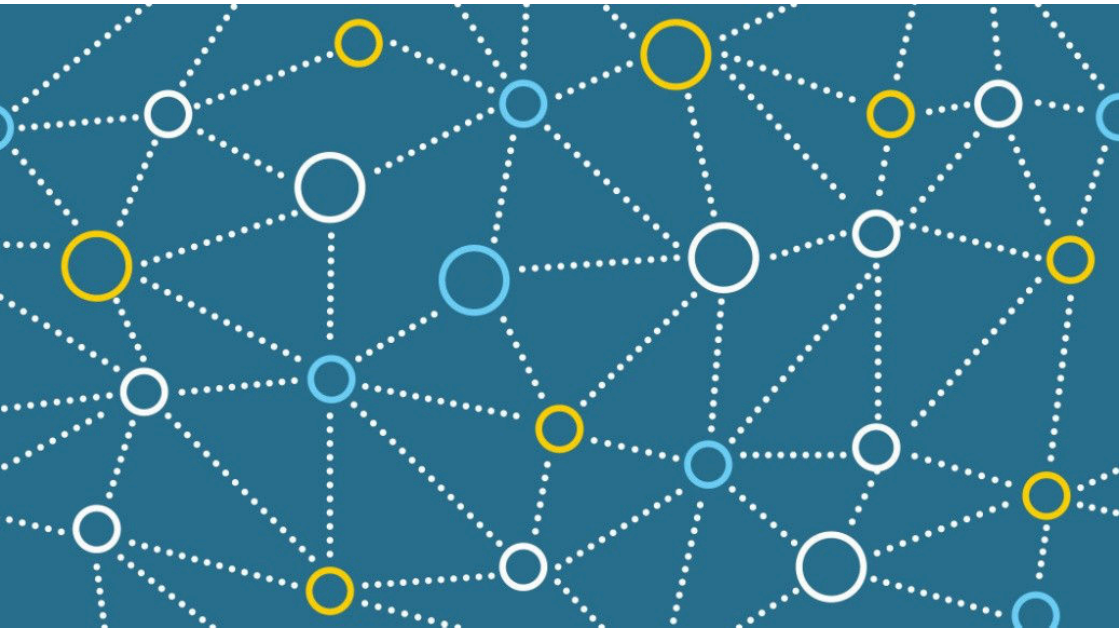
Centralizing Routing Control Across Domains: Architectural Approach and Prominent Use Cases

Vasileios Kotronis



Diss. ETH No. 22897

Diss. ETH No. 22897
TIK-Schriftenreihe Nr. 158

Centralizing Routing Control Across Domains: Architectural Approach and Prominent Use Cases

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

Vasileios Kotronis
Diplomate Engineer, Electrical and Computer Engineering
National Technical University of Athens, Greece
born on March 10, 1989
citizen of Greece

accepted on the recommendation of
Prof. Dr. Bernhard Plattner, examiner
Prof. Dr. Nick Feamster, co-examiner
Asst Prof. Dr. Xenofontas Dimitropoulos, co-examiner
Dr. Bernhard Ager, co-examiner

2015

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Dr. Bernhard Plattner, for giving me the opportunity to do my PhD in his group and his valuable help and generous support in fulfilling my goals as a researcher and making me feel at home at ETH.

I would also like to express my sincere gratitude to my advisors Asst Prof. Dr. Xenofontas Dimitropoulos and Dr. Bernhard Ager. Dr. Dimitropoulos supported me since day one of my academic career, and I would like to thank him for all the technical discussions, our efficient and fruitful cooperation and the exchanges of ideas and research endeavors during all those years. Dr. Bernhard Ager always helped me by providing a critical assessment of my achievements and made me go the extra mile in order to verify my claims and hypotheses. I would also like to thank him, together with Dr. Dimitropoulos, for his close collaboration on almost all of the publications on which this thesis was based. Each of my advisors had a unique and critical contribution to my success and I am grateful for that. I would also like to thank Prof. Dr. Nick Feamster, who agreed to be the third co-examiner in my doctoral examination committee, for reviewing this thesis.

Special thanks go to my close collaborators and people who helped me and encouraged me over the course of this PhD: Rowan Klöti, Dr. Panagiotis Georgopoulos, Dr. Stefan Schmid, Matthias Rost, Dr. Dominik Schatzmann, Dr. Paul Smith, Dimitrios Gkounis and Dr. Christos Liaskos. I am grateful to them for our exemplary academic cooperation.

The time I invested studying and working at CSG was composed of a very interesting set of experiences; in fact, I learned a lot about doing research, teaching and presenting my ideas. Many thanks go to

everyone who was with the CSG group for many interesting discussions and exchanges of ideas in the fabled coffee corner.

I am also grateful that I could work on the European Union research project OFELIA (FP7-ICT-258365), which enabled me to collaborate with other academic and industrial partners within a large-scale federation environment, ignited my interest in SDN and funded part of my PhD research during my first two years as a candidate. Many thanks go to everyone who was involved in OFELIA for their cooperation on administering the testbed and developing the OCF software.

I would like to thank the following students who did their projects in the context of Software Defined Networking and/or Internet Peering (in chronological order): Jochen Mattes, Rowan Klöti, Adrian Friedli, Adrian Gämperli, Dimitrios Gkounis, Demian Jäger, Remo Balaguer and Damian Scherrer. It was my honor being one of their advisors. Special thanks go to Rowan Klöti and Adrian Gämperli for their cooperation and valuable assistance in the implementation of many of the concepts that are described in this thesis.

I am also grateful for the great administrative and technical support I got from the following people: Beat Futterknecht, Thomas Steingruber, Friederike Brütsch, Benny Gächter and Tanja Lantz. Special thanks go to Beat for exceeding the boundaries of his role to support me during my years at ETH by providing valuable help and information.

Last but not least, I would like to send my warmest thanks to my family, namely my father Panagiotis, my mother Eirini and my brother Christos. They supported me throughout my entire academic career and helped me stay the course even when I was uncertain of my final goal; as most PhD candidates know, research is usually an abstract set of objectives that become clear only after a lot of manual effort, personal toil and experimentation. My family's support was an extremely important factor that shaped my personality during this process. Their unconditional love and affection always inspire me to see the best in people and give me the strength to pursue my goals to the very end. Thank you for always believing in me!

Abstract

The coordination and management of inter-domain routing in the Internet are crucial tasks that happen "behind the scenes", while we use the Internet for checking our emails, surfing the web, making Skype video calls, watching NetFlix episodes or engaging in social networking. The basic protocol that governs inter-domain routing, i.e., routing between the diverse administrative network domains, is the Border Gateway Protocol (BGP). Researchers have pinpointed several issues with the design and operation of BGP over the years, such as slow convergence times or lack of end-to-end service guarantees, and have proposed various solutions and improvements. Despite this large body of proposals, BGP is extremely resistant to change, albeit the only core protocol that virtually every Internet Service Provider (ISP) uses today to route packets between domains. Therefore, we need to find ways to address the requirements of new Internet services and evolve routing while maintaining compatibility with BGP. We thus propose a new architectural approach which can be incrementally deployed in today's Internet and which can gradually tackle BGP-related challenges while its adoption gains momentum. Within the scope of this work, we address the challenges related to the suboptimal convergence behavior, suboptimal path computations and lack of inter-domain service guarantees. The approach is general enough to be used for optimizing further aspects of inter-domain routing beyond the scope of this work.

As our basic approach, we make the case for gradual routing centralization across domains based on Software Defined Networking (SDN) principles and outsourcing mechanisms. The envisioned setting is composed of contractors offering Routing as a Service (RaaS), serving—contiguous or disjoint—clusters (i.e., groups) of Autonomous Systems (AS), which are their clients. In general, contractors may operate on abstract views of their client networks. We show how the proposed architecture may be used as the basis for such improvements, by investigating two concrete and prominent use cases.

As a first use case, the RaaS contractors talk to external ASes via BGP, while providing inter-domain optimizations and new routing services within

their respective client clusters via SDN mechanisms. These optimizations, such as more stable routing and better convergence, may also benefit non-client ASes. In fact, we show that routing convergence improves w.r.t. time and load of routing updates, as the sphere of influence of RaaS contractors expands over hybrid BGP-SDN multi-domain networks. We have thus developed an emulation framework to run hybrid routing experiments.

As a second use case, we investigate RaaS contractors that work independently from BGP by performing overlay stitching of partial paths crossing ISP domains. The goal of this process is to form end-to-end routes with certain properties, such as guaranteed bandwidth, latency or availability. The stitching can be implemented, e.g., using well placed and connected programmable switching points at ISP interconnects. The setup on which we simulate our proposal is the rich overlay fabric that ISPs and their peering interconnections compose, utilized in a novel way for the provisioning of guaranteed end-to-end services across domains. This setup is aligned to the status quo in the ISP ecosystem, in terms of inter-domain topologies and peering relationships. In fact, our research is based on Internet eXchange Points (IXPs); these are physical infrastructures where multiple ISPs exchange data with each other.

To investigate the potential of the aforementioned peering fabric in terms of AS-level path diversity and associated metrics, we further develop a graph transformation algorithm. This algorithm can be used for the computation of min-cuts on arbitrary network graphs governed by policies. One of the application contexts of this computation is an inter-domain setting governed, e.g., by valley-free routing policies. Metrics that rely on policy-compliant min-cuts can be calculated centrally by a multi-domain RaaS contractor to evaluate routing availability, multipath throughput and reliability against link failures.

