AWS Direct Connect, Amazon cherche être étroitement relié aux opérateurs et aux IXPs les plus populaires afin de permettre à ses clients de se connecter à ses services en utilisant un chemin Internet le plus court possible [AWS].

Sur la figure 6.3, nous observons la distribution des paires de PoPs en fonction de la diversité de chemin maximal (telle que définie dans la section 6.3) pour Atos et pour Amazon. On remarque que la diversité maximale des paires de PoPs d'Atos est moins importante que celle des PoPs d'Amazon. En particulier, pour près de 70% des paires de PoPs d'Atos, un seul chemin peut être emprunté entre les nœuds. Cette différence s'explique par les différentes stratégies de connectivité adoptées par ces deux CSPs. Les efforts d'Amazon pour se rapprocher "topologiquement" des gros opérateurs et des IXPs populaires se traduisent par un nombre plus faible de paires de PoPs dont la diversité de chemins maximale est limitée.

Nous comparons le nombre de chemins disjoints qui peuvent être trouvés sur

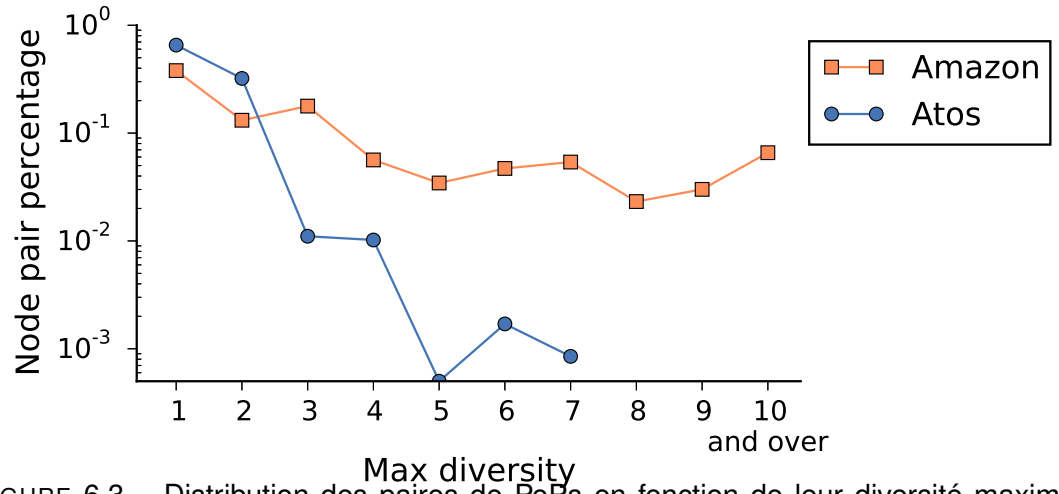


FIGURE 6.3 – Distribution des paires de PoPs en fonction de leur diversité maximale pour Amazon et Atos.

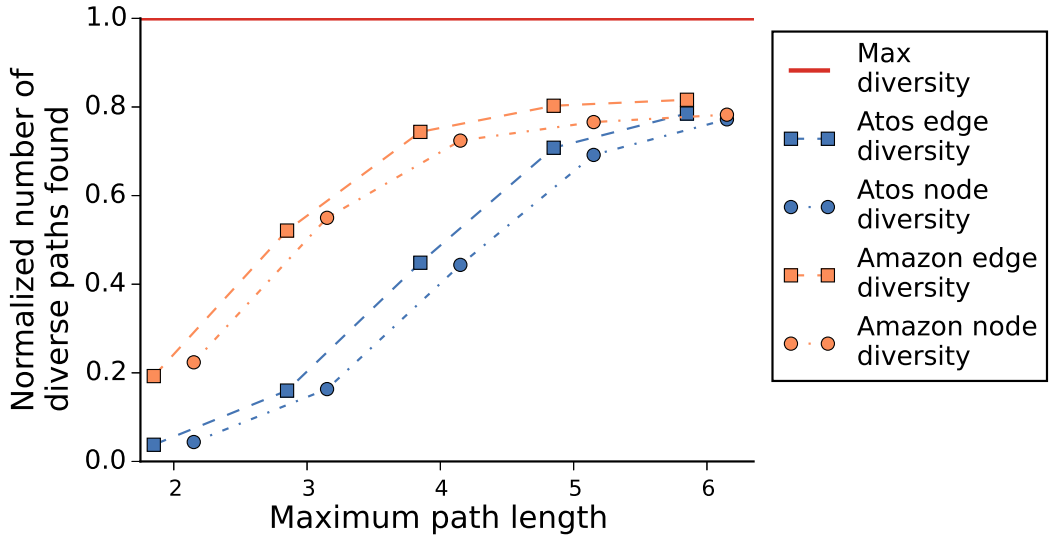


FIGURE 6.4 – Nombre normalisé de chemins divers arc-disjoints et nœud-disjoints trouvés sur Internet en fonction de la longueur maximale du chemin pour Amazon et Atos.

Internet en utilisant notre algorithme entre les PoPs d'Atos et d'Amazon. La figure 6.4 représente les distributions cumulatives du nombre de chemins divers que nous avons pu trouvé pour les deux CSPs. Cette figure représente, pour les deux CSPs de notre étude, le nombre normalisé de chemins à sommets et à arcs disjoints trouvés pour chaque paire de PoPs, en posant une limite croissante sur la taille maximale de ces chemins. La figure montre que le nombre de chemins à sommets ou à arcs disjoints que nous avons pu trouver pour les deux CSPs est moins important que la diversité maximale que nous avons pu calculer. Dans le meilleur des cas, on ne peut utiliser que 80% de la diversité maximale. Cette limitation est liée au fait que tous les chemins topologiques existants entre les PoP ne peuvent pas être empruntés sur Internet du fait

de limitations liées aux politiques de routage des différents opérateurs.

TABLE 6.2 – Scores de diversité obtenus avec une connectivité simple via Internet pour Atos et Amazon

Métrique	Amazon	Atos
EPD **arc-disjoint**	0.50637	0.17020
EPD **nœud-disjoint**	0.50261	0.16431

Nous calculons maintenant les scores de diversité EPD présentés dans la section 6.3.2 pour les chemins à arcs et à sommets disjoints que nous avons trouvés pour les paires PoP d'Atos et d'Amazon. Les résultats que nous avons obtenus sont présentés dans le tableau 6.2 ? On peut constater que ces scores sont relativement faibles. Cela signifie que les chemins trouvés pour les différentes paires PoPs empruntent des arcs ou passent par des sommets qui sont des points de contention du réseau.

6.4.2 Amélioration de la résilience avec Kumori

Notre caractérisation de la diversité de chemins entre les PoPs d'Atos et d'Amazon montre que ces deux CSPs ne bénéficient pas pleinement de leur diversité potentielle. Les scores de diversité calculés à partir des chemins disjoints que nous avons trouvés montrent une diversité assez limitée, en particulier pour Atos. La différence entre la diversité topologique mise en évidence par la diversité maximale et le nombre de chemins routables accessibles sur Internet tend à montrer que les limitations que nous avons observées sont associées aux politiques de routage appliquées entre les différents ASs sur Internet.

Nous tentons de caractériser les avantages potentiels en termes de résilience qu'Atos et Amazon peuvent tirer de l'utilisation de l'*overlay* Kumori. Pour cela, nous utilisons notre méthodologie d'évaluation de la diversité de chemins. Kumori utilise des points d'inflexion de routage présents à différents IXPs et utilise un mécanisme d'encapsulation de trafic réseau entre ces points d'inflexion pour dépasser certaines restrictions imposées par les politiques de routage BGP aux chemins qui peuvent être empruntés entre deux PoPs appartenant à un fournisseur de services Cloud donné. Nous faisons l'hypothèse que l'usage de points d'inflexion placés aux IXPs permettra d'accroître le nombre de chemins disjoints utilisables entre les PoPs afin de s'approcher de la diversité maximale.

Nous utilisons la méthode d'évaluation de la diversité de chemins présentée plus tôt pour déterminer quels sont les avantages associés à l'utilisation de Kumori pour Atos et Amazon. Dans cette évaluation, nous comparons la diversité de chemins pour chaque paire PoP de chacun des deux CSPs sur Internet directement et en utilisant un *overlay*

Kumori doté de 5 points d'inflexion de routage. Notre algorithme de recherche de chemins est paramétré de manière à rechercher les chemins d'une longueur maximale de 7 sauts. Les points d'inflexion de routage que nous utilisons sont situés au sein d'IXPs situés à Francfort, à Sao Paolo, à Hong-Kong, à New-York et à Seattle. Ces points d'inflexion de routage permettent d'avoir une couverture géographique importante.

Distribution cumulative de la diversité de chemins

Avantages associés à l'utilisation de Kumori en termes de diversité de chemin : La figure 6.5 représente le nombre moyen de chemins à arcs et à sommets disjoints sur Internet et au sein de l'*overlay* Kumori qui ont été trouvés au moyen de notre algorithme de recherche pour chaque paire PoP d'Amazon et d'Atos. A travers cette figure, nous voulons déterminer si l'utilisation de Kumori augmente le nombre de chemins disjoints trouvés pour les deux CSP. La figure montre que, pour Atos, l'utilisation de l'architecture Kumori permet une augmentation du nombre de chemins disjoints accessibles. Pour Amazon, on constate qu'il y a un bénéfice en termes de chemins à arcs disjoints tandis que nous observons une diminution du nombre de chemins à sommets disjoints accessibles. Nous expliquons cette dégradation par le fait que nous n'avons utilisé que 5 points d'inflexion de routage pour Kumori dans notre évaluation.

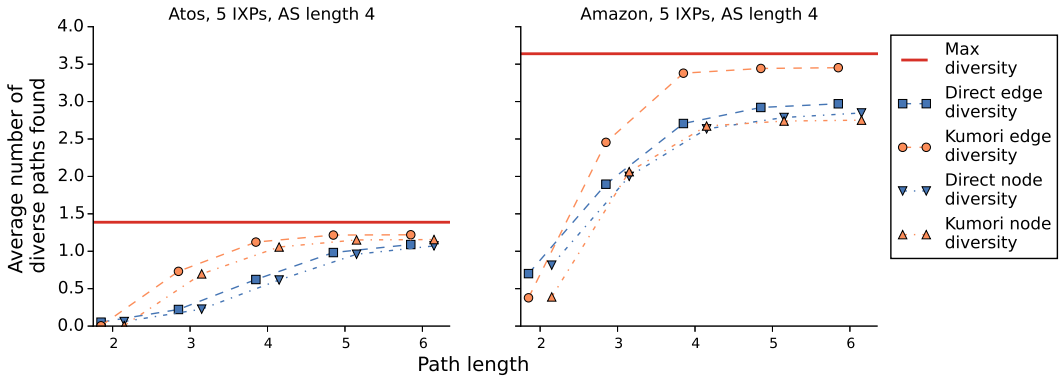


FIGURE 6.5 – Nombre moyen de chemins divers trouvés en fonction de la longueur maximale des chemins pour Amazon et Atos

Impact de la longueur maximale du chemin sur la diversité observée : Sur la figure 6.5, nous étudions maintenant la diversité de chemins trouvée en fonction de la longueur maximale des chemins. Cette comparaison est utile pour déterminer s'il existe une longueur de chemin au-delà de laquelle il n'est pas nécessaire de rechercher des chemins disjoints alternatifs pour augmenter significativement la diversité de chemins proposée. Pour les deux CSP, nous observons qu'avec l'architecture Kumori, au delà de 4 à 5 sauts, on ne peut pas trouver un nombre de chemins diversifiés significativement

plus important.

Bénéfices apportés par Kumori pour différentes catégories de nœuds : Nous considérons maintenant toutes les paires de PoPs de chacun des deux CSPs et nous les classons en fonction de leur diversité de chemin maximale. Ensuite, pour chaque catégorie, nous comparons le nombre de chemins à sommets disjoints et à arcs disjoints trouvés au sein de l'architecture Kumori et directement sur Internet. La figure 6.6 représente le nombre normalisé de chemins trouvés par catégorie de paires PoPs pour Amazon et pour Atos. Sur cette figure, nous observons que, pour les deux CSP, l'architecture Kumori rend un plus grand nombre de chemins accessibles pour des paires de nœuds dont la diversité maximale est faible, alors que le bénéfice est moins évident pour les paires de nœuds dont la diversité maximale est plus élevée. Il est à noter que, pour Amazon, l'architecture Kumori est efficace dans la mise à disposition de chemins à arcs disjoints, tandis que la capacité de l'architecture à fournir des chemins à sommets disjoints tend à diminuer lorsque la diversité maximale augmente. Nous expliquons cet effet par le nombre limité de points d'inflexion de routage utilisés par l'architecture Kumori dans notre évaluation. Ces nœuds d'indirection constituent alors des points de contention du réseau pour des paires de PoPs disposant d'une diversité topologique importante.