We investigate the benefits, implications and outlook of the general approach and its associated use cases, supporting the following hypothesis: *Inter-domain routing centralization, performed in a staged and consistent manner, can help to evolve and improve Internet routing on the AS level.* In the context of the studied use cases, the routing algorithms, protocols and services are evolving on top of the RaaS contractor's platform. This platform is the logically centralized routing control plane which operates on an abstract global view of the (inter-)network.

Kurzfassung

Die Koordination und die Verwaltung des Inter-Domain-Routings sind unverzichtbare Aufgaben im heutigen Internet, welche weitgehend unbemerkt "hinter den Kulissen" stattfinden, während wir das Internet nutzen um E-Mails zu lesen, auf Webseiten zu surfen, zu skypen, NetFlix-Serien anzuschauen oder Zeit in sozialen Netzwerken zu verbringen. Das Border-Gateway-Protocol (BGP) ist das Standardprotokoll, welches das Routing zwischen verschiedenartigen administrativen Netzwerkdomänen ermöglicht. Zahlreiche Forscher haben in den letzten Jahren verschiedene Schwachstellen der Gestaltung und Implementierung des BGP-Protokolls aufgezeigt, wie zum Beispiel hohe Konvergenzzeiten oder das Fehlen von Ende-to-Ende Service Garantien. In der Folge haben sie verschiedene Änderungen am BGP-Protokoll vorgeschlagen, welche die aufgezeigten Probleme beheben können. Trotz der zahlreichen Änderungsvorschläge ist das BGP-Protokoll sehr schwer zu ändern. Es ist zwar das einzige Protokoll, das heutzutage von praktisch jedem Internet-Service-Provider (ISP) unterstützt wird, um Pakete zwischen verschiedenen Domänen zu leiten. Daher benötigen wir einen neuartigen Ansatz für die Entwicklung des Inter-Domain-Routings, welcher auf der einen Seite die gestiegenen Anforderungen heutiger Internet-Dienste berücksichtigt, und auf der anderen Seite kompatibel zum BGP-Protokoll ist. Deshalb schlagen wir einen neuartigen, inkrementellen Ansatz vor, welcher schrittweise im heutigen Internet eingesetzt werden kann, um die erwähnten Herausforderungen nach und nach zu bewältigen. Im Rahmen dieser Arbeit befassen wir uns mit den Problemen, die aufgrund des suboptimalen Konvergenzverhaltens, der suboptimalen Pfadberechnungen und des Fehlens von Dienstgütegarantien im Betrieb des Inter-Domain-Routings zu lösen sind. Unser generischer Ansatz kann auch angewendet werden, um weitere Aspekte des Inter-Domain-Routings zu optimieren, die in dieser Arbeit nicht behandelt werden. Mit unserem Ansatz werben wir für eine stufenweise Zentralisierung des Inter-Domain-Routings, basierend auf den Prinzipien des Software-Defined-Networkings (SDN) und mittels Outsourcing-Mechanismen. In dem von uns betrachteten Szenario bieten Dienstleister Routing-as-a-Service (RaaS) an, welche von zusammenhängenden oder disjunkten Gruppen von Autonomen Systemen (AS) erworben werden können. Die Dienstleister (RaaS-Provider) operieren dabei mit einer abstrakten Sicht auf ihre Kunden-Netzwerke. In dieser Doktorarbeit untersuchen wir unseren Ansatz in zwei konkreten und prominenten Anwendungsfällen.

In unserem ersten Anwendungsfall kommunizieren die RaaS-Provider mit den AS ihrer Kunden über SDN-Mechanismen und mit externen AS über das BPG-Protokoll. Sie bieten dabei ihren Kunden sowohl Optimierungen des Inter-Domain-Routings als auch neue Routing-Dienstleistungen innerhalb der Kundengruppen an. Hierbei können auch AS, die keine Kunden sind, von den angebotenen Optimierungen, wie ein stabileres Routing oder ein besseres Konvergenzverhalten, profitieren. In der Tat zeigen wir, dass sich das Konvergenzverhalten des Routings in Bezug auf die Zeit und die Anzahl an Routing-Updates verbessert, wenn sich der Einflussbereich der RaaS-Provider über hybride BGP-SDN Multi-Domain Netzwerke ausweitet. Dazu haben wir ein Emulations-Framework entwickelt, mit dem wir hybride Routing-Experimente durchführen können.

In unserem zweiten Anwendungsfall betrachten wir RaaS-Provider, die virtuelle Routen—sog. Pathlets—die sich über verschiedene ISP-Domänen hinweg erstrecken, zusammenknüpfen, ohne dabei das BGP-Protokoll zu verwenden. Hierbei ist es das Ziel, Ende-to-Ende-Verbindungen aufzubauen, die bestimmte Eigenschaften erfüllen, wie zum Beispiel eine garantierte Übertragungskapazität, Latenz oder Verfügbarkeit. Pathlets verschiedener Domänen können über programmierbare Vermittlungsstellen miteinander verknüpft werden. Diese Vermittlungsstellen sind so ausgewählt, dass sie als leistungsfähige Schnittstellen zwischen verschiedenen ISPs dienen können. In Frage dafür kommen vor allem sog. Internet eXchange Points (IXP); dies sind physische Infrastrukturen, an welchen verschiedene ISPs untereinander Daten austauschen. Dabei wird das virtuelle Netzwerk so aufgebaut, dass Ende-zu-Ende Dienste mit garantierter Dienstgüte zwischen verschiedenen Domänen angeboten werden können. Unsere Simulationsumgebung berücksichtigt den gegenwärtigen Stand des ISP-Ökosystems, in Bezug auf die Inter-Domain-Topologien und Peering-Beziehungen.

Darüber hinaus haben wir einen Algorithmus für die Transformation der Inter-Domain-Graphen entwickelt, so dass wir das erwähnte virtuelle Netzwerk in Bezug auf die Diversität von Pfaden zwischen Autonomen Systemen sowie weitere assozierte Metriken untersuchen können. Dieser Algorithmus kann dazu verwendet werden, minimale Schnitte in beliebigen Netzwerkgraphen zu berechnen. Diese Netzwerkgraphe werden durch Richtlinien (Policies) gesteuert, wie zum Beispiel eine Valley-free Routing-Policy bei einem Inter-Domain-Setup. Metriken, welche auf Policy-konforme, minimale Schnitte angewiesen sind, können von einem Multi-Domain RaaS-Provider zentral berechnet werden. Hierbei ist es das Ziel, das Routing für einzelne Attribute wie die Verfügbarkeit, den Multi-Path-Durchsatz und die Ausfallsicherheit gegenüber Verbindungsfehler zu bewerten.

Wir untersuchen den Nutzen, die Auswirkungen und den Ausblick unseres generischen Ansatzes und der assozierten Anwendungsfälle, um damit die folgende Hypothese zu stützen: *Eine stufenweise und konsistente Zentralisierung des Inter-Domain-Routings kann das Routing zwischen Autonomen Systemen verbessern und weiterentwickeln.* In den von uns untersuchten Anwendungsbeispielen werden die Routing-Algorithmen, die Routing-Protokolle und die Routing-Services auf der Plattform jedes RaaS-Providers weiterentwickelt. Diese Plattform stellt eine logisch zentralisierte Steuerung für das Routing dar und verwendet eine abstrakte, globale Sicht auf die beteiligten Netzwerke und die dazugehörigen netzwerkübergreifenden Links.

Contents

List of Figures

List of Tables

Abbreviations

AS Autonomous Systems.

ASN Autonomous System Number.

BGP Border Gateway Protocol.

CRWI Cluster Waiting Recomputation Interval.

CXPs Control eXchange Points.

eBGP external BGP.

FIB Forwarding Information Base.

iBGP internal BGP.

IGP Interior Gateway Protocol.

ISP Internet Service Provider.

IXPs Internet eXchange Points.

MED Multi-Exit Discriminator.

NaaS Network as a Service.

NFV Network Function Virtualization.

NOC Network Operations Center.

NOS Network Operating Systems.

PCE Path Computation Element.

PoPs Points of Presence.

QMRP QoS Multigraph Routing Problem.

QoS Quality of Service.

RaaS Routing as a Service.

RIB Routing Information Base.

SDN Software Defined Networking.

SDX Software Defined Internet eXchange.

SLA Service Level Agreement.

TE Traffic Engineering.

Chapter 1

Introduction

> "Begin at the beginning," the King said, very gravely,
> "and go on till you come to the end: then stop."
>
> —Lewis Carroll, *Alice in Wonderland*

We start with the motivation behind this dissertation and the scope of its creation in Section 1.1. We continue with the research questions that we address with our work and the basic hypothesis that we investigate in Section 1.2. Our contributions in the context of this investigation are summarized in Section 1.3. Lastly, we describe the structure of this thesis and the relationship between our publications and the corresponding chapters in Section 1.4.

1.1 Motivation and Scope

Our work revolves around two primary axes: *(i)* the need for evolvable inter-domain routing, and *(ii)* the advent of SDN principles, such as the gradual logical centralization of the routing control plane, which may enable such an evolution. We next elaborate on these axes.

1.1.1 The Need for Evolvable Inter-Domain Routing

Inter-domain routing is based on a fully decentralized model, where multiple ASes[1] interact and exchange routing information via BGP [212]. Although BGP has significantly contributed to the impressive scale and success of the Internet, being the de facto standard of core inter-domain routing for almost three decades, it is nonetheless one of the most inflexible and ossified protocols of the routing protocol stack[2]. While it is by design capable of exchanging prefix reachability information and expressing multifarious policies on the inter-AS level, it has several limitations [260] relating to its fully distributed[3] nature [159, 160, 196], routing policy enforcement capabilities and implications [47, 86, 102, 184, 237], security [25, 140] and management complexity [30, 31, 235]. For example, the BGP control plane can take several minutes to converge after a routing change [160]; during this period instabilities and anomalies may occur as ASes explore paths [196] or try to rate-limit the update churn [128, 244], affecting the routing availability for network applications [159, 179]. Furthermore, BGP was not designed to provide reliable end-to-end circuits with guaranteed properties [261]; mission-critical services which depend on these properties [120] may not run efficiently over a BGP core.

Despite the many research proposals for incremental improvements and clean-slate redesigns of how inter-domain routing should work, BGP has not successfully retrofitted the proposed ideas. One can not simply change BGP at one go. Therefore, we need to find an avenue towards evolvable inter-domain routing, while being compatible with BGP and generally with the status quo in the Internet ecosystem (e.g., market practices). At this point, we should note that the challenges related to

[1] The total number of ASes in the Internet is estimated between 40,000 and 50,000 ASes [49], with around 80% being edge networks and 20% core networks.

[2] and probably the entire TCP/IP core.

[3] Full distribution is simultaneously a strength and weakness of BGP. On the one hand, it is responsible for some aspects of routing scalability and separation of concerns regarding network governance, but on the other hand it adds to complexity, convergence times and difficulties with problem troubleshooting. For instance, a malfunction or misconfiguration of a single BGP router can uncontrollably propagate to the entire Internet and persist for hours before the problem can be effectively pointed out and resolved [176], e.g., by proper filtering in the control plane. Such errors can be either intentional [40] or simply the result of a human mistake [69].

changing and evolving the inter-domain routing system are clearly not only of technical nature, but also of economic/political nature: the task of convincing stakeholders to adopt changes requires financial incentives. Among these incentives we pinpoint the undisrupted use of the current network services, even if the currently deployed network base (such as routers, switches, etc.) is to be altered or extended gradually. The promise of profit in exchange for new services is also important. We further need to take into account that Internet stakeholders, such as transit ISPs, have already made large investments in acquiring and maintaining this legacy equipment; reducing their Operating Expenses (OPEX) while keeping this equipment provides strong motivation. On the one hand, clean slate approaches are suitable for theoretical reasoning about benefits and implications introduced by a novel architecture. On the other hand, a practical solution should in most usual cases preserve interoperability with a variety of legacy equipment while it enters the Internet landscape. For instance, "talking" BGP is an important requirement (cf. Chapter 4), which can though be sidestepped with the proper use of overlay networks and network virtualization (cf. Chapter 6).

1.1.2 SDN and Routing Centralization

SDN is an emerging network architecture which aims to change the current paradigm of networking, based on abstractions of the data, control and management planes. SDN aims at addressing the following challenge: making the data plane more programmable, while allowing the control plane to evolve in an independent fashion. The complexity of network control is meant to be "moved to the right place", i.e., to be streamlined and abstracted away at the appropriate layers of the control plane. For example, the state distribution challenge can be solved once and managed by a central point of control, in contrast to being replicated for every new protocol that an operator wants to add to her network, e.g., for hosting new services over the infrastructure.

The key concept of SDN is the separation of the network control plane from the data plane, e.g., separating routing from routers [85] making these devices simple forwarding elements. SDN enables logically centralized Network Operating Systems (NOS) and controllers [45, 114, 146]. A NOS implements the state distribution abstraction of the layered SDN model [222] and interacts with packet forwarding elements

based on forwarding abstractions such as OpenFlow [183], i.e., the *southbound* interface. Control features and applications, including routing algorithms, can be deployed on top of the NOS and run as software modules using a *specification* abstraction API, i.e., the *northbound* interface. The NOS presents a consistent network-wide view to the centralized control logic running on top of it. Multiple NOS systems and network control and management applications can run over the same network substrate, using network hypervisors implementing the virtualization layer of the SDN control stack [11, 224]. The following question naturally arises in this context: *Can we take advantage of SDN concepts, such as logical centralization and abstractions, on an inter-domain level?* The main possibility that we want to investigate is the use of logically centralized routing controllers on the inter-domain level, acting as the "thin waist" on a layered model of the Internet control plane. Network control applications such as end-to-end service provisioning may flourish above this waist while the data plane of the networks and the low-level control may evolve independently below the waist. We visualize the high-level concept in Figure 1.1.

1.1.3 The Scope of This Thesis

In this thesis, we leverage SDN to improve inter-domain routing properties while enabling innovation in routing applications running across multiple domains. This can be achieved by gradually forming logically centralized inter-domain routing controllers and AS clusters which are served by these controllers. We note that inter-domain centralization is an unconventional idea, which we though investigate with an open mind in terms of potential benefits. As a financial and technical means towards inter-domain centralization, we propose to outsource the routing control plane of an AS to external trusted providers, i.e., "Routing-as-a-Service" [162] contractors, according to our previous work [148, 149]. The high-level incentives that drive this process and govern the trust-based relationship are the opportunities for the reduction of the business expenses and for the provision of new services that are hard to offer without the presence of a contractor acting as a mediator, as we show later in the thesis. The contractor specializes in routing management and can relieve the ASes of the burden of maintaining expensive, highly-trained staff who manage the cumbersome routing complexity [30, 31]. Since a contractor manages

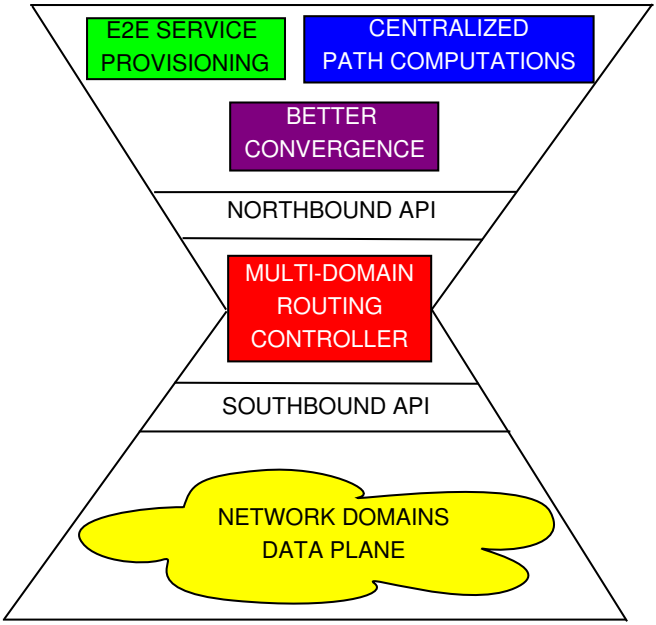


Figure 1.1: The promise of logical SDN-based routing centralization on the multi-domain level: routing controllers would be the "thin waist" of the control plane, with control and management applications and the data plane evolving independently north and south of them, respectively. The three multi-domain applications presented here correspond to the chapters 4, 5 and 6 of this thesis.

routing for several ASes, it can take advantage of this multi-AS level of logical centralization and aggregation in order to improve multiple aspects of inter-domain routing, while maintaining legacy compatibility with non-client ASes. We note that each AS preserves its policy-shaping capability, privacy and business identity; the contractor can for example operate only on a virtual slice [224] of the client network, managing inter-domain interactions. Outsourcing is only a means to an end; there can be alternative paths to inter-domain centralized control e.g., based on ISP coalitions occurring at IXPs, mediated via the IXP's SDN controller [117].

Based on our architectural approach using multi-domain Routing-as-a-Service controllers, we examine potential use cases where we can improve on current inter-domain routing, along with other associated benefits and challenges. The objective is to present and analyze the architectural approach, and go in depth in two instantiations of this approach applied on potential use cases. Within the scope of this thesis, we attack the challenges of BGP convergence, optimal centralized path computations and support of end-to-end service guarantees using the appropriate emulation/simulation tools. We do not claim to implement the whole architecture, but rather take a close look at two vertical slices, i.e., applications and their underlying—properly abstracted—infrastructure. For each of the slices/applications we use a suitable validation approach (first emulation, then simulation), combined with algorithmic frameworks where required. We would like to stress that we followed a use case-based approach instead of a full engineered implementation of the architecture for the following reasons. *(i)* Our main goal was to show that our idea makes sense before we proceed to a full implementation. *(ii)* There are large groups, both in the research and industrial communities, that work heavily on the engineering challenges of scaling up and securing such SDN controllers; due to lack of such resources on our side, both in terms of people and time, we focused more on the promise of the idea and the verification of the benefits of its potential applications. *(iii)* We believe that using this thesis as a basis, future researchers could combine the work of heavy-weight controllers implemented and deployed by the industry and the publicly available tools that we produced for investigating our use cases, and deploy these ideas within a temporal horizon of a few years on the real Internet.

Finally, we note that the proposed multi-domain RaaS controllers may coordinate using classic BGP or with new protocols, based for example on the notion of a "flow-space" instead of plain IP prefix-based routing. We do not investigate those in this thesis, since they do not belong to the primary scope of our research. We propose though how support for multiple SDN controllers and corresponding clusters communicating via BGP could be added to our emulation framework (cf. Chapter 4) in future work (cf. Section 7.4).

1.2 Research Questions and Basic Hypothesis

In this work, we address the following research questions that lie within the scope of the thesis (cf. Section 1.1).