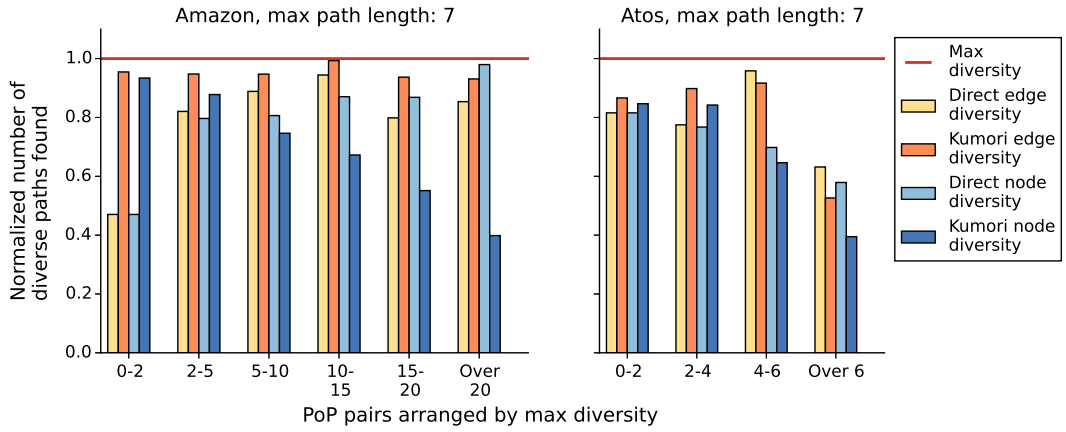


FIGURE 6.6 – Nombre de chemins trouvés normalisé en fonction du type de nœud pour Amazon et Atos

Scores de diversité obtenus avec Kumori

Nous évaluons le bénéfice, en termes de résilience, de l'architecture Kumori en utilisant le score de diversité présenté dans la section 6.3.2.

Le tableau 6.3 montre les scores EPD arc-disjoints et nœud-disjoints obtenus pour toutes les paires de PoPs des CSPs. On observe que l'utilisation de l'architecture Kumori améliore la diversité globale des chemins entre les paires de PoPs des CSPs.

TABLE 6.3 – Scores de diversité EPD pour les paires de PoPs des CSPs

Type	Amazon			Atos		
	Direct	Kumori	gain de Kumori	Direct	Kumori	gain de Kumori
EPD **arc-disjoint**	0.50637	0.55877	**10.35 %**	0.17020	0.23953	**40.73 %**
EPD **nœud-disjoint**	0.50261	0.53413	**6.27 %**	0.16431	0.21383	**30.14 %**
EPD **arc-disjoint, paires où** $EPD_{Kumori} = 0$ **ou** $EPD_{Direct} = 0$	0	0.87300	N.A.	0.02828	0.81320	N.A.
EPD **arc-disjoint, pas div. nulle**	0.95303	0.95178	**-0.13 %**	0.85520	0.84459	**-1.24 %**
EPD **nœud-disjoint, paires où** $EPD_{Kumori} = 0$ **ou** $EPD_{Direct} = 0$	0.13984	0.74271	N.A.	0.18118	0.66518	N.A.
EPD **nœud-disjoint, pas div. nulle**	0.94669	0.93433	**-1.31 %**	0.86844	0.85199	**-1.89 %**

Dans les deux cas, le gain est plus important pour la diversité de chemins à arcs disjoints que pour la diversité des chemins à sommets disjoints. Pourtant, on peut observer que le gain en termes de diversité est beaucoup plus important pour Atos que pour Amazon : il est multiplié par un facteur 4 environ.

Si l'on regarde les deux dernières lignes du tableau 6.3, on constate que l'architecture Kumori n'améliore pas les scores EPD arc-disjoints ou nœud-disjoints pour les paires de nœuds qui sont déjà connectées à travers 2 chemins disjoints ou plus. Au contraire, certaines paires de PoPs bénéficient fortement de l'utilisation de l'architecture Kumori pour augmenter leur diversité de chemins accessibles. ceci est montré par le gain affiché par les scores EPD arc-disjoint et nœud-disjoint obtenus pour les paires de PoPs pour lesquelles EPD_{Kumori} ou EPD_{Direct} est égal à 0 dans le tableau 6.3. En examinant de plus près nos données pour les paires de PoPs d'Amazon, l'utilisation de l'architecture Kumori donne accès à au moins un second chemin à arcs disjoints pour 194 paires de PoPs connectés entre eux par au plus un chemin disjoint directement par Internet. Il donne accès à au moins un deuxième chemin à nœuds disjoints pour 170 paires de PoPs connectés entre eux par au plus un

chemin disjoint directement par Internet. Nous pouvons conclure que Kumori améliore la diversité de chemins pour un nombre significatif de paires de nœuds pour lesquelles un seul chemin peut être trouvé sur Internet. Le gain est moins important pour les paires de nœuds qui bénéficient déjà d'une diversité suffisamment importante sur Internet.

6.4.3 Influence de différents paramètres de construction de l'architecture Kumori sur les gains en termes de résilience

Dans l'évaluation réalisée à ce stade, nous avons utilisé un *overlay* Kumori constitué de 5 points d'inflexion de routage sélectionnés pour la diversité de leur localisation. En outre, dans l'exécution de notre algorithme de découverte de chemin, nous avons limité le nombre d'ASs traversé par un chemin à 4. Ces choix peuvent avoir un impact sur les propriétés de l'architecture Kumori et sur sa capacité à assurer la résilience des connexions entre les PoPs d'un CSP. Dans cette section, nous examinons l'influence de deux paramètres : la politique de placement des points d'inflexion de routage de Kumori et le nombre de points d'inflexion de routage utilisés dans l'*overlay*.

Influence de la politique de placement des points d'inflexion de routage :

Nous comparons le nombre moyen de chemins disjoints trouvés parmi les paires de PoPs d'Amazon pour deux configurations de Kumori se différenciant par la politique adoptée pour placer les 5 points d'inflexion de routage utilisés. Pour la première configuration, nous avons choisi de placer nos points d'inflexion de routage au sein d'IXPs géographiquement diversifiés. Notre objectif était de couvrir les différents continents où Amazon est présent. C'est la politique que nous avons initialement adoptée dans l'évaluation principale. Dans cette configuration, les points d'inflexion de routage sont situés aux IXPs de Francfort (DE-CIX), de Sao Paolo (PTT), de Hong-Kong (HKIX), de New-York (NYIIX) et de Seattle (SIX). Dans la deuxième configuration, nous avons utilisé une autre stratégie pour placer nos points d'inflexion de routage. Nous avons cherché quels étaient les cinq IXPs les plus importants en termes de nombre d'ASs membres dans la base de données PeeringDB, puis avons choisi d'y placer nos points d'inflexion de routage. En conséquence, les points d'inflexion de routage ont été placés à Amsterdam (AMS-IX), Londres (LINX), Francfort (DE-CIX), Sao Paulo (PTT) et Hong-Kong (HKIX).

Les résultats de l'évaluation de la diversité de chemins effectuée pour ces deux configurations de l'architecture Kumori sont présentés sur la figure 7.1. Cette figure montre une différence mineure entre la diversité de chemins obtenue avec les deux politiques de placement de points d'inflexion. Pourtant, on peut constater qu'il est préférable de favoriser la diversité géographique dans le placement des points d'inflexion de routage.

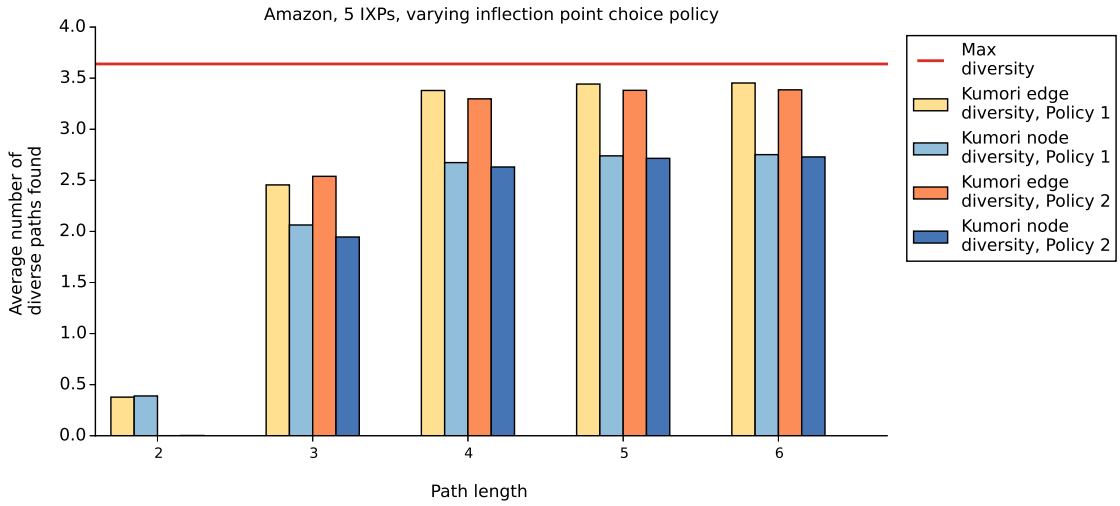


FIGURE 6.7 – Nombre moyen de chemins trouvés entre les PoPs d'Amazon en fonction de la politique de placement des points d'inflexion de routage

Influence du nombre de points d'inflexion de routage utilisés dans l'architecture Kumori :

Nous comparons maintenant différentes configurations de Kumori dans lesquelles un nombre variable de points d'inflexion de routage est utilisé. Dans cette expérience, nous examinons les performances en termes de résilience d'*overlays* Kumori dotés de 3, 5, 7 et 10 points d'inflexion de routage. Les IXPs où ces points d'inflexion de routage sont placés sont choisis en suivant la même politique de placement. A cet effet, nous choisissons de placer les nœuds aux IXPs les plus importants en termes de nombre d'ASs participants. Nous effectuons une évaluation de la diversité de chemins à arcs disjoints et à sommets disjoints pour Amazon et Atos.

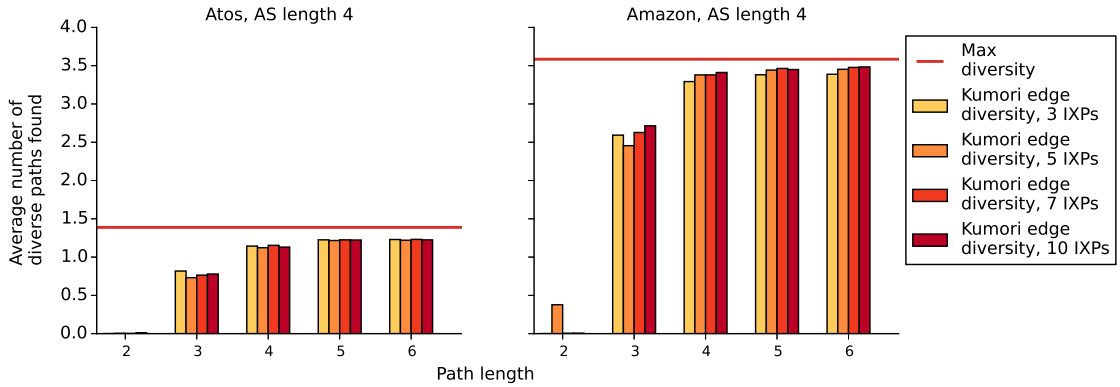


FIGURE 6.8 – Nombre moyen de chemins arc-disjoints trouvés entre les paires de PoPs d'Amazon et d'Atos en fonction du nombre de points d'inflexion de routage utilisés par l'architecture Kumori

La figure 6.8 représente l'évolution du nombre de chemins à arcs disjoints en

fonction de la longueur maximale du chemin pour les différentes configurations de l'architecture Kumori que nous considérons. Ces résultats expérimentaux montrent que le nombre de points d'inflexion de routage dans l'*overlay* Kumori a une influence marginale sur la diversité de chemins obtenue pour les paires de PoPs d'Amazon et d'Atos. En effet, les performances des différentes configurations sont extrêmement similaires au point d'être indiscernables pour des chemins de 5 sauts ou plus.

Notre étude de l'influence de la politique de placement des points d'inflexion de routage et du nombre de points d'inflexion de routage dans l'*overlay* est potentiellement très intéressant pour les personnes qui seraient amenées à déployer l'architecture Kumori. Ils montrent qu'il est plus avantageux de constituer l'*overlay* de manière à disposer d'un nombre limité de points d'inflexion de routage géographiquement diversifiés plutôt que de disposer d'un plus grand nombre de nœuds localisés uniquement au sein des IXPs les plus grands. Cette propriété peut aussi être vue comme une confirmation du fait que la création et le développement d'IXPs locaux de taille modeste contribue à renforcer la résilience d'Internet.