- What new possibilities does inter-domain routing centralization enable? Which are the associated challenges and implications? What are the gains from such an approach in terms of performance, simplification of management or new opportunities such as on-the-fly provisioning of routing paths?
- Could the radical proposal of routing outsourcing across domains be realized in some form? Which mechanisms would be suitable for this realization?
- Which are the main entities and components of a potential routing outsourcing framework, operating on the inter-domain level, and how do they interact with each other?
- How would a SDN-based centralization scheme interact with classic BGP in terms of routing convergence? What is the impact of incremental deployment in terms of convergence times and load of routing updates in this context? Can the scheme solve some of the problems that BGP is currently facing?
- How could the routing centralization approach be applied in the context of guaranteed end-to-end services over the current Internet, working in parallel with BGP? How could staged upgrades in well-targeted parts of the supporting infrastructure be applied?
- A centralized routing platform may implement multifarious routing policies on the inter-domain level. Metrics such as path diversity or policy-compliant path lengths and bandwidth are affected by such policies. What is the effect of routing policy on path diversity on the AS level?

We will try to tackle the challenges associated with the aforementioned questions both from a system's perspective (multi-domain routing controllers and emulation framework) and a more theoretical perspective

(routing algorithms, path diversity calculations and simulations). In general, we will investigate the validity of the following hypothesis: *Inter-domain routing centralization, performed in a staged and consistent manner, can help to evolve and improve Internet routing on the AS level.* Additional questions, such as what are the characteristics of the economics of such an approach, will not be answered in this thesis but will be addressed in follow-up projects, such as Netvolution [189].

1.3 Contributions

With this dissertation we propose and look into a novel architectural approach for Internet routing, where parts of the inter-domain routing control plane are logically centralized. The aim of this proposal is to address some of the problems that BGP-based routing faces, such as slow convergence or the lack of end-to-end guarantees across the Internet. This is performed in a legacy-compatible way that is aligned with the status quo in the ISP market in particular, and the Internet ecosystem in general. The detailed contributions are listed below in chronological order. The associated publications and a brief recap of these contributions can be found in Section 7.2.

Proposal of a New Architecture. We describe a new, incrementally deployable architectural approach for Internet routing in which the routing control plane of multiple networks/domains is logically centralized. This follows the SDN paradigm based on the decoupling of the network control from the data plane and the consequent centralization of control, although at the inter-domain level involving multiple ASes. We do not interpret SDN in the narrow light of OpenFlow; control/data decoupling and control plane centralization may also be realized with legacy mechanisms, as examples of routing control platforms using internal BGP (iBGP) or any Interior Gateway Protocol (IGP) have shown [45, 245]. We see our work as an alternative path towards extending the value proposition of SDN over multiple domains, in parallel with research efforts on the Software Defined Internet eXchange (SDX) [117] or proposals for BGP-like communication between per-domain SDN controllers or instances of a NOS [29]. The proposed approach of SDN-based routing centralization across domains promises to become a vehicle for evolving and improving BGP, leveraging the bird's eye view over multiple networks to benefit several

aspects of inter-domain routing. These aspects may range from faster, less "chatty" convergence and policy conflict resolution, to end-to-end Quality of Service (QoS) and collaborative security and troubleshooting applications on the AS level.

How to Realize our Proposal in Today's Internet. Multi-domain SDN-based routing centralization can be realized by outsourcing routing functions to an external Routing-as-a-Service contractor, which provides inter-domain services facilitated through a multi-AS network controller or NOS. We note that the controller is a logically centralized construct that may be physically implemented using distributed systems mechanisms for resiliency and scale. In the outsourcing context, each AS preserves its policy-shaping capability, privacy and business identity; the contractor can for example operate only on a virtual slice of the client network, managing inter-domain interactions. As mentioned before, there can be alternative ways towards outsourced centralized control across domains.

Use Case 1: Better Convergence. With this use case we investigate the improvements induced in routing convergence, as more and more ASes perform cumulative outsourcing of (parts of) their routing control plane using a common RaaS contractor. We therefore introduce a publicly available emulation platform built on top of popular SDN emulation and legacy BGP routing software, for experimenting on hybrid BGP-SDN AS-level networks. We focus on external BGP (eBGP) sessions between the AS border routers. The framework capitalizes on the "big router" abstraction for the ASes, due to lack of internal topology and policy data. This abstraction is not fundamental, but it allows us to capture some basic properties of hybrid inter-domain routing, even in the absence of complete information per domain. Since we focus on exploiting multi-domain centralization to improve BGP's slow convergence in the context of routing outsourcing, we build and make publicly available a first multi-AS controller tailored to this use case. The controller speaks OpenFlow within the served AS clusters, and BGP with the rest of the Internet. Using the controller and the emulation framework, we demonstrate experimentally that SDN centralization helps to reduce BGP convergence times and churn rates with expanding SDN deployments. We note that while BGP is not used for routing within the clusters, it is still needed for interoperability with the rest of the Internet; we cannot—and probably do not want to—centralize all

routing decisions for the whole Internet. Thus we evaluate the benefits of gradual centralization of some of its parts. Further insights regarding the interplay between path-vector BGP and link-state SDN routing are analyzed, along with the associated challenges and trade-offs, such as controller scalability, state consistency and northbound interfaces. The latter challenges are dealt with using the abstractions offered by modern SDN controller frameworks.

Min-cuts on Graphs under Network Policies. During the course of our research, the issue of routing policies and their impact on path diversity on AS-level topologies came into play. The reason is that environments such as the AS-level Internet are governed by routing policies [47]. We thus needed to address the following questions. How many links can be cut before a network is bisected? How diverse or "rich" is a network path-wise? What is the maximal bandwidth that can be pushed between two nodes of a network? These questions are closely related to network resilience, path choice for multipath routing or bisection bandwidth estimations in multiple network environments. The answer is quantified using metrics such as the number of edge-disjoint paths between two network nodes and the cumulative bandwidth that can flow over these paths. In practice though, such calculations are far from simple due to the restrictive effect of network policies on path selection. Policies are set by network administrators to conform to service level agreements, protect valuable resources or optimize network performance. We thus introduce a general methodology for estimating lower and upper bounds for the policy-compliant path diversity and bisection bandwidth between two nodes of a network, effectively quantifying the effect of policies on these metrics. Exact values can be obtained if certain conditions hold. The approach is based on regular languages and can be applied in a variety of use cases. In the multi-AS RaaS context, the algorithm may run on top of a logically centralized routing controller. In fact, in such a setup, calculating the policy-conformant min-cuts is a complex task while the policies themselves restrict the corresponding values. Moreover, the RaaS controller operates on the global abstract view of the underlying network graph where policies from multiple domains are applied, governing their interactions. A class of such policies is the valley-free regime [98] (cf. Section 2.1.2), which we study in practice, together with some of its relaxations (cf. Chapter 5, Chapter 6).

Use Case 2: End-to-End Guaranteed Services, Across Domains. With this use case we investigate Internet QoS pathlet (i.e., partial path) brokers, which we treat as a family of centralized inter-domain QoS enablers, called Control eXchange Points (CXPs). CXPs are routing control plane intermediaries which work independently from BGP and stitch paths end-to-end while maintaining QoS on an inter-domain level. These paths are offered with specific properties and guarantees by ISPs, connecting their domain edges. ISPs exchange control with CXPs along the end-to-end path in order to benefit from the service (e.g., bandwidth- or latency-guaranteed VPN tunnel) running on top. In the context of the rising interest in IXPs and the enormous peering ecosystem that ISPs, Content Providers and IXPs compose, we research IXP deployments as potential points at which ISPs and CXPs can meet and operate. Looking beyond the system aspects, based on a novel view of the overlay graph formed using inter-IXP paths which cross ISP domains under the supervision of CXPs, we perform the following steps. First, we process and compare public peering datasets and consequently we form and measure inter-IXP overlay graphs on which CXPs may operate. We show that these graphs are rich enough to support the CXP use case in terms of IP address coverage and path diversity, and that incremental deployment is viable. Their potential is investigated both in *(i)* time, as the peering ecosystem evolves, and *(ii)* space, in terms of incremental deployment on multiple geographically dispersed IXPs. Policy-compliant path diversity extending beyond valley-free policies, within an environment where every ISP has incentives to offer pathlets, is taken into consideration for the calculations using our previous methodology. Furthermore, we introduce and evaluate an algorithmic framework that addresses CXP pathlet stitching over dense IXP-based overlay graphs. Simulations of QoS-constrained paths over such graphs are performed using variants of online, offline and hybrid algorithms and real datasets providing IXP facility and membership information, such as Euro-IX and PeeringDB.

To summarize, in this thesis we make the case for gradual routing centralization across domains based on SDN principles and outsourcing mechanisms. The envisioned setting is composed of multiple Routing-as-a-Service contractors serving contiguous or disjoint clusters of ASes. These contractors can talk to legacy ASes with BGP, while providing inter-domain optimizations and new services within the clusters. These optimizations, such as more stable routing and improved convergence,

may also benefit ASes outside of the cluster. In the CXP context, such contractors may work independently from BGP performing overlay pathlet stitching for pathlets crossing ISP domains, while being anchored to IXPs. The benefits, challenges and outlook of the general paradigm are investigated, validating the basic hypothesis of this thesis (cf. Section 1.2). We do not claim that this thesis is an exhaustive analysis of all potential inter-domain routing centralization schemes and potential architectures; we rather propose *an* approach, and analyze its merits and implications. Moreover, the goal of this thesis is not to implement all engineering aspects of the architecture, such as making it reach Internet scales in practice, but rather to show how its basic principles can be applied in different contexts to solve some of the problems that inter-domain routing faces today.

1.4 Thesis Outline

This thesis is structured as follows, chapter-wise. The publications on which each chapter was based are mentioned for clarity.