6.4.4 Comparaison de l'architecture Kumori avec l'architecture proposée par Kotronis

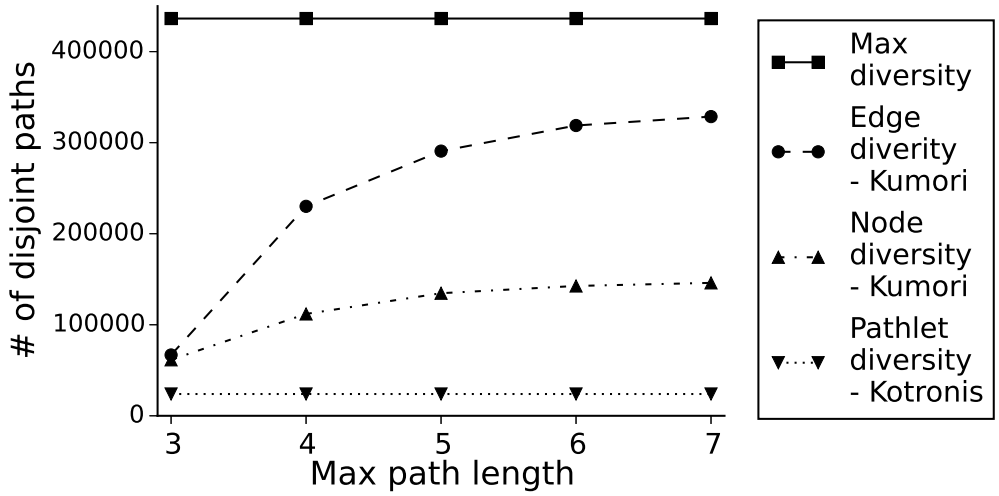


FIGURE 6.9 – Fonction de répartition du nombre de chemins arc-disjoints, nœud-disjoints et AS-disjoints entre les 54 plus grands IXPs

Dans cette section, nous comparons l'architecture Kumori avec une autre architecture *overlay* utilisant des nœuds localisés à différents IXPs qui a été présentée récemment. En effet, Kotronis a présenté dans [KKR+16] un *overlay* dans lequel certains IXPs sont instrumentés de manière à offrir aux ASs membres des connexions alternatives aux routes Internet classiques. Alors que dans Kumori la conception de

l'architecture est motivée par un objectif de résilience, Kotronis vise à offrir de meilleures performances en termes de latence ou de gigue aux utilisateurs de son architecture. Une autre différence entre les deux architectures consiste dans le fait que, dans leurs travaux, Kotronis *et al.* supposent qu'entre les différents IXPs le trafic suivra un *pathlet*, *i.e.* un chemin composé de nœuds se trouvant au sein d'un AS unique. Kumori propose une approche plus générale, dans laquelle les chemins doivent simplement se conformer à la politique de routage *valley-free*, en traversant potentiellement plusieurs ASs. Dans la section précédente, les bénéfices en termes de résilience observés pour Kumori sont également valables pour l'*overlay* de Kotronis. En effet, dans notre évaluation, les chemins utilisés par Kumori ne traversaient qu'un seul nœud placé à un IXP, alors que la différence entre Kumori et l'*overlay* proposé par Kotronis réside dans la façon dont les chemins entre plusieurs IXPs sont gérés.

Nous comparons les deux architectures *overlays* en recherchant les chemins disjoints offerts par chacune des deux architectures entre les 54 plus grands IXPs en terme de nombre d'ASs membres. Nous évaluons la diversité de chemins proposée par l'architecture de Kotronis en déterminant le nombre de chemins où les ensembles des ASs traversés sont disjoints. Nous déterminons le nombre de chemins à arcs disjoints et à sommets disjoints accessibles en utilisant Kumori comme dans les sections précédentes. Les résultats de cette évaluation sont présentés sur la figure 6.9.

On peut observer que la diversité est significativement plus élevée en utilisant l'architecture Kumori en comparaison avec l'architecture proposée par Kotronis. Ceci est lié au choix effectué par ce dernier d'utiliser des pathlets constitués de nœuds appartenant à un AS unique pour acheminer le trafic entre les IXPs. Ce choix vient du souhait de Kotronis d'effectuer un contrôle de certaines métriques de QoS sur ces chemins, ce qui est plus aisé si l'ensemble des nœuds du chemin sont sous la responsabilité d'un seul fournisseur de service Internet. Pour Kumori, la diversité des chemins à arcs disjoints est significativement plus élevée que la diversité des chemins à sommets disjoints, et cette différence augmente avec la longueur maximale du chemin. Cette évaluation montre que l'un des avantages majeurs de l'architecture Kumori est sa capacité à utiliser des chemins multi-AS sur Internet entre différents IXPs tout en permettant une relaxation des politiques de routage BGP par l'utilisation de points d'inflexion de routage aux IXPs. Les résultats obtenus par Kotronis montrent que cette architecture est moins adaptée que Kumori pour améliorer la diversité des chemins inter-PoPs.

6.5 Conclusion

L'évaluation que nous avons faite de la diversité de chemins que Kumori peut offrir aux fournisseurs de services Cloud relativement à la diversité de chemins accessible directement sur Internet souligne les avantages fournis par cette architecture. En moyenne, l'utilisation de Kumori améliore le nombre de chemins à arcs disjoints accessibles de 10,35% pour Amazon et de 40,73% pour Atos, tout en augmentant le nombre de chemins à sommets disjoints de 6,27% pour Amazon et de 30,14% pour Atos. Une étude détaillée montre que l'utilisation de Kumori est particulièrement intéressante pour les paires de PoPs qui souffrent d'une diversité topologique limitée. Nous pouvons également souligner que même lorsque l'utilisation de Kumori n'améliore pas de façon significative le nombre de chemins disjoints accessibles, ces chemins disjoints sont généralement plus courts en termes de nombre de sauts, ce qui a un impact sur leur performance et sur notre capacité à les surveiller.

Un examen approfondi de l'influence de certains paramètres de construction de l'architecture sur ses performances a souligné que l'usage de Kumori permet d'améliorer significativement la résilience des connexions entre PoPs sans qu'il soit nécessaire de déployer un grand nombre de points d'inflexion de routage aux IXPs les plus grands. Notre évaluation souligne qu'il est préférable d'utiliser un ensemble de points d'inflexion de routage géographiquement diversifiés soigneusement sélectionnés de manière à fournir une meilleure diversité de chemins aux utilisateurs de l'architecture. Cette constatation est particulièrement intéressante d'un point de vue économique car le coût de mise en place, d'exploitation et de maintenance d'une architecture *overlay* telle que Kumori dépend du nombre de nœuds qui la constituent.

Chapitre 7

Le modèle économique de l'architecture Kumori

Contents

Dans ce chapitre, nous évaluons l'architecture Kumori présentée dans les chapitres précédents sous un angle économique. Au-delà des bénéfices techniques en termes de résilience des connexions entre datacenters liés à l'utilisation de l'architecture Kumori, nous voulons analyser la viabilité économique de cette architecture pour les fournisseurs de services Cloud (CSP). Afin de présenter le modèle économique de l'architecture Kumori, nous effectuons d'abord un rappel d'ordre général sur le marché

des interconnexions entre opérateurs sur Internet. Nous détaillons ensuite la structure de coûts de plusieurs méthodes d'interconnexion entre opérateurs sur Internet. Nous évaluons l'efficacité économique de l'architecture Kumori en comparant celle-ci à une stratégie de connectivité où un CSP établit un maillage de connexions privées à longue distance entre ces centres de données.

7.1 Rappels sur les aspects économiques du cœur d'Internet

Internet est constitué d'un ensemble de réseaux hétérogènes, assemblages de routeurs IP interconnectés par des liens à longue distance. Ces liens sont eux-mêmes agrégés dans des liaisons optiques WDM (Wavelength Division Multiplexing). Dans cette thèse, nous ne traitons que des connexions IP, sur la couche 3 du modèle OSI. Les réseaux IP hétérogènes constituant l'Internet sont appelés Systèmes Autonomes (AS). Ils sont gérés par des opérateurs de télécommunications, par des fournisseurs d'accès Internet (FAI ou ISP) ou par des fournisseurs de services Cloud (CSP). Les CSP, les fournisseurs de contenus ou les services de diffusion de vidéos fournissent des services aux utilisateurs finaux (entreprises, administrations publiques ou particuliers). Dans ce contexte, chaque AS sert un ensemble d'utilisateurs finaux. Sa responsabilité consiste à connecter ces utilisateurs au reste du monde, *i.e.* à leur fournir une méthode pour atteindre n'importe quel autre AS sur Internet. Du point de vue administratif, un AS est identifié sur Internet par un numéro d'AS qui est unique. Ce numéro AS est délivré par un registre Internet régional tel que RIPE, ARIN ou APNIC. En outre, cet AS peut également disposer d'une plage d'adresses IP qui lui sont propres. Un AS peut se connecter aux autres AS d'Internet en utilisant deux types de mécanismes : le *transit* et *peering*.

Le *transit* peut être défini comme étant le service fourni par un FAI pour connecter les utilisateurs d'un de ses clients au reste d'Internet. Dans une connexion de transit, le client paie pour envoyer et recevoir du trafic réseau par l'intermédiaire de son fournisseur de transit. Pour ce faire, du point de vue du routage, le fournisseur de transit annonce les préfixes IP appartenant à son client au reste du monde et publie vers son client toutes les destinations qu'il peut atteindre. Le client paie pour envoyer et recevoir du trafic depuis son fournisseur de connectivité. Le prix de ce service est généralement fixé en fonction d'une consommation de bande passante. Un AS donné peut utiliser simultanément plusieurs fournisseurs de transit pour des raisons d'arbitrage des coûts ou de résilience. Dans ce cas, nous qualifions ce système autonome d'AS multi-attaché ou *multihomed*.

Par ailleurs, le *peering* est une méthode d'interconnexion dans laquelle deux ASs

acceptent d'échanger du trafic à partir de et vers leurs utilisateurs respectifs, souvent gratuitement. Du point de vue du routage, les deux systèmes autonomes annoncent à leurs pairs les préfixes IP de leurs utilisateurs ou de leurs clients. Une relation de peering n'est pas transitive : un réseau ne peut pas annoncer les préfixes IP annoncés de ses pairs à ses autres pairs. La plupart du temps, les accords de peering sont pris entre des réseaux qui échangent du trafic de manière équilibrée : leurs utilisateurs envoient et reçoivent une quantité semblable de trafic d'un AS à l'autre. Les déséquilibres importants dans ces échanges peuvent l'objet d'une compensation financière pour veiller à l'équilibre de la relation économique entre pairs.

Pour s'échanger du trafic, les systèmes autonomes doivent se connecter à leurs fournisseurs de transit ou à leurs pairs. Cela peut prendre la forme de liens d'interconnexion bilatéraux privés reliant les infrastructures réseau de chaque partie. Les systèmes autonomes désireux de se connecter à un nombre assez important de réseaux peuvent installer des équipements sur des sites d'interconnexion privés spécifiques ou sur des points d'échange Internet (IXP). Les points d'échange Internet se sont progressivement développés sur Internet avec l'augmentation du trafic Internet et du nombre d'AS. Les IXP jouent le rôle de facilitateur d'interconnexion. Au sein des IXPs, les systèmes autonomes peuvent interagir avec d'autres ASs ou établir des liens de transit avec des fournisseurs de services Internet. Ainsi, les IXP sont des sortes de "zones démilitarisées" où les interconnexions entre opérateurs sont facilitées. Les systèmes autonomes qui souhaitent participer à un IXP doivent payer une cotisation pour en devenir membre. Cette cotisation est payée à l'organisation qui gère l'IXP en fonction de la capacité d'interconnexion demandée. Les ASs doivent également payer les coûts de colocation pour faire héberger et exploiter leurs équipements sur le site de l'IXP.

Les systèmes autonomes ne s'appuient pas exclusivement sur le peering ou sur le transit pour se connecter au monde extérieur. La plupart des ASs utilisent simultanément les deux stratégies pour optimiser leurs coûts de connectivité tout en maintenant un niveau suffisant de résilience et de qualité de service. Pourtant, les liens de transit et de peering ne peuvent pas être utilisés de la même manière. Comme cela a été expliqué dans la section 2.3.1, les relations économiques induites par les accords de transit ou de peering aboutissent à une asymétrie des relations entre les systèmes autonomes, et donc à l'utilisation de politiques de routage différentes sur ces deux types de liens.

Le transit et le peering sont utilisés pour connecter des ASs à Internet. La connectivité Internet est souvent fournie par un opérateur sous un mode contractuel dans lequel cet opérateur a une obligation de moyen et non de résultats : c'est ce qu'on appelle un service *best effort*. Cependant, certains systèmes autonomes peuvent avoir besoin de connecter des datacenters ou des sites distants tout en maintenant une

qualité de service donnée en termes de latence, de gigue, (*i.e.* de variance du délai de propagation de paquets IP successifs d'un même flux de données), de bande passante garantie ou de fiabilité. La fiabilité peut être évaluée comme la probabilité qu'un paquet soit perdu en raison de l'expiration de son TTL (Time-to-Live). Dans de tels cas, les ASs peuvent utiliser des liens privés à longue distance exploités par de grands opérateurs réseau sous la forme de circuits MPLS ou d'interconnexions Ethernet de qualité opérateur. Le prix de ces liens à longue distance dépend de la capacité du lien, de son niveau de fiabilité et de l'emplacement des sites qu'il interconnecte.

Le but de l'architecture Kumori est de fournir une méthode de connectivité alternative intermédiaire entre, d'une part, le service *best effort* fourni de manière usuelle par les fournisseurs d'accès Internet et, d'autre part, un maillage de connexions privées constitué de liens à longue distance fournis par des opérateurs distincts. Dans la section suivante, nous détaillons la structure des coûts des différentes méthodes de connectivité présentées dans ce chapitre.