Chapter 2 reviews related work and highlights the novel aspects of this thesis, in particular the routing outsourcing proposal (cf. Chapter 3), compared to previous work. The background information that is relevant to the main topics that this work touches on is also provided for completeness. This chapter was heavily based on the background work performed for the formation of my research plan at the beginning of the second year of my PhD studentship. The plan has been approved by Prof. Plattner and was formally submitted to the doctoral administration office of ETH Zurich for the continuation of my studies.

Chapter 3 introduces the architectural approach for gradual inter-domain routing centralization based on SDN. The associated benefits, challenges and implementation mechanisms—such as cumulative routing outsourcing—are analyzed in detail. This chapter is based on our Elsevier journal publication in the Special Issue on SDN & NFV [151], our HotNets paper [148] and our ONS extended abstract [149]. These publications were joint work with Xenofontas Dimitropoulos, Adrian Gämperli and Bernhard Ager.

Chapter 4 describes the first use case where the approach of Chapter 3 can be applied, namely improving the convergence properties of inter-domain routing while being compatible with BGP. This chapter

presents SIREN, a custom hybrid BGP-SDN emulation framework for running AS-level routing convergence experiments, a proof-of-concept multi-domain routing controller that runs on top of SIREN, and the evaluation results and associated insights in terms of convergence times and churn rates for custom multi-AS topologies. This chapter is based on our Elsevier journal publication in the Special Issue on SDN & NFV [151], our HotNets paper [148], our ONS extended abstract [149] and our SICGOMM demo extended abstract [97]. These publications were joint work with Xenofontas Dimitropoulos, Adrian Gämperli and Bernhard Ager.

Chapter 5 presents a general algorithmic framework for estimating the policy-compliant path diversity and bisection bandwidth of a network, including the AS-level path diversity. In particular, it describes important use cases for policy-compliant min-cuts, such as AS-level graphs governed by valley-free policies or more liberal extensions of valley-free. These use cases further motivate the analysis of the IXP-based overlay graph calculations in Chapter 6. Afterwards, it gives the idea and the algorithm for a custom graph transform to be used for calculating policy-compliant min-cuts on an arbitrary network graph, modeling policies as regular expressions and consequently as finite state automata. The methodology is further applied in Chapter 6. Chapter 5 is based on our INFOCOM paper [144], which was joint work with Rowan Klöti, Bernhard Ager and Xenofontas Dimitropoulos.

Chapter 6 presents the second use case where the approach of Chapter 3 can be applied, namely the provisioning of guaranteed end-to-end services using inter-domain mediators deployed on IXPs, called Control eXchange Points (CXPs). This chapter studies the associated IXP multigraph and its potential as the data plane for such services in terms of client reach, path diversity and incremental deployment. We formulate and evaluate pathlet stitching algorithms over dense IXP-based multigraphs using a custom simulation framework and real peering datasets. We further analyze observations and insights relevant to the CXP concept and its potential in detail. This chapter is based on our ETH Zurich Technical Report [153], our SIGMETRICS extended abstract [154] and our ONS extended abstract [150]. These publications were joint work with Rowan Klöti, Matthias Rost, Xenofontas Dimitropoulos, Bernhard Ager, Stefan Schmid and Panagiotis Georgopoulos.

Chapter 7 concludes this thesis with a general summary of the work and the review of its contributions. It also provides a critical assessment of the work within a larger research context and gives an outlook on future work in the associated areas of interest.

Lastly, **Appendix A** gives the full list of publications and software-related contributions that were produced during the process of writing this thesis. **Appendix B** briefly presents some additional research work in which I participated, and which is relevant to this thesis as a complementary body of research. The main axis of that work is the use of SDN in multiple contexts, ranging from network security to testbeds and new services; such services may be deployed under the supervision of a multi-domain RaaS contractor according to our basic proposal and architectural approach.

Chapter 2

Towards a New Routing Architecture: Related Work and Background

> To know what you know and what you do not know, that is true knowledge.
>
> —Confucius

This chapter provides the related work and background information that is required to support our work on Chapter 3 (inter-domain Routing-as-a-Service architectural approach) and Chapter 4 (first use case of the approach: inter-domain routing convergence). The content is based on the literature survey that I carried out during the first year of my PhD, accompanied by the updated knowledge added during the course of the following two years. It contains a survey on inter-domain routing and BGP (cf. Section 2.1), an overview of Software Defined Networking (SDN) (cf. Section 2.2), and the research motivation behind outsourcing network management functionality (cf. Section 2.3). We conclude this chapter in Section 2.4. We further note that the comparison with related work and the required background knowledge with respect to Chapter 5 and Chapter 6 are included within the body of the respective chapters, in order to produce a coherent flow for the reader.

2.1 Inter-Domain Routing and BGP: A Survey

In this section, we perform a brief and concise survey on inter-domain routing in the context of BGP. The survey is not intended to be exhaustive but rather to capture the most important—i.e., heavily cited—research work in the field. In some cases we prefer to favor the importance of the work as opposed to its freshness. We start with the basics of BGP in Section 2.1.1, and then continue in Section 2.1.2 with the policies that govern inter-domain routing; these are important for researchers to understand the way in which operators use BGP in practice. We then describe the most significant challenges that BGP faces (cf. Section 2.1.3), together with solutions proposed in order to address these challenges (cf. Section 2.1.4). Lastly, we conclude with some remarks and observations on where exactly our work comes into play in the context of BGP-related research.

2.1.1 Basics

An Overview. BGP has been the de facto inter-domain routing standard in the Internet for the last three decades [172, 173, 212]. The basic specification[1] of BGP—current version 4—is provided within the IETF RFC 4271 [212]. The primary function of this routing protocol is the exchange of IP prefix reachability information between ASes, i.e., network domains which are governed by separate administrative entities and are interconnected within the Internet. Each AS is identified using a globally unique identifier, the Autonomous System Number (ASN) [59]. Not all identifiers are used though in global routing decisions; there can be private ASNs assigned e.g., by an ISP to its customer that simply operates on default routes received from its provider.

The exchange of routing information is based on the *policies* of the individual domains, as we will explain in Section 2.1.2. BGP routing converges in a distributed fashion into policy-compliant paths, i.e., IP-level routes, crossing the Internet on the AS level. Prefix reachability information is propagated between ASes over eBGP sessions formed between their peering border routers. This information spreads

[1] The numerous RFCs that describe proposed extensions of the protocol are not mentioned here for brevity.

internally within each AS based on iBGP sessions between all the BGP-speaking routers, connected either directly via a logical full mesh or indirectly with the help of a route reflector [26]. For the rest of this work we will focus primarily on eBGP and how routing takes place on the inter-domain level. Griffin et al. [112] introduce the stable paths problem across domains and exhibit that BGP can be viewed as a distributed policy-constrained algorithm for solving this problem. The solution does not represent a global optimum, but rather an equilibrium point in which each node is assigned its local optimum. The latter is based on the routes that each AS's neighbors have selected and the preferences of the AS itself. This observation is very important for our work, which aims at finding optima involving multiple domains under the supervision of a logically centralized entity, as we will show in Chapter 3, Chapter 4, Chapter 5 and Chapter 6.

(Best) Path Selection and Propagation. BGP is a distributed path-vector routing protocol; every router that speaks BGP decides locally the "best" AS path per destination prefix that it learns, and then propagates this information further to its neighbors. The AS path is a sequence of ASNs, denoting the AS-level path towards a reachable IP prefix. The choice of the best path is based on checking the list of attributes corresponding to the learned route in an ordered fashion, using priorities [64]. These attributes include local policies, AS path lengths and other fields such as router IDs that serve as tie-breakers when all the preceding checks have failed to reach a decision. We note the importance of the *local preference* attribute, which is used for setting policies for outbound traffic; this aligns the path selection process to the policies employed by the ISP. Inbound traffic engineering can be performed, e.g., via the Multi-Exit Discriminator (MED) attribute, albeit in a quite limited way in practice [47]. Applying filtering on the BGP updates that are received from neighbors (inbound) or on the updates that are exported to neighbors (outbound), is another way to influence the path selection process on the inter-domain level, based on the domain policies. Routing loops involving the selected paths are avoided by checking whether a received AS path announcement includes the local ASN; if yes, it is discarded from the process. However, state inconsistencies between routers, e.g., during convergence periods, can lead to transient loops [202]. The path propagation process can be rate-limited using the Minimum Route Advertisement Interval (MRAI) timer of BGP [212]. We will revisit the implications of distributed BGP-

based path selection and propagation when interacting with its logically centralized counterpart (i.e., a multi-AS SDN routing controller) in Chapter 4.

2.1.2 Inter-Domain Routing Policies

BGP is primarily a policy-based routing protocol. Caesar et al. [47] identify common design patterns pertaining to BGP policies that are typically employed by ISPs. The most common patterns are: enforcement of business relationships based on the role of the ISP within the Internet ecosystem (customer, provider, peer), traffic engineering (inbound, outbound), Service Level Agreement (SLA) with client networks, peering agreements based, e.g., on balanced ratios for the exchanged traffic, reduction of transit costs via direct peering, scaling up the routing information exchange process, stabilizing and reducing routing fluctuations and securing the interactions between the ISP's AS and its neighbors.