7.2 Analyse des coûts associés aux différentes méthodes de connectivité

Les différentes méthodes de connectivité présentées dans la section précédente ne diffèrent pas seulement par leurs caractéristiques techniques ou de leurs avantages relatifs en termes de connectivité. Ils diffèrent également par leur structure de coûts pour les systèmes autonomes clients. Dans cette section, nous présentons les structures de coûts de ces différentes méthodes. Nous fournissons ensuite des estimations de leur coût en mai 2016. Ces prix sont souvent couverts par des accords de confidentialité entre les fournisseurs et leurs clients. Dans cette thèse, nous utilisons les données recueillies à partir de présentations faites lors des réunions du RIPE ou lors de conférences par un cabinet d'analystes, Telegeography
([SHK15], [Bry16], [BBM16]. Nous avons complété ces données par des éléments extraits d'articles techniques rédigés par de grands CSPs ([Pri14]) ou par des informations échangées sur la liste de diffusion de NANOG, une communauté regroupant des administrateurs de réseaux du monde entier ([SAH]). Ces données nous donnent des estimations approximatives des prix qui ont cours sur le marché mondial de la connectivité. Nous utiliserons ces données dans la suite de ce chapitre pour calculer le coût et la viabilité économique de l'architecture Kumori.

7.2.1 Modèle économique du transit

Comme expliqué précédemment, les systèmes autonomes peuvent utiliser les services de transit de certains fournisseurs de services Internet pour accéder à Internet au-delà de leur voisinage immédiat. Le transit est un service commercial pour lequel les systèmes autonomes paient des droits d'accès aux fournisseurs de services Internet. Sur le marché mondial de l'Internet, le trafic de transit est payé en fonction de la bande passante consommée sur un lien donné, sur la base de la méthode du 95^{e} percentile. D'après cette méthode, la bande passante utilisée par un système autonome client sur ses liens de transit avec un FAI est mesurée à intervalles de temps réguliers. Ces mesures sont synthétisées, et on décide de retenir la valeur de consommation de bande passante la plus élevée après avoir enlever les 5% de mesures les plus importantes. Cette mesure de consommation de bande passante est alors utilisée pour calculer le prix du transit pour ce client. En outre, les systèmes autonomes clients s'engagent souvent à une utilisation de bande passante minimale qui couvre les coûts d'exploitation des liens de transit. Le prix unitaire d'un lien de transit est donné pour un mégabit par seconde et par mois, comme précisé dans [SHK15], [BBM16] et [Bry16].

Le coût du transit n'est pas uniforme partout dans le monde. Dans certaines régions très bien connectées, comme en Europe ou en Amérique du Nord, le transit coûte environ 1$ par Mbps par mois, tandis qu'en Australie, le transit peut coûter jusqu'à 18$ par Mbps par mois. Dans notre évaluation économique de l'architecture Kumori, nous considérons un prix moyen de transit dans 5 grandes régions : l'Asie, l'Amérique du Nord, l'Europe, l'Australie et l'Amérique du Sud. Ces prix sont déterminés à partir des données présentées dans [SHK15], [BBM16] et [Bry16]. Le tableau 7.1 présente les prix que nous utilisons dans notre évaluation économique.

TABLE 7.1 – Prix du transit dans 5 grandes régions du monde

Region	Prix du transit ($ par Mbps par mois)
Europe	1
Asie	9
Amérique du Nord	1
Amérique du Sud	17
Australie	18

Les AS clients peuvent acheter du transit à des fournisseurs de services Internet sur la base des coûts fournis dans le tableau 7.1 s'ils peuvent transporter leur trafic Internet vers un site d'interconnexion où les fournisseurs de services Internet sont présents ou vers un point d'échange Internet. Pour relier leurs datacenters à ces points d'interconnexion, les ASs clients utilisent un lien d'interconnexion à courte distance. Une telle interconnexion est un lien privé et dédié. Le coût de ces liens loués dépend de divers paramètres :

- La bande passante maximale que le système autonome client cherche à consommer,
- La distance entre le datacenter et un site appartenant au fournisseur du lien dédié,
- Les accords de qualité de service sur ce lien dédié,
- La capacité du client à négocier,
- L'importance du fournisseur sur le marché.

Pour notre évaluation, nous utilisons une estimation du coût des interconnexions de 10 Gbps avec les IXPs présentée dans le tableau 7.2 en nous basant sur les informations extraites des présentations faites par les analystes de Telegeography ([SHK15], [BBM16], [Bry16]).

TABLE 7.2 – Prix d'une interconnexion de 10 Gbps à un IXP dans 5 grandes régions du monde

Region	Prix de l'interconnexion ($ par mois pour 10 Gbps)
Europe	3000
Asie	9000
Amérique du Nord	3000
Amérique du Sud	9000
Australie	9000

7.2.2 Modèle économique du peering

Lorsqu'un système autonome échange du trafic de manière quasi-symétrique avec un autre réseau, il peut établir une relation directe de pair à pair avec cet autre système autonome : il s'agit d'un lien de *peering*. Comme indiqué précédemment, le peering peut se faire sur des liens privés dédiés reliant deux réseaux. Ces liens peuvent être établis sur des sites de peering privés ou à des points d'échange Internet. Les points d'échange Internet sont des emplacements neutres sur Internet où plusieurs acteurs du marché peuvent se rencontrer et interconnecter leurs infrastructures. Le rôle de ces points d'échange Internet a été mis en évidence dans [ACF+12]. Dans cet article, les auteurs soulignent l'importance croissante des échanges de trafic sur un grand IXP européen. Les points d'échange Internet (IXP) ont divers modèles de gouvernance. Les IXP peuvent être gérés soit par des organisations à but non lucratif, soit par des entreprises privées qui agissent comme intermédiaires entre des systèmes autonomes.

Pour se connecter à un IXP et à d'autres réseaux pairs, les systèmes autonomes doivent avoir conclu un accord avec cet IXP. Dans la plupart des cas, un AS doit devenir membre d'un IXP en louant un port sur l'un des commutateurs de l'IXP. Le tarif de ces

ports dépend de l'entité qui gère l'IXP et de la bande passante que l'AS cherche à utiliser pour échanger son trafic via cet IXP. Une étude de la liste de prix recueillie auprès de la communauté NANOG par Snijders dans [SAH] montre que les prix des ports chez les IXPs que nous considérons tendent à être relativement similaires au sein des diverses régions mondiales, car les IXPs de ces grandes zones géographiques sont en concurrence pour attirer les systèmes autonomes chez eux. Aussi, les tarifs moyens des ports que nous utiliserons dans notre étude économique sont présentés dans le tableau 7.3.

TABLE 7.3 – Prix d'un port de 10 Gbps chez un IXP dans 5 grandes régions du monde

Region	Prix du port ($ par mois pour 10 Gbps)
Europe	1400
Asie	1000
Amérique du Nord	1200
Amérique du Sud	2000
Australie	2000

Au-delà du coût de location et d'utilisation d'un port à un IXP, les systèmes autonomes qui cherchent à échanger du trafic à un IXP donné doivent investir dans des équipements de routage et de commutation qu'ils doivent installer, gérer et connecter à un ou plusieurs ports des équipements de l'IXP. Ainsi, un AS doit acheter ces équipements et payer les frais de colocation de ces équipements sur les sites des IXPs qui comprennent la location des baies où sont installés les équipements et la fourniture de l'électricité. En outre, les ASs doivent payer les frais d'interconnexion aux IXPs pour transporter leur trafic réseau depuis leurs datacenters jusqu'au site de l'IXP. Dans notre évaluation économique, nous supposons que, en moyenne, chaque système autonome place deux routeurs à chaque IXP auquel il veut participer. Le coût estimé pour une telle opération est d'environ 4000$ pour chaque routeur. Nous pu estimer les frais de colocation moyens dans les différentes régions mondiales à partir des données publiées par Telegeography ([SHK15], [BBM16], [Bry16]. Le tableau 7.4 présente les prix de colocation que nous considérons dans notre modélisation économique.

TABLE 7.4 – Prix de colocation pour un demi-rack dans 5 grandes régions du monde

Region	Prix de colocation ($ par mois pour un demi-rack)
Europe	1400
Asie	1000
Amérique du Nord	1200
Amérique du Sud	2000
Australie	2000

7.2.3 Comparaison du peering et du transit dans l'Internet d'aujourd'hui

Le peering et le transit présentent des avantages techniques et économiques différents pour les systèmes autonomes qui cherchent à accéder à Internet. Dans notre évaluation économique de l'architecture Kumori, nous avons cherché à déterminer comment les services clients des gros opérateurs réseaux, en particulier les fournisseurs de services Cloud, utilisent le transit ou le peering pour transmettre leur trafic. Cloudflare, un fournisseur de services Cloud ayant une présence mondiale, a publié des informations sur son utilisation du peering et du transit dans différentes régions mondiales. Dans cet article, Cloudflare souligne que globalement la part de leur trafic opéré sur le mode du peering augmente de façon permanente. Pourtant, le rapport entre le volume du trafic opéré en peering et le volume du trafic de transit observé auprès des différents IXPs dont ils sont membres diffère selon la région. Ainsi, en Europe, en Amérique du Sud, en Asie ou en Australie, Cloudflare opère de 50 à 60% de son trafic de données en utilisant des liens de peering, alors qu'en Amérique du Nord, 20% seulement du trafic est transféré sur un lien de peering.

Dans notre évaluation économique de l'architecture Kumori, nous devons tenir compte de cette répartition entre peering et transit pour évaluer correctement le coût du trafic réseau traversant les différents points d'inflexion de routage. Le tableau 7.5 présente le pourcentage de trafic opéré en peering observé par Cloudflare dans les différentes régions mondiales.

TABLE 7.5 – Part du trafic opéré en peering dans 5 grandes régions du monde

Region	Part du trafic opéré en peering (% du trafic réseau)
Europe	50%
Asie	55%
Amérique du Nord	20%
Amérique du Sud	60%
Australie	50%

7.2.4 Modèle économique des liens à longue distance

Les systèmes autonomes peuvent vouloir isoler leur trafic WAN de leur trafic Internet pour des raisons opérationnelles. Plusieurs grands fournisseurs de services Cloud utilisent aujourd'hui un ensemble de liens privés à longue distance pour interconnecter des datacenters situés dans différentes régions du monde. Ces mêmes CSPs cherchent également à assurer un niveau suffisant de résilience ou de qualité de service pour leurs clients en utilisant un réseau de liens privés. Ces liens privés sont soit construits spécifiquement par le CSP, ce qui représente de gros investissements, soit loués auprès de gros opérateurs Internet qui construisent ces liens en profitant de

leur infrastructure existante.

Selon les éléments que nous pourrions recueillir à partir des données de Telegeography ([SHK15], [BBM16], [Bry16]) et des discussions avec différents acheteurs dans le domaines des télécommunications, le coût des liens à longue distance peut se décomposer en plusieurs éléments de coût. Il comprend une composante dite longue distance et deux composantes d'accès. La composante longue distance rend compte du prix de la connexion entre deux points de présence (PoPs) distants de l'opérateur du lien. Les composants d'accès couvrent le coût de la connexion reliant ces points d'accès aux sites du client. La composante longue distance est composée d'une composante sous-marine (parfois appelée capacité humide), et d'une composante terrestre (parfois appelée "backhaul"). L'exploitation d'un câble terrestre est généralement plus coûteuse que l'exploitation d'un câble sous-marin.

L'architecture Kumori est destinée à remplacer les réseaux de connexions privées et doublées utilisés traditionnellement par les fournisseurs de services Cloud pour interconnecter leurs datacenters. Aussi, nous devons déterminer un prix moyen pour les liens privés à longue distance dans les différentes régions mondiales et entre ces régions pour évaluer le coût de ces réseaux privés d'interconnexion. Le tableau 7.6 présente le coûts moyens des liens à longue distance que nous avons déterminé en utilisant les données extraites de [SHK15], [BBM16] et [Bry16].

TABLE 7.6 – Prix moyen d'une connexion privée de 10 Gbps au sein de 5 grandes régions du monde et entre ces régions

Région	Prix du lien ($/mois pour 10 Gbps)
Europe	1600
Asie	42000
Amérique du Nord	4250
Amérique du Sud	30000
Australie	8500
Asie - Amérique du Nord	21050
Amérique du Nord - Europe	6700
Europe - Asie	30000
Amérique du Nord - Amérique du Sud	33900
Asie - Australie	30000
Australie - Amérique du Nord	60000

7.2.5 Évolution des coûts dans le temps

Dans les données présentées dans [SHK15], [BBM16] et [Bry16], nous observons une réduction régulière des prix de plusieurs éléments de coûts présentés dans les sections précédentes. Le prix des connexions privées à longue distance et des équipements réseau a tendance à baisser de 20% chaque année. Pareillement, le prix du trafic réseau en transit tend à baisser de 30% par an. Comme ces évolutions des coûts ne

sont pas identiques, elles peuvent induire à la marge un biais dans notre évaluation économique de l'architecture Kumori pour les années à venir. C'est pourquoi nous avons l'intention d'étudier l'impact de ces réductions de coûts sur la viabilité économique de l'architecture Kumori.

7.3 Structure de coût de l'architecture Kumori

Dans l'architecture Kumori, les fournisseurs de services Cloud utilisent des points d'inflexion de routage situés à différents IXPs afin de contrôler le routage de leur trafic sur Internet entre leurs datacenters. Pour les CSPs, mettre en œuvre l'architecture Kumori consiste à placer des points d'inflexion de routage à différents IXPs autour du globe. Dans cette section, nous détaillons le coût de la mise en place et de l'exploitation d'un point d'inflexion de routage Kumori à divers endroits.