We observe that the enforcement of multifarious routing policies is a complex multi-objective optimization problem that is closely coupled with the business logic of the AS (enterprise, ISP, content provider, etc.). Ideally, the solution to this problem should be compiled into low-level configuration of the Routing Information Base (RIB) on the domain routers, which is by itself an important research challenge due to the numerous BGP knobs that are available for this compilation [31]. A seminal work in the context of BGP policies is the one of Gao et al. [98]. That work observes conventional traffic engineering practices of ISPs that provide each AS with significant flexibility in selecting its local policies. In consequence, the authors derive associated policy guidelines for ASes. These guidelines ensure BGP routing convergence even under topology or policy changes, and represent the famous "valley-free" conditions pertaining to the roles of customer, provider and peer. In a nutshell, these conditions state that an AS prefers to reach IP prefixes using paths going over its customers more than paths going over its peers, and paths going over its peers more than paths traversing its upstream transit providers. Route export rules are enforced accordingly to adhere to these conditions. The valley-free conditions correspond to financial incentives and the flow of money involved in transit routing in the Internet. More details can be found in the work of Gao et al. [98]. These policies are naturally a simplification of the actual business relationships

in the Internet; in reality inter-domain policies are much richer than that [47].

2.1.3 Challenges

While BGP is a well established Internet routing protocol, there are numerous open issues and research challenges associated with its design and operation that still remain unaddressed. We have seen a lot of proposals for incremental improvements and even clean-slate redesigns of how inter-domain routing should work, but BGP remains one of the most ossified protocols of the Internet architecture and has not retrofitted the proposed ideas. Yannuzzi et al. [260] provide a comprehensive survey of the challenges faced in the context of inter-domain routing, along with the most compelling contributions and research efforts that aim at addressing each of the exposed issues. It also presents perspectives on why these issues remain largely unsolved and why most of the proposals have not been adopted. The basic problems mentioned are: slow convergence and chattiness of BGP, scalability, expressiveness and safety of policies, robustness, security, lack of multipath routing and absence of end-to-end QoS support. Our work touches on most of these problems from a different angle, attacking mainly the convergence and QoS support challenges. We also refer the reader to the work of Rexford [213] for further challenges faced by Internet routing, for example policy reconciliation, routing consistency across (multi-)AS network fabrics, end-to-end path quality setup and monitoring as well as traffic management and routing function placement (e.g., distributed vs centralized, trade-offs between placement and performance).

Convergence and Stability

Previous research has studied BGP convergence properties using models of the BGP protocol [113, 115, 246]. Griffin et al. [113] perform a full (static) analysis of the convergence properties of BGP examining the notorious “dispute wheels” and convergence/divergence scenarios. In particular, they present an abstract model of BGP and define several global sanity conditions on routing policies that are related to BGP convergence/divergence. For each condition it is shown that the complexity of statically checking BGP convergence is either NP-complete or NP-hard. Early experimental studies have also highlighted the issue of

delayed Internet convergence due to BGP [111,160]. Labovitz et al. [160] examine the latency induced due to the convergence properties of inter-domain routing during Internet path failure, fail-over, and repair. These delays stem from temporary routing table fluctuations which take place during the operation of the BGP path selection process on Internet backbone routers. During these periods of delayed convergence, it is shown that end-to-end Internet paths experience intermittent loss of connectivity, as well as increased packet loss and latency.

More recent studies of BGP dynamics suggest that some aspects are getting better over the years, but temporal artifacts of BGP convergence still persist [169], while there is great difficulty in predicting or even explaining BGP's behavior at scale. One of the main causes for delayed convergence is the path exploration phenomenon: after a routing change (e.g., due to topological failures) that invalidates a current best path, a BGP router will select a new best path. The router, however, may choose and propagate a path that has been already obsoleted during its selection process. This obsolete path may, in turn, be chosen by other nodes as their new best path, resulting in invalid paths being propagated further. Path exploration has been quantified by Oliveira et al. [196]; BGP can actually take several minutes to converge after a routing change. The important impact of slow BGP convergence on VoIP services running over the Internet has been demonstrated in the work of Kushman et al. [159]. In general, BGP convergence times in the order of seconds up to minutes [160] may not be acceptable for the stable operation of various Internet applications, such as real-time voice or video communications.

Moreover, BGP convergence may be severely affected by route flapping. Route flapping may occur when a router alternately advertises a destination prefix via one route and then another (or as unavailable, and then available again) in quick sequence. This behavior is caused by pathological conditions (hardware errors, software errors, configuration errors, intermittent errors in communication links, unreliable connections, etc.) within the network which cause certain reachability information to be repeatedly advertised and withdrawn. Route flapping may delay convergence in large (inter-)networks such as the AS-level Internet.

Another important aspect of convergence and stability is the churn rate of BGP, i.e., the number of routing messages per second that need

to be exchanged during the convergence period. Churn can be influenced by a variety of factors, namely the size of the routing table and the number of routing prefixes affected by a routing change, time scheduling artifacts in the router BGP stack, MRAI timers, route suppression mechanisms etc. Elmokashfi et al. [74] analyze the impact of the growth of Internet topology and the churn induced by this growth on the scalability of BGP and the number of update messages needed for it to work. As a follow-up work [75], the authors performed a longitudinal study of BGP churn as seen from four networks in the core of the Internet. Their primary goal was to see whether churn in the Internet increases so fast that it poses a challenge to BGP operation. The answer to this is mainly negative: churn at the core of the Internet increases at a rate that is lower than the growth in routing table size, and does not pose an immediate threat to routing scalability. However, many local effects and misconfigurations contribute significantly to churn and can actually have a severe effect on the proper function of the routing system. Such phenomena become more prevalent with multi-homing and load balancing practices, address fragmentation and lack of efficient prefix aggregation. In a different context, Teixeira et al. [237] analyze the effect of intra-domain churn and the dynamics of hot-potato routing in IP networks: the authors show that the propagation of this churn over BGP events can be harmful for inter-domain routing stability.

Policy Management and TE Enforcement

Feamster et al. [86] explore the fundamental trade-off between the expressiveness of route ranking and policy-based routing information filtering in the context of AS autonomy on the one hand, and routing safety and stability on the other hand. To illustrate the difficulties of using BGP for policy enforcement we refer the reader to the work of Quoitin et al. [208], explaining how BGP can be tuned for inter-domain Traffic Engineering (TE). The authors show that an AS has more control on its outgoing rather than its incoming traffic. Several techniques can be used to control the incoming traffic, but they have limitations. For example, selective advertisements and more specific prefixes may increase the size of BGP routing tables, limiting scalability. AS path prepending might be employed, but it is very difficult to select the appropriate value of ASN repetitions to be prepended to achieve a given goal due to the dynamic nature of the BGP path exploration process [196]. As an

alternative, redistribution communities [56] might be used to flexibly influence the redistribution of routes towards non-directly connected ISPs. This mechanism still requires agreement between the involved ISPs regarding the symbolic meaning of communities, while it is important that the community information is preserved across the multi-AS redistribution paths. This is not always guaranteed over multiple AS hops. As we explain in Chapter 3, inbound TE on the inter-domain level might become easier with the help of multi-AS routing controllers that resolve policy conflicts (i.e., coupling the outbound policies of the source AS with the inbound policies of the destination AS) to enable flexible and policy-compliant TE. Decoupling the high-level policy from the low-level routing mechanisms might also prove beneficial in terms of innovation and evolvability for new services according to other recent works in the field of intra- and inter-domain SDN [238].

Security

In the field of security, Ballani et al. [25] present a detailed study of Internet prefix hijacking and interception. The authors estimate that ASes higher up in the routing hierarchy can both hijack and intercept traffic directed to any prefix from a significant fraction ($> 50\%$) of ASes in the Internet. Lack of authentication mechanisms is one of the main reasons why such security holes in BGP can be exploited and cause such widespread damage. Khare et al. [141] apply a detection algorithm to BGP routing updates from 2003 through 2010 and identify five to twenty prefix hijacking incidents every year, most of which were previously unknown to the research and operation communities at large. They typically hijack prefixes owned by a few tens of networks, last from a few minutes to a few hours, and pollute routes at most vantage points. We believe that a logically centralized supervisor that has a wide enough view of (parts of) the inter-AS network may be able to shed some light on such incidents or even prevent them in the first place. The cooperation between multiple such supervisors may accelerate this process. Furthermore, traffic engineering actions applied by a logically centralized platform can be coupled with detection mechanisms [105,106] in order to detect or even repel wide area DDoS link-flooding attacks, such as Coremelt [231] and Crossfire [136].

2.1.4 Solutions

Tools/Practices for Fast Convergence

Operations: MRAI and Route Flap Damping. Best practices refined over years of operational experience have introduced mechanisms such as MRAI timers to rate-limit routing advertisements, or selective route flap damping to absorb routing oscillations caused by unstable prefixes [184]. Persistent route oscillations that make routing unstable and which stem from route reflection and AS confederation schemes are analyzed in detail in RFC 3345 [184]. In reality, operators follow diverse practical strategies to achieve fast convergence, e.g., ditching the MRAI usage altogether [102] to avoid "idle waiting" during route propagation, or abolishing route flap damping as harmful for routing performance. We refer the reader to the thesis of Gill [103] for a detailed analysis of the impact of MRAI timers on BGP convergence time. In their seminal work [179], Mao et al. analyze the interaction between the BGP route withdrawal process and its route flap damping mechanism for ensuring the overall stability of the Internet routing system. This interaction can, depending upon the topology, suppress for up to one hour the propagation of a route that has been withdrawn once and re-announced, exacerbating the convergence times of relatively stable routes. The paper proposes a simple fix to this withdrawal-triggered suppression called selective flap damping. Despite these shifts and changes over the years, the basic problem with slow convergence persists due to basic design artifacts of the BGP protocol and the difficulty in globally enforcing countermeasures on the AS-level Internet.