Afin de mettre en place un nœud Kumori à un IXP, un CSP doit devenir membre de cet IXP. À cette fin, ce CSP doit louer un port sur un équipement de l'IXP. Ensuite, le CSP doit placer deux routeurs SDN haut de gamme connectés aux ports qu'il a loués chez cet IXP. Le CSP doit payer les frais de colocation de ces deux routeurs haut de gamme sur le site de l'IXP. Ces frais correspondent au coût mensuel de location des baies de serveur chez l'IXP (y compris l'alimentation électrique et le refroidissement des deux routeurs). Dans le secteur de l'IT, le coût des équipements tels que les routeurs ou les serveurs est amorti sur 3 ans. Ainsi, chaque mois, 1/36 ^{e} du coût total de l'équipement est payé par le CSP en plus des frais de colocation. Enfin, le CSP doit payer pour le trafic relayé par le point d'inflexion de routage Kumori à l'IXP. Si nous voulons que l'infrastructure puisse transférer 10 Gigabits par seconde de trafic, la bande passante totale qui doit être disponible à chaque point d'inflexion de routage est de 20 Gigabits par seconde (en additionnant le trafic d'entrée et le trafic de sortie).

Le tarif des éléments qui rentrent dans le coût d'exploitation global d'un point d'inflexion de routage Kumori varient selon l'endroit où ce point d'inflection est exploité. Le tableau 7.7 répertorie les coûts que nous avons déterminé à partir des données présentées dans la section **??**. Nous détaillons les différents éléments qui rentrent dans le calcul du coût d'opération d'un point d'inflexion de routage et nous présentons deux scénarios pour évaluer le prix du trafic réseau opéré par le point d'inflexion. Dans un premier temps, nous considérons que tout le trafic est relayé au moyen d'accords de transit. Ensuite, nous considérons qu'une part de ce trafic est relayée en utilisant des liens de peering.

Si on regarde attentivement les coûts présentés dans le tableau 7.7, on remarque que la connectivité réseau représente la part la plus importante du coût global d'opération d'un nœud Kumori, alors que les coûts d'équipement et d'hébergement

TABLE 7.7 – Coût d'opération d'un nœud de l'architecture Kumori dans différentes régions du monde

Région	Europe	Asie	Amérique du Nord	Amérique du Sud	Australie
Coût d'un port chez l'IXP ($/mois)	1400	1000	1200	2000	2000
Frais de colocation ($/mois)	1700	2000	1400	3000	3000
Coût des équipements ($/mois)	222.22	222.22	222.22	222.22	222.22
Coût réseau pour 10 Gbps en transit ($/mois)	10240	92160	10240	174080	184320
Coût réseau pour 10 Gbps en mix transit/peering ($/mois)	5120	41472	8192	69632	92160
Coût d'opération d'un nœud Kumori, Transit ($/mois)	23 802.22	187 542.22	233 02.22	353 382.22	373 862.22
Coût d'opération d'un nœud Kumori, mix Peering/Transit ($/mois)	13 562.22	86 166.22	19 206.22	144 486.22	189 542.22

chez les IXPs sont relativement similaires d'une région à l'autre. En outre, il existe une forte incitation à essayer d'établir des relations de peering au même titre que les CSP tels que Cloudflare pour réduire les coûts réseau.

7.4 Comparaison de l'architecture Kumori avec des infrastructures réseau WAN privées

7.4.1 Topologies utilisées pour l'évaluation

Afin d'évaluer la pertinence économique de l'architecture Kumori, nous comparons son coût avec le coût d'un maillage inter-datacenters de liens dédiées à longue distance. Afin d'effectuer cette comparaison, nous avons d'abord Cherché quelles étaient les topologies des réseaux inter-datacenters utilisés par les fournisseurs de services Cloud. Nous avons trouvé trois exemples de telles topologies : le réseau WAN B4 utilisé par Google, le réseau inter-datacenter de Facebook et l'infrastructure réseau WAN d'Amazon.

La topologie du réseau B4 a été présentée par Jain *et al.* dans [JKM+13], qui présente l'utilisation de SDN par Google pour optimiser l'utilisation de son réseau WAN

inter-datacenters. Dans le réseau B4, 12 datacenters situés en Asie, en Europe et en Amérique du Nord sont connectés en utilisant 18 liens à longue distance. Pour évaluer l'architecture Kumori, nous faisons l'hypothèse que Google achète du transit Internet à 6 points d'interconnexion pour assurer la connectivité de ses datacenters.

La topologie du réseau inter-datacenter de Facebook est plus difficile à déterminer. A partir du site Web de Facebook, nous savons que l'entreprise dispose de 5 grands centres de données en Amérique du Nord ([FBDf], [FBDb], [FBDc], [FBDd], [FBDa] et d'un centre en Europe ([FBDe]). Nous supposons que Facebook utilise six liens à longue distance pour interconnecter ses datacenters américains, et que deux liens sont utilisés pour relier ces datacenters américains au datacenter européen. En outre, nous supposons que Facebook achète du trafic de transit pour ses datacenters sur 3 sites, 2 en Amérique du Nord et 1 en Europe.

Le réseau inter datacenter d'Amazon a été présenté au public par James Hamilton, vice-président et ingénieur principal pour Amazon Web Services en 2014 au cours de la conférence AWS re :Invent [Van]. Amazon regroupe ses datacenters au sein de régions. Une région peut regrouper de 2 à 5 datacenters. Les régions sont reliées entre elles par des liens à longue distance. Ces liens sont chacun reliés à 2 équipements qui se chargent de l'opération du trafic de transit ou de peering. Le site web d'Amazon [AMZ] présente une carte des différentes régions ainsi que le nombre de datacenters dans chacune d'elles. Dans notre évaluation, nous incluons les datacenters existants et futurs représentés sur cette carte. Cette carte ne révèle aucune information sur le nombre et sur la disposition des liaisons longue distance reliant les régions. Nous faisons des hypothèses raisonnables sur ces liens en nous conformant à la densité de liens utilisée par Google et à différents éléments communiqués par James Hamilton dans sa présentation.

Outre ces trois topologies existantes, nous avons envisagé six autres configurations imaginaires pour mieux étudier la viabilité économique de l'architecture Kumori. Tout d'abord, nous avons regroupé les trois topologies existantes dans un grand réseau inter-datacenters. Nous avons incorporé les datacenters et les liens à longue distance utilisés par Amazon, Facebook et Google, tout en ne gardant que les sites d'interconnexion et de peering utilisés par Amazon. En évaluant la pertinence de l'architecture Kumori pour cette topologie, nous voulons évaluer l'effet de la mutualisation des éléments de l'architecture pour plusieurs CSPs sur les coûts d'exploitation de l'architecture. Ensuite, nous avons construit cinq autres topologies réseau en plaçant l'ensemble des nœuds de la topologie B4 dans chacune des cinq régions globales que nous étudions. A cet effet nous avons étudié la possibilité d'utiliser Kumori pour remplacer un réseau de 18 liens à longue distance reliant 12 datacenters situés en Europe, en Asie, en Amérique du Nord, en Amérique du Sud et en Australie. Compte tenu de la différence entre les marchés de connectivité dans ces régions, ces

cinq configurations imaginaires nous aideront à comprendre comment les prix du trafic opéré sur des liens à longue distance et sur le mode du transit affectent la viabilité économique de l'architecture Kumori en comparaison avec une stratégie consistant à établir un maillage de liens privés redondés ou non.

7.4.2 Comparaison des structures de coût de Kumori et d'une infrastructure d'interconnexion classique

En utilisant les éléments de coûts présentés dans les sections précédentes, nous comparons la structure de coût de l'architecture Kumori avec celle d'un ensemble de paires de liens privés reliant les datacenters d'un CSP. Dans cette évaluation, nous avons considéré que chaque lien privé a une capacité de 10 gigabits par seconde, et que l'architecture Kumori doit être capable de relayer le même volume de trafic à chaque point d'inflexion de routage au sein de l'*overlay*. Chaque liaison privée entre datacenters consiste en une paire de liens privés pour prendre en compte la stratégie de résilience classique utilisée par les fournisseurs de services Cloud.

En ce qui concerne l'*overlay* Kumori, nous avons détaillé quatre cas. Tout d'abord, nous avons considéré que le CSP n'est pas connecté à des sites de peering pour acheminer son trafic réseau, et nous incluons l'amortissement du prix d'installation et de maintenance d'équipements sur ces sites d'interconnexion dans les coûts d'opération de l'architecture. Ensuite, nous supposons que ce CSP dispose déjà d'un accès aux sites d'interconnexion. Dans ce cas, nous considérons le coût de l'augmentation de la capacité des liens entre les datacenters du CSP et ces sites. Pour les deux situations, nous avons considéré deux stratégies de connectivité : Dans la première stratégie, le CSP exploitant l'*overlay* Kumori parvient à établir des relations de peering de même nature que Cloudflare ([Pri14]). Dans la deuxième stratégie, le CSP doit payer des frais pour opérer l'ensemble de son trafic réseau en transit.

Le tableau 7.8 présente les coûts associés aux différentes solutions de connexion inter-datacenter que nous étudions pour les différentes topologies que nous avons considérées. Dans ce tableau, le prix du réseau de liens privés redondés entre datacenters est pris comme référence et le prix des différentes configurations de l'architecture Kumori est comparé à cette référence. Afin de faciliter la compréhension de ce tableau, les cellules colorées en rouge indiquent que le coût de l'*overlay* Kumori est plus élevé que celui de la méthode classique. À l'opposé, les cellules colorées en vert indiquent que les coûts associés à l'*overlay* Kumori sont plus faibles.

Tout d'abord, il convient de constater qu'un CSP qui cherche à utiliser l'architecture Kumori doit être prêt à négocier le plus d'accords de peering possibles pour abaisser le coût d'opération de son trafic réseau aux différents points d'inflexion de Kumori. En effet, le coût du trafic réseau aux points d'inflexion représente une part importante du

TABLE 7.8 – Comparaison des coûts de l'architecture Kumori et d'une infrastructure d'interconnexion privée. Les prix sont donnés en $/mois

Modèle	Interco. privée	Kumori, Installation infra peering, transit	Kumori, Augmentation capacité peering, transit	Kumori, Installation infra peering, mix transit/ peering	Kumori, Augmentation capacité peering, mix transit/ peering
B4 (Google)	289 100	634 804,44	568 471,11	370 612,44	304 279,11
Facebook	79 933,33	158 813,33	131 846,67	125 021,33	98 054,67
Amazon	1 532 733,33	3 157 146,67	2 800 791,11	1 707 674,67	1 351 319,11
Global	1 901 766,66	3 571 306,67	3 148 951,11	1 972 330,67	1 549 975,11
B4 Europe	62 400	226 417,78	170 484,44	144 497,78	88 564,44
B4 Asie	1 516 800	1 608 337,78	1 481 004,44	797 329,78	669 996,44
B4 Amérique du Nord	157 800	222 417,78	169 484,44	189 649,78	136 716,44
B4 Amérique du Sud	1 084 800	2 935 057,78	2 795 724,44	1 263 889,78	1 124 556,44
B4 Australie	310 800	3 098 897,78	2 959 564,44	1 624 337,78	1 485 004,44

coût total de l'architecture Kumori. En outre, en observant notamment les résultats obtenus pour la topologie B4 en Asie et en Amérique du Nord, on constate que la viabilité économique de l'architecture Kumori comparativement à une stratégie de connectivité classique augmente avec la différence de prix entre le coût du trafic réseau de transit et le coût des liens à longue distance. En effet, si l'on considère un réseau inter-datacenters ayant la même densité de liens privés que la topologie B4, on constate que l'utilisation de l'architecture Kumori devient rentable lorsque le rapport de coût entre le coût du megabit par seconde en transit et sur un lien privé est inférieur à 2,2, comme c'est le cas pour La topologie B4 en Asie (de 0,98 à 2,18) ou pour la topologie B4 en Amérique du Nord (1,86). Cette différence de prix entre le transit et les liens à longue distance explique également les bons résultats obtenus avec l'architecture Kumori pour la topologie Amazon et pour la topologie globale regroupant les trois CSPs.