Research: Stabilizing BGP. In the research literature, multiple proposals have spawned in order to counter the convergence problems of BGP. Bremler et al. [37] propose a modification to the BGP waiting rules in order to limit both the update "chattiness" and the convergence times after link-up events. Lambert et al. [163] propose the addition of a timer mechanism to enforce order in routing messages, to reduce path exploration, and to control—i.e., regulate—convergence time. The recent work of Godfrey et al. [107] describes a modification in the route selection process of BGP, favoring stability with some deviation from the operator's preferred routes. In BGP-RCN, i.e., BGP with Route Cause Notification [201], each BGP update message carries information about the specific cause which triggered it; nodes can thus discard new

paths that have been obsoleted by the same failure. This modification enables to predict an upper bound of $O(d)$ on routing convergence delay for BGP, where d is the network diameter as measured by the number of AS hops. Path exploration damping is analyzed by Huston et al. [128]; its goal is to reduce update churn and decrease average times to restore reachability, as compared to current BGP mechanisms such as MRAI timers. The common denominator of such approaches is the requirement of global modifications to the protocol itself. In contrast, our intention is to use a new architectural approach for routing that is compatible with BGP and can help improve its behavior through safe and staged deployment, shielding the legacy network from unintended changes.

New Architectures and Approaches

Hybrid Routing. Subramanian et al. [232] describe a hybrid link-state and path-vector protocol called HLP as an alternative to BGP that has better scalability, isolation and convergence properties according to the authors. HLP involves link-state routing within the customer cones of tier-1 ISPs and path-vector routing between tier-1s. Using this work as their starting point, Alim et al. [13] propose an algebraic specification of a path problem and decompose it into sub-problems where different protocols are applied. The authors attempt to clarify the trade-offs between the fast convergence of link-state and the low space requirements of path-vector; however, modeling mixed BGP-like protocols such as HLP is still an open problem due to the inability to adequately model BGP with semirings. In contrast, in our work (cf. Chapter 4) we *emulate* a hybrid path-vector and link-state multi-AS environment using production BGP and SDN code, and we measure their interplay regarding convergence as a use case.

More Flexible Routing. Yang et al. [259] present the design of NIRA, an inter-domain routing system that practically supports user choice, enabling source-determined routing and flexible route selection. The downside is that source routing leaves transit domains with very little control and introduces difficult scalability and security challenges. Wen et al. [257] present a multipath inter-domain routing protocol, called MIRO. MIRO defaults to the single-path routing provided by conventional BGP, but allows ASes to negotiate alternative paths as needed. This provides flexibility where required while remaining compatible with BGP. Compared to source routing, MIRO gives transit

ASes more control over the flow of traffic in their networks. Teixeira et al. [235] states that the current mechanism of early-exit or hot-potato routing, where each router in an AS directs traffic to the "closest" border router based on the intra-domain path costs, is convoluted, restrictive, and sometimes quite disruptive [237]. The authors propose a flexible mechanism for routers to select the egress point for each destination prefix, allowing network administrators to satisfy diverse goals, such as traffic engineering and robustness to equipment failures. Such routing schemes could be potentially accommodated over the control plane of routing outsourcing contractors (cf. Chapter 3) and Control Exchange Points as path mediators (cf. Chapter 6).

Next-hop and Negotiation-based Routing. Schapira et al. [218] propose making BGP simpler. The authors argue that the AS path attribute, which lists the sequence of ASes that propagated the route, is the root of many of BGP's problems. They propose a transition from today's path-based routing to a solution where ASes select and export routes based only on neighboring ASes and consequently discuss the merits and limitations of next-hop routing. In preceding work, Mahajan et al. [177] outline an architecture based on explicit coordination between ISPs to predictably control traffic and to bring policy conflicts under the spotlight so that they can be resolved. The authors acknowledge the fact that implicit policy conflicts can lead to unpredictable and unstable outcomes. Such conflicts are difficult to resolve in a distributed manner in the limited frame of an information-hiding protocol as BGP. In their follow-up work, the authors explore negotiation as the basis for cooperation between competing entities, for the specific case of routing between two neighboring ISPs [178]. They present a negotiation framework where adjacent ISPs share information using coarse preferences and jointly decide the paths for the traffic flows they exchange. The challenge of remote control of routing across multi-hop AS chains is not considered. In contrast, our work sheds more light on how it can be addressed, proposing RaaS mediators for implicit collaboration among ISPs to achieve joint goals (e.g., end-to-end QoS).

Secure Routing. Regarding security counter-measures, Kent et al. [140] describe a secure, scalable, deployable architecture (S-BGP) for an authorization and authentication system that addresses most of the security problems associated with BGP, such as prefix hijacking from untrusted entities propagating forged routing updates. The paper

discusses the vulnerabilities and security requirements associated with BGP and explains how S-BGP addresses these vulnerabilities and requirements. Regarding troubleshooting, Feamster et al. [84] describe the design and implementation of RCC, the router configuration checker, a tool that finds faults in BGP configurations using static analysis. These faults and misconfigurations may raise security and availability concerns conflicting with the ISP policies. RCC detects faults by checking constraints that are based on a high-level correctness specification. The tool detects two broad classes of faults: route validity faults, where routers may learn routes that do not correspond to usable paths, and path visibility faults, where routers may fail to learn routes for paths that exist in the network. Furthermore, Zhu et al. [265] propose a new routing system which bifurcates structural and dynamic information in order to gain resiliency to minority compromises in the infrastructure and achieve higher availability than BGP. Finally, Zhang et al. [262] present SCION, a new routing paradigm designed to provide route control, failure isolation, and explicit trust information for end-to-end routes. SCION separates ASes into groups of independent routing sub-planes, called trust domains, which then interconnect to form complete routes. Such architectures may be logically hosted on the control plane of multi-domain routing controllers[2], as a set of applications related to security, troubleshooting (i.e., fault detection) and path computation.

Pathlet Routing and Dynamic Route Computation. Godfrey et al. introduce a new scheme, called pathlet routing [108], in which networks advertise fragments of end-to-end paths from which a source can assemble an end-to-end route. The authors claim that this is a flexible mechanism that can emulate a number of existing routing protocols, including BGP and unrestricted source routing, with better scalability regarding policy expressiveness. This work was one of the basic related papers for the CXP-based routing scheme which we will describe in Chapter 6. Moreover, we note a very interesting paper regarding the need for dynamic computation of routes in routing protocols [46] in general. This paper advocates a different approach to reduce routing convergence issues: essentially sidestepping the problem by avoiding it in the first place. Rather than recomputing paths after temporary topology changes, the authors argue for a separation of

[2]For example, a hierarchy of controllers corresponding to the trust structure of SCION, with centralization taking place only on the routing sub-plane level.

timescale between offline computation of multiple diverse paths and online spreading of load over these paths. They believe that decoupling failure recovery from path computation leads to networks that are inherently more efficient, more scalable, and easier to manage. This offline computation of inter-domain paths at the contractor's side is suitable for our approach [148], since expensive offline calculations can be performed by a centralized entity (multi-domain controller of outsourcing service contractor) while online failure recovery can occur in a different time-scale. Ascigil et al. [18] further consider the challenge of scale in terms of domain-level end-to-end route computation in the Internet. According to the authors, their performance results seem to debunk the conventional wisdom that centralized routing computations do not scale; this is an encouraging observation that supports the CXP concept and pathlet stitching algorithms that we present in Chapter 6. In particular, it provides an incentive for our experiments on the computations of domain-level routes that involve a large fraction of the ASes present in the Internet.

2.1.5 Summary and Observations

We have seen that there is still a large pool of challenges to address in the context of inter-domain routing in general and BGP in particular. We note that most of the aforementioned solutions have not succeeded in actually changing the way the inter-domain routing system works, mainly because of three reasons. First, BGP is a widely adopted protocol implemented by many stakeholders and is therefore very difficult to change. Second, ISPs cannot be easily convinced to take the risk of adopting a proposed improvement unless substantial profit is imminent. Third, the strict requirement of maintaining compatibility with legacy equipment nips many good ideas in the bud. Our work tries to attack some of these challenges with a partially disruptive and novel, yet incrementally deployable approach. Finally, we note that none of the aforementioned approaches proposes partial centralization at the inter-domain level; while this idea may seem radical and contradictory with the basic principles of Internet design [66], in reality there is a strong case behind such a proposal as we will show in Chapter 3, with tunable trade-offs between full distribution and partial centralization.

2.2 Software Defined Networking

In this section, we first analyze the basic principles behind SDN (cf. Section 2.2.1). We continue with related work on centralized SDN-like routing control planes within a network domain (cf. Section 2.2.2), and end the section with research work that is conducted in parallel in the field of inter-domain SDN by other researchers (cf. Section 2.2.3).

2.2.1 Basic Principles

Separation of Control and Data Plane using Abstractions. The main idea behind SDN is the escape from the "closed-box" or "mainframe" model of network components (switches, routers), and the transition to open forwarding APIs and protocols. Using these interfaces, certain entities called the controllers—constituting the control plane—can effectively control the forwarding behavior of the network devices—constituting the data plane. In general, the separation of the planes follows a long history of proposals, such as the ITU SS7 [131] and the IETF ForCES [258] protocols, as well as the active networks' initiative [206], and is based on the following fundamental abstractions.

First, the *forwarding abstraction* abstracts away the underlying proprietary hardware and software of the network devices, allowing to express intent independently from implementation. One example of such an abstraction is OpenFlow [183], which enables a controller to manipulate the Forwarding Information Base (FIB) of an OpenFlow-enabled switch. Control is exerted in the form of flow rules that are installed by the controller and reside within the flow table of the switch. Rules are built using a match-action primitive: *match* selected parts of the traffic (e.g., a range of source and destination IP addresses), apply *actions* on the associated packets (e.g., forward, drop, rewrite, etc.).