7.4.3 Évolution dans le temps de la structure de coût de l'architecture Kumori

Comme expliqué dans la section 7.2.5, le prix des différents éléments composants le coût de l'architecture Kumori varie d'une année à l'autre de manière disparate. En effet, le prix du transit réseau diminue plus rapidement que le prix des équipements et des

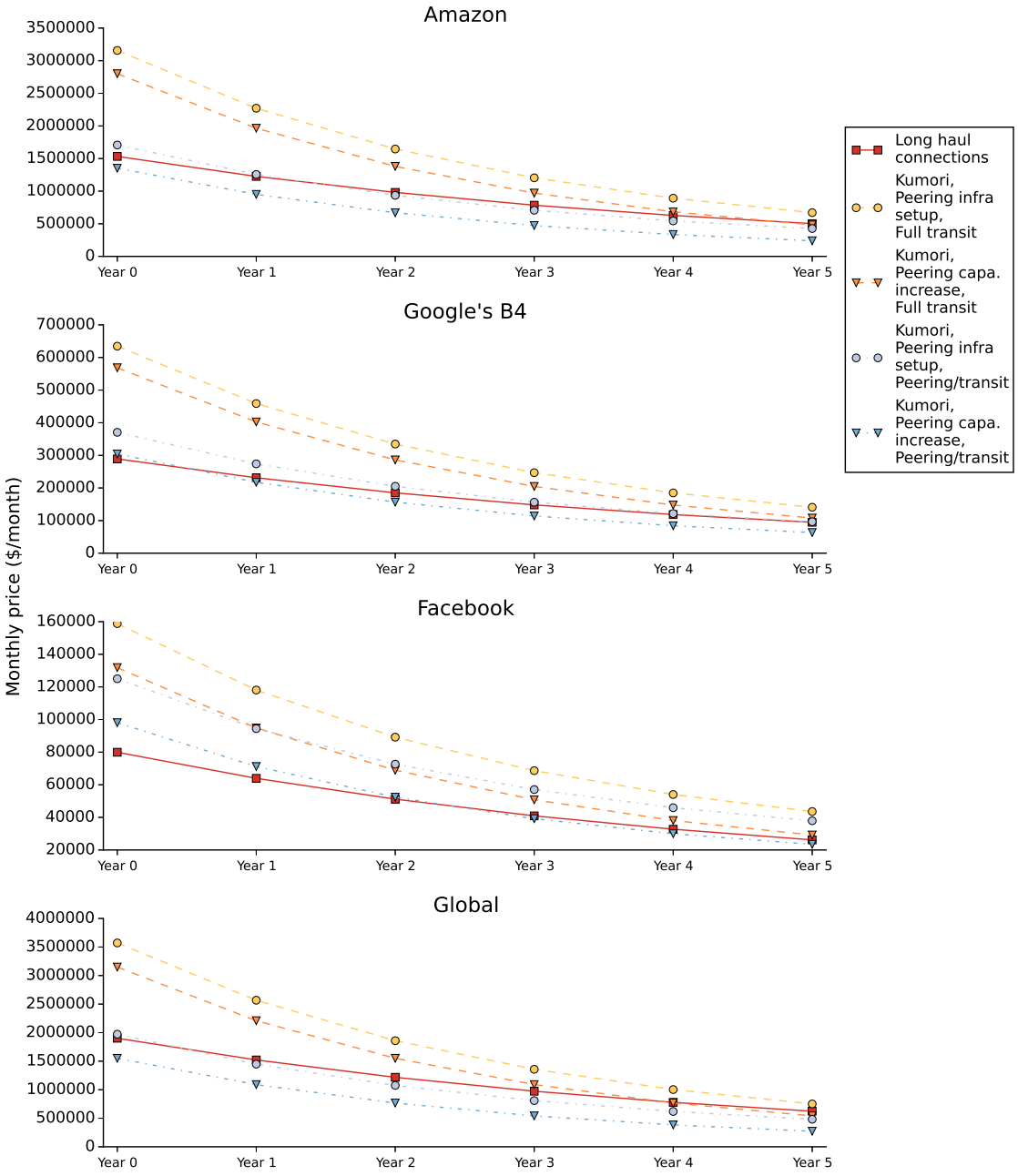


FIGURE 7.1 – Influence de l'évolution temporelle des coûts d'interconnexion sur le coût d'opération des différents modèles d'interconnexion considérés pour différents CSPs

liens à longue distance. La figure 7.1 illustre l'effet de cette réduction de prix à un horizon de cinq ans pour cinq stratégies de connectivité étudiées dans la section 7.4.2 et pour 4 topologies de réseau inter-datacenter : B4 de Google, Amazon, Facebook et notre topologie imaginaire englobant les 3 CSPs précédents. Dans cette comparaison, nous supposons que les besoins en termes de trafic réseau n'augmentent pas au cours des cinq ans. Une observation des courbes d'évolution des coûts indique que l'utilisation de l'architecture Kumori tend à être de plus en plus intéressante au fur et à

mesure du temps. Ceci s'explique par le fait que le rythme de baisse des prix du transit réseau est plus élevé que le rythme de diminution des prix des liens à longue distance. Par conséquent, le ratio entre le coût du trafic réseau en transit et le coût d'opération du trafic réseau sur un lien privé tend à diminuer d'année en année et passe en dessous de 2,2. Si nous combinons cet effet avec l'augmentation à venir de la demande de trafic réseau, nous prévoyons que la viabilité économique de l'architecture Kumori en comparaison avec les méthodes de connectivité classiques sera plus importante à l'avenir qu'elle ne l'est aujourd'hui dans certaines régions du monde.

7.5 Conclusion

Outre ses possibilités techniques, une analyse de la structure de coût de l'architecture *overlay* Kumori montre qu'elle représente une alternative économiquement viable aux stratégies de connectivité classiques utilisées par les fournisseurs de services Cloud dans certaines conditions. Ainsi, notre étude révèle que la rentabilité de l'architecture Kumori dépend du rapport entre le coût du trafic réseau en transit et sur des liens à longue distance lorsque l'architecture Kumori est utilisée par un CSP unique. Plus le coût du transit est faible, plus l'architecture Kumori est rentable. Aujourd'hui, les marchés nord-américain et asiatique présentent des caractéristiques telles que l'architecture Kumori se révèle être une solution économiquement intéressante pour les CSPs disposants de datacenters dans ces deux régions. Compte tenu de l'évolution du prix des liens privés et du trafic réseau en transit, nous prévoyons que l'architecture Kumori deviendra plus rentable à l'avenir. Ce bénéfice devrait être encore plus perceptible avec l'augmentation de la demande de capacité réseau entre les datacenters liée au développement du *Cloud Computing*.

Chapitre 8

Conclusion

Contents

8.1 Résultats

Dans cette thèse, nous avons cherché à proposer une solution au problème de la résilience du Cloud computing, en visant plus spécifiquement à renforcer les connexions entre les datacenters des fournisseurs de service Cloud (CSP). En effet, ces CSPs gèrent plusieurs datacenters répartis au sein d'un continent ou même globalement dans différentes régions du monde. Chacun de ces datacenters peut accueillir plusieurs milliers de serveurs et des millions de machines virtuelles (VM). Plus la taille des datacenters est importante, plus un CSP opéré un nombre important de ces datacenters, plus l'économie d'échelle fournie par la réutilisation des ressources physiques utilisant les capacités de multiplexage statistiques inhérentes à l'utilisation des technologies de virtualisation est importante. Les grands datacenters des CSPs sont traditionnellement interconnectés par l'intermédiaire de liaisons à longue distance utilisant des fibres optiques. Ces liens sont le plus souvent loués par les CSP aux opérateurs de télécommunications ou déployés spécifiquement dans le cas de certains CSPs importants.

La continuité de service est primordiale pour les CSPs. De multiples pannes peuvent menacer cette continuité. Sur la couche physique, une coupure de fibre, la défaillance d'un émetteur laser ou d'un photodétecteur, etc. induisent *de facto* une panne de lien pouvant impacter l'exécution de calculs répartis. Des dysfonctionnements des couches logiques (par exemple en cas d'adressage erroné d'un équipement ou de débordement

de buffer sur un routeur IP) peuvent également entraîner une panne des connexions entre datacenters. Outre ces pannes qui peuvent affecter les liens entre datacenters, une panne réseau intervenant au sein même des datacenters peut également induire une perturbation du service Cloud opéré par le fournisseur. Aussi, dans cette thèse, nous nous sommes concentrés uniquement sur la résilience de l'infrastructure réseau inter-datacenter.

L'objectif a été de proposer une solution au problème de la résilience des connexions entre datacenters pour contrer les éventuelles défaillances de liens physiques ou logiques. Les approches existantes pour assurer cette résilience reposent sur l'utilisation d'un maillage de liens privés redondés, c'est-à-dire doublés pour faire face à la panne d'un des liens déployés. Cette approche nécessite qu'un opérateur installe, déploie ces liens entre les différents datacenters participants au réseau. Les temps de déploiement sont généralement très longs, de l'ordre de quelques semaines à quelques mois, et ne permettent pas d'associer de manière dynamique ou temporaire un datacenter à un réseau déjà déployé. En outre, dans ces approches, la résilience est un service contractuellement fourni par les opérateurs fournissant les liens réseau. L'efficacité des opérateurs dans les opérations de rétablissement de leur infrastructure peut varier considérablement d'un opérateur à un autre, et dépend souvent des contrats de qualité de service (SLA) qui lient les opérateurs aux fournisseurs de service Cloud. Une telle dépendance vis-à-vis des opérateurs de télécommunication est incompatible avec les attentes des CSPs qui cherchent à assurer proactivement la viabilité et la résilience de leurs infrastructures. Afin de résoudre ce problème, nous avons proposé dans cette thèse une solution permettant aux CSPs de pouvoir contrôler eux-mêmes les liens qui connectent leur datacenters et d'agir en cas de panne d'une de ces connexions. Par ailleurs, nous avons cherché à réduire les délais nécessaires à l'installation et à la configuration du rattachement d'un datacenter à d'autres datacenters déjà interconnectés.

En considérant ces besoins des CSPs, nous avons proposé dans cette thèse une nouvelle architecture *overlay*, Kumori. Nous avons choisi de dénommer notre architecture ainsi en référence au mot japonais signifiant "nuage", après un stage de recherche effectué à Tokyo. L'architecture Kumori consiste en un *overlay* réseau contrôlé de manière centralisée tout comme la technique SDN. L'*overlay* Kumori permet d'opérer un contrôle du trafic réseau au sein de différents datacenters et entre ces centres afin de d'influencer la manière dont les flux réseaux sont routés depuis un serveur donné situé dans un centre de données vers tout autre serveur exploité par le CSP dans l'un de ses centres. Entre les datacenters d'un CSP, l'architecture Kumori consiste en un ensemble de points d'inflexion de routage. Ces points d'inflexion sont situés à différents points d'échange Internet (IXP) qui sont des sites indépendants des différents opérateurs sur Internet. Ces points d'échange proposent un écosystème de

connectivité riche par la présence de nombreux systèmes autonomes (AS) cherchant à y établir des relations de peering afin d'échanger leur trafic réseau avec leurs pairs. Inspiré par les architectures SDN, nous avons choisi de contrôler la politique de routage appliquée par les points d'inflexion de routage au sein de l'*overlay* Kumori par l'intermédiaire d'un contrôleur centralisé. Ce contrôleur est responsable de la collecte de mesures effectuées par les différents nœuds de l'*overlay* pour surveiller l'état des connexions au sein de l'*overlay*. Ces mesures sont utilisées par le contrôleur pour détecter les pannes de lien ou de nœud entre les points d'inflexion. Si une telle panne est détectée, le contrôleur peut demander aux nœuds de l'*overlay* de rerouter le trafic concerné pour éviter la panne.

Nous avons évalué les performances de l'*overlay* utilisé entre les datacenters par l'architecture Kumori afin d'évaluer les caractéristiques des chemins utilisables par l'*overlay* et les comparer aux caractéristiques offertes par d'autres *overlays* plus classiques en termes de longueur de chemin, de diversité de chemin et de nombre de nœuds composants l'*overlay*. Tout d'abord, nous avons comparé l'architecture Kumori avec l'architecture Resilient Overlay Network (RON), un *overlay* de référence dont l'objectif est d'assurer la résilience de connexions entre différents points d'Internet. Pour cette première évaluation, nous avons quantifié la performance relative des deux *overlay* en mesurant la longueur des chemins utilisables par chaque système. Cette évaluation montre que l'architecture Kumori est plus avantageuse que RON en termes de performance pour différents types de CSP. Pour les CSP les plus grands comme Amazon, Microsoft ou Google, la longueur des chemins alternatifs utilisables par le biais de Kumori est similaire à celle des chemins atteignables par un usage de RON. Néanmoins, pour ces grands CSPs, le nombre de nœuds requis dans l'*overlay* pour utiliser ces chemins est plus faible dans Kumori que dans RON. Ce résultat est particulièrement intéressant car l'un des principaux inconvénients de l'architecture RON est son incapacité à fonctionner correctement lorsque l'architecture utilise plus de cinquante nœuds dans l'*overlay*. De plus, le coût de mise en œuvre d'un *overlay* dépend fortement du nombre de nœuds qui le compose. Pour les CSP plus petits tels que Atos ou Dimension Data, l'architecture Kumori offre des chemins alternatifs significativement plus courts que RON. Ainsi, pour ces CSPs, l'utilisation de chemins alternatifs fournis par Kumori entraîne une amélioration des performances par rapport à ce que permet de réaliser l'architecture RON.

Nous avons ensuite évalué le gain en termes de résilience fourni par l'*overlay* Kumori. Pour ce faire, nous avons comparé la diversité de chemins routables offerte par Kumori à la diversité des chemins empruntables sur Internet entre différents points de présence géographiques de fournisseurs de services Cloud. Pour mettre en œuvre cette deuxième évaluation, nous avons construit un graphe dirigé représentant Internet au niveau PoP en nous appuyant sur les données produites et publiées par les projets

iPlane, DRAGON et PeeringDB. Dans ce graphe, les arcs sont marqués de manière à représenter la complexité des relations entre les systèmes autonomes sur Internet. Nous avons utilisé ce graphe pour évaluer le nombre de chemins à arcs disjoints et à nœuds disjoints parmi toutes les paires de PoPs appartenant à deux CSPs, Amazon et Atos, représentant les deux groupes de CSPs que nous avons identifiés dans notre première évaluation. Les résultats que nous avons obtenus montrent que l'architecture Kumori améliore globalement le nombre de chemins disjoints empruntables entre deux PoPs d'un CSP. Une analyse plus détaillée souligne que les bénéfices associés à une utilisation de Kumori sont plus importants pour les paires de PoPs dont la diversité topologique est limitée. Cette observation souligne le potentiel de l'*overlay* Kumori pour les CSPs les plus petits dont les PoPs ne bénéficient typiquement pas d'une diversité topologique élevée. En outre, une comparaison du nombre de chemins disjoints obtenu pour différentes configurations de Kumori, en faisant varier le nombre de points d'inflexion de routage ou la politique de placement de ces nœuds, souligne que les performance de Kumori en termes de résilience dépendent moins du nombre de points d'inflexion de routage utilisés que de leur placement à des points d'échange Internet géographiquement diversifiés.

Enfin, nous avons évalué Kumori d'un point de vue économique. Nous avons évalué la structure de coût de l'*overlay* Kumori, et nous l'avons comparé au coût d'un réseau de connexions privées et redondées pour plusieurs topologies de réseau inter-datacenter. Cette étude montre que la viabilité économique de l'architecture Kumori dépend fortement du rapport entre le coût du trafic réseau de transit et le coût du trafic réseau pour une utilisation de liens à longue distance, attendu que les frais de connectivité constituent la majeure partie du coût de l'architecture Kumori. Suite à une évaluation numérique basée sur des données de marché collectées auprès d'un cabinet d'analyste, nous pouvons dire que l'architecture Kumori est économiquement intéressante en Amérique du Nord et en Asie. De plus, nous montrons également que la dynamique des prix des liens privés et du transit réseau fait que la rentabilité économique de l'architecture Kumori s'accroîtra à l'avenir, et permettra d'envisager un usage de notre architecture dans d'autres régions du monde.

8.2 Perspectives

Dans ce travail de thèse, nous avons concentré nos efforts sur la conception et sur l'évaluation des propriétés de Kumori en utilisant un graphe représentant Internet que nous avons élaboré pour mener à bien notre évaluation. Afin de poursuivre notre travail sur Kumori, nous avons l'intention de mettre en œuvre un prototype de cette architecture. Pour cela, nous utiliserons un contrôleur SDN pour implémenter le comportement de chaque type de nœud de Kumori. Ainsi, chaque nœud consistera en

une application lancée sur un contrôleur et exécutée sur un équipement ou un nœud SDN. A cet effet, nous pourrons par exemple utiliser le framework SDN Ryu [Ryu]).

Après ce travail de développement, nous avons l'intention de tester notre prototype sur un réseau émulé. Le but de ce test est d'évaluer notre capacité effective à rediriger le trafic réseau et à réagir rapidement en cas de détection d'une panne. Nous prévoyons de tester les différentes options de conception de l'architecture que nous avons étudiées et envisagé d'utiliser dans la section 3.3 de ce manuscrit. En effet, nous envisageons d'évaluer les mécanismes d'encapsulation du trafic réseau que nous pourrions utiliser pour contrôler le routage du trafic entre les nœuds de Kumori. Par ailleurs, nous avons l'intention d'évaluer la charge générée par les messages de remontées de mesures du trafic réseau envoyés par les nœuds de l'architecture Kumori au contrôleur centralisé. L'évaluation du temps de détection d'une panne à partir de ces mesures fera également l'objet d'une attention particulière. Au titre de ces évaluations, nous envisageons d'utiliser Mininet [Min] ou GNS3 [GNS] pour émuler le déploiement de l'*overlay* Kumori au sein d'un réseau entre différents datacenters d'un CSP. Dans ce test sur un environnement émulé, nous devrons modéliser le trafic circulant entre les différents datacenters gérés par le CSP. À notre connaissance, il n'existe pas de modèle détaillé pour un tel trafic réseau inter-datacenter dans la littérature. Aussi, la construction d'un tel modèle de trafic serait très utile pour notre évaluation, et au-delà pour la communauté de recherche travaillant sur l'optimisation du trafic réseau entre datacenters.

Dans l'architecture Kumori, nous supposons qu'il est possible pour un contrôleur centralisé de contrôler les nœuds de l'architecture malgré des distances assez longues sur Internet. Cette hypothèse est assez forte si l'on considère que, dans les réseaux SDN classiques, les contrôleurs sont utilisés pour gérer des équipements situés dans leur voisinage direct. Dans l'architecture Kumori, la latence et la gigue potentielles sur le lien entre le contrôleur et les points d'inflexion de routage peuvent avoir un impact important sur la capacité de l'architecture à réagir rapidement aux pannes ou aux défaillances qui surviendraient. Nous devons évaluer cet impact sur un démonstrateur déployé sur Internet. Ce démonstrateur pourra également être utilisé pour évaluer les politiques de filtrage des en-têtes et des options protocolaires utilisés dans les méthodes d'encapsulation que nous pourrions utiliser par les différents opérateurs. Ces prolongements de notre travail pourraient probablement justifier la réalisation d'un autre travail de thèse.

Dans la conception de notre *overlay*, nous faisons l'hypothèse que l'augmentation de la diversité des chemins IP entre les datacenters améliore la résilience des interconnexions entre datacenters. Néanmoins, des travaux récents tels que [CCGF14] mettent en évidence l'émergence massive de technologies dites de peering à distance. Ces technologies permettent à des ASs de se connecter à des IXPs en utilisant des

liens physiques à longue distance qui sont invisibles dans les données que nous avons utilisées pour établir notre représentation d'Internet. De ce fait, nous n'avons pas pu évaluer la diversité des chemins entre les PoPs des CSPs en prenant en compte ces liens de peering à longue distance. Selon [CCGF14], le peering à distance est utilisé par environ 20% des ASs membres de certains grands IXPs. Étant donné les caractéristiques des liens de peering à longue distance, ceux-ci peuvent avoir un impact significatif sur la diversité de chemin offerte sur Internet, et donc sur les résultats que nous avons obtenus concernant les bénéfices offerts par l'architecture Kumori en termes de résilience.

Enfin, nous pensons que le graphe Internet orienté de niveau de PoP que nous avons construit à l'occasion de cette thèse pour évaluer l'architecture Kumori pourrait être amélioré. En effet, dans notre travail, nous avons utilisé deux méthodes de "clustering" génériques. Or, nous savons que les connexions entre les routeurs constituant les points de présence des systèmes autonomes sur Internet suivent des modèles d'architecture spécifiques. Aussi, revoir notre méthode de "clustering" pour utiliser un algorithme apte à reconnaître ces modèles d'architecture pourrait améliorer la pertinence et l'exactitude du graphe représentant Internet que nous avons construit. Pour ce faire, nous pourrions tirer parti des progrès récents en matière d'analyse de données. Aujourd'hui, la plupart des modèles représentant Internet sont, à quelques exceptions notables, peu à jour, et ne permettent pas de constituer une image directement exploitable d'Internet. Nous chercherons donc à publier l'algorithme que nous développerons en parallèle des données que nous aurons synthétisées afin de permettre une reprise et une critique de nos travaux par la communauté scientifique du domaine.

Bibliography

[AAK14] Niels L. M. van Adrichem, Benjamin J. van Asten, and Fernando A. Kuipers. Fast recovery in software-defined networks. In *Proceedings of the 2014 Third European Workshop on Software Defined Networks*, EWSDN '14, pages 61–66, Washington, DC, USA, 2014. IEEE Computer Society.

[ABKM01] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. *SIGOPS Oper. Syst. Rev.*, 35(5):131–145, October 2001.

[ACF+12] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. Anatomy of a large european ixp. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, pages 163–174, New York, NY, USA, 2012. ACM.

[AKW09] Brice Augustin, Balachander Krishnamurthy, and Walter Willinger. Ixps: Mapped? In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pages 336–349, New York, NY, USA, 2009. ACM.

[AMS+03] Aditya Akella, Bruce Maggs, Srinivasan Seshan, Anees Shaikh, and Ramesh Sitaraman. A measurement-based analysis of multihoming. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 353–364, New York, NY, USA, 2003. ACM.

[AMZ] Global infrastructure - amazon web services. https://aws.amazon.com/fr/about-aws/global-infrastructure/. Accessed: 2015-08-20.

[AWS] Aws direct connect product details. https://aws.amazon.com/directconnect/details/. Accessed: 2016-05-02.

[BBC+11] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31 –39, 2011.

[BBM16] S. Beckert, B. Boudreau, and A. Mauldin. Telegeography workshop: International market trends. https://www.ptc.org/assets/uploads/papers/ptc16/, jan. 2016.

[Bry16] G. Bryan. Enterprise network pricing trends for wan benchmarks, 2016.

[CBAB14] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba. Payless: A low cost network monitoring framework for software defined networks. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–9, May 2014.

[CCGF14] Ignacio Castro, Juan Camilo Cardona, Sergey Gorinsky, and Pierre Francois. Remote peering: More peering without internet flattening. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pages 185–198, New York, NY, USA, 2014. ACM.

[CHW+11] Kai Chen, Chengchen Hu, Xitao Wen, Yan Chen, and Bin Liu. Toward internet emergency response via reconfiguration in internet exchange points reconfiguration in internet exchange points. Technical report, Northwestern University, Electrical Engineering and Computer Science Department, Aug 2011.

[CMM99] K. Claffy, T. Monk, and D. McRobb. Internet tomography. *Nature*, Jan 1999.

[DBEH+07] Giuseppe Di Battista, Thomas Erlebach, Alexander Hall, Maurizio Patrignani, Maurizio Pizzonia, and Thomas Schank. Computing the types of the relationships between autonomous systems. *IEEE/ACM Trans. Netw.*, 15(2):267–280, April 2007.

[DD10] Amogh Dhamdhere and Constantine Dovrolis. The internet is flat: Modeling the transition from a transit hierarchy to a peering mesh. In *Proceedings of the 6th International COnference*, Co-NEXT '10, pages 21:1–21:12, New York, NY, USA, 2010. ACM.

[DG16] B. Davie and J. Gross. A stateless transport tunneling protocol for network virtualization (stt). Internet-draft, IETF Secretariat, April 2016.

[DS14] Angel Felipe Diaz Sanchez. *Cloud brokering : nouveaux services de valeur ajoutée et politique de prix*. PhD thesis, 2014. Thèse de doctorat dirigée par Gagnaire, Maurice Informatique et réseaux Paris, ENST 2014.

[EHM+06] Thomas Erlebach, Alexander Hall, Linda Moonen, Alessandro Panconesi, Frits Spieksma, and Danica Vukadinović. Dependable systems. chapter Robustness of the Internet at the Topology and Routing Level, pages 260–274. Springer-Verlag, Berlin, Heidelberg, 2006.

[FBDa] Facebook altoona data center. https://www.facebook.com/AltoonaDataCenter/. Accessed: 2015-08-20.

[FBDb] Facebook clonee data center. https://www.facebook.com/CloneeDataCenter/. Accessed: 2015-08-20.

[FBDc] Facebook forest city data center. https://www.facebook.com/ForestCityDataCenter/. Accessed: 2015-08-20.

[FBDd] Facebook fort worth data center. https://www.facebook.com/FortWorthDataCenter/. Accessed: 2015-08-20.

[FBDe] Facebook lulea data center. https://www.facebook.com/LuleaDataCenter/. Accessed: 2015-08-20.

[FBDf] Facebook prineville data center. https://www.facebook.com/PrinevilleDataCenter/. Accessed: 2015-08-20.

[FBG+14] Y. Fu, J. Bi, K. Gao, Z. Chen, J. Wu, and B. Hao. Orion: A hybrid hierarchical control plane of software-defined networking for large-scale networks. In *2014 IEEE 22nd International Conference on Network Protocols*, pages 569–576, Oct 2014.

[FFML13] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. The Locator/ID Separation Protocol (LISP). RFC 6830 (Proposed Standard), January 2013.

[FG14] A. Fressancourt and M. Gagnaire. A dynamic offer/answer mechanism encompassing tcp variants in heterogeneous environments. In *Advanced Networking Distributed Systems and Applications (INDS), 2014 International Conference on*, pages 7–12, June 2014.

[FG15] A. Fressancourt and M. Gagnaire. A sdn-based network architecture for cloud resiliency. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 479–484, Jan 2015.

[FLH+00] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. Generic Routing Encapsulation (GRE). RFC 2784 (Proposed Standard), March 2000.

[FPG16] A. Fressancourt, C. Pelsser, and M. Gagnaire. Kumori: Steering cloud traffic at ixps to improve resiliency. In *2016 12th International Conference on the Design of Reliable Communication Networks (DRCN)*, pages 138–144, March 2016.

[FS08] D. Feldman and Y. Shavitt. Automatic large scale generation of internet pop level maps. In *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, pages 1–6, Nov 2008.

[FZRL08] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *2008 Grid Computing Environments Workshop*, pages 1–10, Nov 2008.

[GALM08] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. The flattening internet topology: Natural evolution, unsightly barnacles or contrived collapse? In *Proceedings of the 9th International Conference on Passive and Active Network Measurement*, PAM'08, pages 1–10, Berlin, Heidelberg, 2008. Springer-Verlag.

[Gao01] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. Netw.*, 9(6):733–745, December 2001.

[GGS16] J. Gross, I. Ganga, and T. Sridhar. Geneve: Generic network virtualization encapsulation. Internet-draft, IETF Secretariat, September 2016.

[GILO11] Enrico Gregori, Alessandro Improta, Luciano Lenzini, and Chiara Orsini. The impact of ixps on the as-level topology structure of the internet. *Comput. Commun.*, 34(1):68–82, January 2011.

[GMG+04] Krishna P. Gummadi, Harsha V. Madhyastha, Steven D. Gribble, Henry M. Levy, and David Wetherall. Improving the reliability of internet paths with one-hop source routing. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 13–13, Berkeley, CA, USA, 2004. USENIX Association.

[GNS] Graphical network simulator - gns3. http://www.gns3.net/. Retrieved Aug. 15, 2014.

[GR00] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. *SIGMETRICS Perform. Eval. Rev.*, 28(1):307–317, June 2000.

[GVS+14] Arpit Gupta, Laurent Vanbever, Muhammad Shahbaz, Sean P. Donovan, Brandon Schlinker, Nick Feamster, Jennifer Rexford, Scott Shenker, Russ Clark, and Ethan Katz-Bassett. Sdx: A software defined internet exchange. *SIGCOMM Comput. Commun. Rev.*, 44(4):551–562, August 2014.

[HCC+12] C. Hu, K. Chen, Y. Chen, B. Liu, and A. V. Vasilakos. A measurement study on potential inter-domain routing diversity. *IEEE Transactions on Network and Service Management*, 9(3):268–278, September 2012.

[HE-] Hurricane electric's bgp toolkit.

[HHL+09] Sing Wang Ho, Thom Haddow, Jonathan Ledlie, Moez Draief, and Peter Pietzuch. Deconstructing internet paths: An approach for as-level detour route discovery. In *Proceedings of the 8th International Conference on Peer-to-peer Systems*, IPTPS'09, pages 8–8, Berkeley, CA, USA, 2009. USENIX Association.

[HNR68] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[HWJ08] Junghee Han, David Watson, and Farnam Jahanian. Enhancing end-to-end availability and performance via topology-aware overlay networks. *Comput. Netw.*, 52(16):3029–3046, November 2008.

[Hyu06] Young Hyun. The archipelago measurement infrastructure, Nov 2006.

[HYZ16] T. Herbert, L. Yong, and O. Zia. Generic udp encapsulation. Internet-draft, IETF Secretariat, July 2016.

[igr] igraph the network analysis package.

[IPS82] A. Itai, Y. Perl, and Y. Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12(3):277–286, 1982.

[Jef14] T. Jeffree. 802.1Q-2014 - Bridges and Bridged Networks, August 2014.

[JKM+13] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. B4: Experience with a globally-deployed software defined wan. *SIGCOMM Comput. Commun. Rev.*, 43(4):3–14, August 2013.

[KA12] M. Kühne and V. Asturiano. Update on as path lengths over time. https://labs.ripe.net/Members/mirjam/update-on-as-path-lengths-over-time, sep. 2012.

[KAK+16] Rowan Klöti, Bernhard Ager, Vasileios Kotronis, George Nomikos, and Xenofontas Dimitropoulos. A comparative look into public ixp datasets. *SIGCOMM Comput. Commun. Rev.*, 46(1):21–29, January 2016.

[KKAD15] R. Klöti, V. Kotronis, B. Ager, and X. Dimitropoulos. Policy-compliant path diversity and bisection bandwidth. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 675–683, April 2015.

[KKR+15] Vasileios Kotronis, Rowan Klöti, Matthias Rost, Panagiotis Georgopoulos, Bernhard Ager, Stefan Schmid, and Xenofontas Dimitropoulos. Investigating the potential of the inter-ixp multigraph for the provisioning of

guaranteed end-to-end services. *SIGMETRICS Perform. Eval. Rev.*, 43(1):429–430, June 2015.

[KKR+16] Vasileios Kotronis, Rowan Klöti, Matthias Rost, Panagiotis Georgopoulos, Bernhard Ager, Stefan Schmid, and Xenofontas Dimitropoulos. Stitching inter-domain paths over ixps. In *Proceedings of the 12nd ACM SIGCOMM Symposium on Software Defined Networking Research*, SOSR '16, New York, NY, USA, 2016. ACM.

[KST+12] Hyojoon Kim, J.R. Santos, Y. Turner, M. Schlansker, J. Tourrilhes, and N. Feamster. Coronet: Fault tolerance for software defined networks. In *Network Protocols (ICNP), 2012 20th IEEE International Conference on*, pages 1–2, Oct 2012.

[KW10a] D. Katz and D. Ward. Bidirectional Forwarding Detection (BFD). RFC 5880 (Experimental), June 2010.

[KW10b] D. Katz and D. Ward. Bidirectional Forwarding Detection (BFD) for Multihop Paths. RFC 5883 (Experimental), June 2010.

[LLD+14] Aemen Lodhi, Natalie Larson, Amogh Dhamdhere, Constantine Dovrolis, and kc claffy. Using peeringdb to understand the peering ecosystem. *SIGCOMM Comput. Commun. Rev.*, 44(2):20–27, April 2014.

[LZQL07] Yi Li, Yin Zhang, Lili Qiu, and S. Lam. Smarttunnel: Achieving reliability in the internet. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 830–838, May 2007.

[MAB+08] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.

[Max] Maxmind geoip databases and services. https://www.maxmind.com/en/geoip2-services-and-databases. Retrieved Jul. 15, 2015.

[MDD+14] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright. Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. RFC 7348 (Proposed Standard), August 2014.

[Min] Mininet: an instant virtual network on your laptop (or other pc). http://mininet.org/. Retrieved Aug. 15, 2014.

[MIP+06] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iplane: An

information plane for distributed services. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 367–380, Berkeley, CA, USA, 2006. USENIX Association.

[Ope13] Openflow switch specification 1.4.0, October 2013.

[OPW+10] Ricardo Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. The (in)completeness of the observed internet as-level structure. *IEEE/ACM Trans. Netw.*, 18(1):109–122, February 2010.

[PBL14] K. Phemius, M. Bouet, and J. Leguay. Disco: Distributed multi-domain sdn controllers. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–4, May 2014.

[PCG+13] F. Paolucci, F. Cugini, A Giorgetti, N. Sambo, and P. Castoldi. A survey on the path computation element (pce) architecture. *Communications Surveys Tutorials, IEEE*, 15(4):1819–1841, Fourth 2013.

[Pee] Peeringdb. https://www.peeringdb.com/. Retrieved Jul. 15, 2015.

[PFF+16] S. Previdi, C. Filsfils, B. Field, I. Leung, J. Linkova, E. Aries, T. Kosugi, E. Vyncke, and D. Lebrun. Ipv6 segment routing header (srh). Internet-draft, IETF Secretariat, September 2016.

[PL05] Pascal Pons and Matthieu Latapy. *Computer and Information Sciences - ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26-28, 2005. Proceedings*, chapter Computing Communities in Large Networks Using Random Walks, pages 284–293. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[Pri14] M. Prince. The relative cost of bandwidth around the world. https://blog.cloudflare.com/the-relative-cost-of-bandwidth-around-the-world/, aug.
2014.

[QMM07] S. Y. Qiu, P. D. McDaniel, and F. Monrose. Toward valley-free inter-domain routing. In *2007 IEEE International Conference on Communications*, pages 2009–2016, June 2007.

[RB07] M. Rosvall and C. T. Bergstrom. Maps of information flow reveal community structure in complex networks. In *Proceedings of the National Academy of Sciences USA*, pages 1118–1123, 2007.

[RBL+09] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galan. The reservoir model and architecture for open

federated cloud computing. *IBM Journal of Research and Development*, 53(4):4:1–4:11, July 2009.

[RI07] Alex Raj and Oliver C. Ibe. A survey of ip and multiprotocol label switching fast reroute schemes. *Comput. Netw.*, 51(8):1882–1907, June 2007.

[RIP10] Ripe atlas. https://atlas.ripe.net, Oct 2010. Accessed: 2015-08-20.

[RJS14] Justin P. Rohrer, Abdul Jabbar, and James P. Sterbenz. Path diversification for future internet end-to-end resilience and survivability. *Telecommun. Syst.*, 56(1):49–67, May 2014.

[RS11] J. P. Rohrer and J. P. G. Sterbenz. Predicting topology survivability using path diversity. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, pages 1–7, Oct 2011.

[RWM+11] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush. 10 lessons from 10 years of measuring and modeling the internet's autonomous systems. *IEEE Journal on Selected Areas in Communications*, 29(9):1810–1821, October 2011.

[Ryu] Ryu sdn framework. http://osrg.github.io/ryu/. Retrieved Aug. 15, 2014.

[SAA+99] Stefan Savage, Thomas Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoff Voelker, and John Zahorjan. Detour: Informed internet routing and transport. *IEEE Micro*, 19(1):50–59, January 1999.

[SAH] J. Snijders and S. Abdel-Hafez. Ixp pricing august 2016. Accessed: 2016-08-21.

[SARK02] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 618–627 vol.2, 2002.

[SHK15] T. Stronge, J. Hjembo, and E. Kreifeldt. Telegeography workshop: International market trends. https://www.ptc.org/assets/uploads/papers/ptc15/, jan. 2015.

[SMW02] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring isp topologies with rocketfuel. *SIGCOMM Comput. Commun. Rev.*, 32(4):133–145, August 2002.

[SS05] Yuval Shavitt and Eran Shir. Dimes: Let the internet measure itself. *SIGCOMM Comput. Commun. Rev.*, 35(5):71–74, October 2005.

[SSC+11] D. Staessens, S. Sharma, D. Colle, M. Pickavet, and P. Demeester. Software defined networking: Meeting carrier grade requirements. In *Local Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on*, pages 1–6, Oct 2011.

[SVLR14] João Luís Sobrinho, Laurent Vanbever, Franck Le, and Jennifer Rexford. Distributed route aggregation on the global network. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pages 161–172, New York, NY, USA, 2014. ACM.

[TMSV03a] Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker. Characterizing and measuring path diversity of internet topologies. *SIGMETRICS Perform. Eval. Rev.*, 31(1):304–305, June 2003.

[TMSV03b] Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker. In search of path diversity in isp networks. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC '03, pages 313–318, New York, NY, USA, 2003. ACM.

[vADK14] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers. Opennetmon: Network monitoring in openflow software-defined networks. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–8, May 2014.

[Van] J. Vanian. Amazon details how it does networking in its data centers.

[WZMS07] Jian Wu, Ying Zhang, Z. Morley Mao, and Kang G. Shin. Internet routing resilience to failures: Analysis and implications. In *Proceedings of the 2007 ACM CoNEXT Conference*, CoNEXT '07, pages 25:1–25:12, New York, NY, USA, 2007. ACM.

[YH16] L. Yong and W. Hao. Tunnel stitching for network virtualization overlay. Internet-draft, IETF Secretariat, July 2016.

[YJG03] Andy B. Yoo, Morris A. Jette, and Mark Grondona. *Job Scheduling Strategies for Parallel Processing: 9th International Workshop, JSSPP 2003, Seattle, WA, USA, June 24, 2003. Revised Paper*, chapter SLURM: Simple Linux Utility for Resource Management, pages 44–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[ZDA07] Yong Zhu, C. Dovrolis, and M. Ammar. Combining multihoming with overlay routing (or, how to be a better isp without owning a network). In

INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, pages 839–847, May 2007.

[ZP09] Xian Zhang and C. Phillips. Network operator independent resilient overlay for mission critical applications (romca). In *Communications and Networking in China, 2009. ChinaCOM 2009. Fourth International Conference on*, pages 1–5, Aug 2009.

Conception et mise en œuvre d'overlays réseau dynamiques pour la résilience du Cloud : Vers une flexibilité et une résilience accrue du Cloud Computing

Antoine FRESSANCOURT

RÉSUMÉ : Dans cette thèse, nous proposons une architecture réseau superposée (ou *overlay*) appelée "Kumori" permettant de détecter et de réagir rapidement à des pannes de liens ou de noeuds sur Internet entre différents centres de données appartenant à un fournisseur de services Cloud (CSP). Cet *overlay* se compose de points d'inflexion de routage placés à différents points d'échange Internet (IXP). Cette architecture Kumori est supervisée par un contrôleur centralisé.

Une fois l'architecture Kumori définie, nous en évaluons les caractéristiques en termes de performance et de résilience dans un contexte le plus réaliste possible. À cette fin, nous comparons les chemins disjoints rendus accessibles par l'architecture Kumori et par l'architecture RON (Resilient Overlay Network) au moyen d'une représentation d'Internet sous forme d'un graphe orienté que nous avons construite à partir de trois jeux de données publics. Enfin, nous présentons une évaluation du coût de l'architecture Kumori afin d'en évaluer la viabilité économique.

MOTS-CLEFS: Résilience, Cloud Computing, Overlay, Réseau, Internet, Graphe

ABSTRACT: In this thesis, we propose Kumori, an overlay network designed to allow a quick detection of failures affecting a network path between two datacenters belonging to a Cloud Services Provider (CSP) and a redirection of this network traffic around the detected failures. This network overlay consists in a set of routing inflection points located at various Internet Exchange Points (IXP). The nodes belonging to the Kumori architecture are coordinated using a centralized controller.

After the definition of the Kumori architecture, we evaluate the properties of our architecture in terms of performance and resiliency in a realistic context. In that extend we compare Kumori to the Resilient Overlay Network (RON) using a directed graph representing the Internet at the PoP level that we built using three publicly available datasets. To conclude, we evaluated the economic aspects of the Kumori architecture to assess its profitability.

KEY-WORDS: Resiliency, Cloud Computing, Overlay, Network, Internet, Graph