Second, the *network state abstraction* abstracts away various distributed mechanisms for gathering and managing network state. This is implemented by a Network Operating System (NOS) or controller that offers a consistent, global network view to the applications running above. A natural instantiation of this view in the networking context is an annotated network graph. This offering enables centralized algorithms to configure the network; the complexity of state distribution is addressed within the NOS rather than at the algorithmic layer, enabling the

creation of simpler code for network control applications. Moreover, this complexity is not replicated at every network device, but resides in the centralized NOS software. An example of such an abstraction is ONIX [146], offering a Distributed Hash Table (DHT) solution for managing dynamic state which should be eventually consistent, and a replicated transactional (SQL) storage option for maintaining strong consistency on critical stable state. We note that network hypervisors, such as FlowVisor [224] or OpenVirtex [11], allow multiple NOSes and network applications to share the same network substrate using virtualization and isolation techniques such as flow space slicing or overlay edge-to-edge tunneling. This capability is based on a *virtualization* abstraction.

Third, the *specification abstraction* allows a network control application running on top of the NOS to express desired behavior and policies at a high level of abstraction. This behavior can relate to isolation, access control, QoS and any other possible network application. This abstraction is not responsible for implementing that behavior on the physical network infrastructure. An example of such an abstraction is Frenetic [90], a high-level language for reading and writing state from/to distributed collections of network switches based on SQL-like queries. The Pyretic [210] framework goes one step further with the modular composition of SDN applications.

A full survey of the benefits, challenges—e.g., scalability and security—and solutions that are associated with the SDN paradigm are out of the scope of this thesis. For further information we refer the reader to the recent works and SDN surveys of Nunes et al. [193], Jarraya et al. [135], Kreutz et al. [157] and Hu et al. [124].

2.2.2 Centralized SDN-like Routing Control Planes

The principle of decoupling the network control and management from the data plane, and the consequent (logical) centralization of routing control, has fueled multiple research proposals on centralized routing control planes, primarily in the context of intra-domain routing. We analyze those proposals in the following.

Feamster et al. [85] state that the limitations in today's routing system arise in large part from the fully distributed path-selection

computation that the IP routers within an AS must perform. To overcome this weakness, the authors claim that inter-domain routing should be separated from today's IP routers, which should simply perform packet forwarding (for the most part). Instead, a separate Routing Control Platform (RCP) could select routes on behalf of the IP routers in each AS and exchange reachability information with other domains. RCP is implemented in the follow-up work of Caesar et al. [45]. The authors present the operation results of an RCP, which collects information about external destinations and the internal topology and selects the BGP routes for each router in an AS. RCP is a logically centralized platform, separate from the IP forwarding plane, that performs route selection on behalf of routers and communicates selected routes to the routers using the unmodified iBGP protocol.

In a similar context, Van der Merwe et al. [242] present the centralized Intelligent Route Service Control Point (IRSCP) that allows route selection to be performed outside the routers, and also allows this selection to be informed by external network intelligence. This provides "dynamic connectivity management", leveraging the mechanisms of the BGP protocol, similarly to the RCP. Rothenberg et al. [215] propose an architecture inspired by the RCP, based on previous insights regarding benefits in flexible routing, enhanced security, and ISP connectivity management tasks. The authors here discuss routing platforms in the context of OpenFlow/SDN, describing potential use cases and identifying deployment challenges and advantages. They propose a controller-centric hybrid networking model and present the design of the RouteFlow Control Platform (RFCP), along with the prototype implementation of an AS-wide abstract BGP routing service, advocating a BGP-free edge. This work is heavily based on the previous design and implementation of "virtual routers as a service" presented by Nascimento et al. [187]. The authors present a commodity routing architecture that combines the line-rate performance of commercial hardware with the flexibility of open-source routing stacks running (potentially remotely) on general purpose computers. This architecture enables a flexible resource association between IP routing protocols and a SDN-programmable physical substrate, enabling a hybrid routing strategy for incremental adoption of RCPs. ONOS [33] is another SDN-based implementation of a RCP/NOS that enables, among other things such as state distribution, seamless inter-networking between SDN routing planes and plain IP.

ONOS is motivated by the performance, scalability, and availability requirements of large operator networks.

Wang et al. [249] present Morpheus, a routing control platform that is designed for policy-based configurability. According to the authors, Morpheus enables a single ISP to realize a much broader range of routing policies than today's tools, without requiring changes to the underlying routers or collaboration with other domains. The design of this platform separates route classification from route selection and allows large ISPs to capitalize on their path diversity. Moreover, Fu et al. [94] strengthen the research direction of centralized routing control schemes, showing that control centralization can provide faster routing convergence than link-state routing protocols within a domain.

In the context of forming centralized inter-domain routing control planes, Gupta et al. [118] revisit the arguments regarding the need for coordination among mutually distrustful parties (i.e., the routing domains). These arguments lead to the requirements that BGP-like protocols should provide, like policy autonomy, flexibility, and privacy. BGP provides these properties via the distributed execution of policy-based decisions during the iterative route computation process. According to the authors, this approach has poor convergence properties, makes planning and fail-over difficult, and is extremely hard to change. To rectify these and other problems, the authors propose a different approach to inter-domain route computation, based on Secure Multi-Party Computation (SMPC) which can run for example on cloud servers at a central location. SMPC may be a useful implementation base for dealing with the security, privacy and trust implications that the routing outsourcing approach of Chapter 3 is associated with.

Lastly, we refer the reader to the PhD dissertation of Iqbal [133] regarding an evaluation of a logically centralized approach for the control and management of large computer networks. This work aims at bridging the gap between the extremes of distribution and centralization of network control. The logically centralized approach for the design of a network decision plane (e.g., for routing-related decisions) can be realized using a set of physically distributed controllers in a network. The results show that logical centralization can provide better scalability and fault-tolerance while maintaining performance similar to a traditional distributed approach. The insights yielded from this work encourage the basic research direction of this thesis w.r.t. routing centralization.

2.2.3 Inter-domain SDN

Gupta et al. [117] propose SDX, a Software Defined Internet eXchange. The SDX notion advocates the deployment of SDN-capable data plane elements within IXPs, controlled by a SDN controller; this setup enables richer traffic matching, more direct control over the data plane and ISP-centric applications such as inbound traffic engineering, WAN load balancing and IXP fabric virtualization. The general areas of application innovation are security, forwarding optimizations, peering and remote control of inter-domain routing. The authors, besides describing the basic SDX architecture and its applications, explain its scalability trade-offs in the control plane (temporal dimension) and the data plane (spatial dimension). SDXes can be used as building blocks for the implementation of the CXP architecture that we describe in Chapter 6, capitalizing on gradual IXP-based deployment.

Bennesby et al. [29] propose the use of an inter-AS routing component on per-domain SDN controllers, replicating BGP-like functionality over a distributed AS controller fabric. This architecture enables the decoupling between BGP routing policy and network infrastructure, allowing for innovation in inter-domain routing. As a next step, Thai et al. [238] introduce an Inter-domain Management Layer (IML), based on the horizontal slicing of network resources. IML enables independent SDN ASes to be formed and operated. Furthermore, it provides tools for managing AS borders and sharing resources with other ASes via domain proxies. Lastly, we mention the work on seamless inter-networking between SDN and IP using SDN NOSes [33], and the RouteFlow approach applied on hybrid legacy-SDN networks [215]. In contrast to these approaches, our proposal is based on logically centralizing the routing logic of a multi-domain environment.

Lastly, looking at the bigger picture, Raghavan et al. [209] take the case of separating routing from routers [85] to the next level. They advocate decoupling architecture from infrastructure by leveraging the recent advances in SDN, the re-emergence of software forwarding, and MPLS's distinction between the network's core and edge. The authors sketch their design, called Software-Defined Internet Architecture (SDIA), and show how it would ease the adoption of various new Internet architectures, e.g., in the field of inter-domain routing. These design principles have inspired parts of the work presented in this thesis.

2.3 Outsourcing Network Functionality

Outsourcing is one of the central pillars of our work, involving the process of yielding—partial—control and management of a certain business function to an external, trusted and highly specialized entity to reduce costs and benefit from this specialization. We begin this section with a general overview of IT and network outsourcing (cf. Section 2.3.1). We continue with the basic motivation behind the proposal of outsourcing the routing management, which is of core importance within our work; one of the reasons is the complexity of this type of management (cf. Section 2.3.2). We end the section with an overview of research proposals in the field of Routing-as-a-Service and Network-as-a-Service (cf. Section 2.3.3).

2.3.1 Network Outsourcing in General

IT outsourcing [166] and in particular network outsourcing [58] is a widely-used business model in which a company contracts out a business process that it has previously preformed internally to a third party, that is highly specialized in the respective business area. This enables the company to benefit from the experience and expertise of the third party and to reduce costs for specialized in-house IT personnel. Of course, IT and specifically network outsourcing are accompanied by risks and trade-offs: the reader is referred to the work of Chaudjhury et al. [58] for more details on these areas.

During the last years, the model of IT outsourcing has evolved to potentially include the management of the network of ISPs and enterprise networks. In particular, operators have traditionally viewed the network as their core business. However, declining profit margins in pure transit and bandwidth[3] provision [192] have mounted pressure to reduce costs and to launch new, higher margin services (IPTV, VoIP, cloud-hosting) on top of routing. This has pushed operators to streamline the operating expenses (OPEX) associated with their networks and has given rise to an emerging market of managed services, in which the operation and maintenance of the network (or parts of it) are outsourced to a third party. The third party is responsible for tasks like running the

[3]While the Internet traffic levels are always rising, thus counter-acting the effect of the plummeting price per Mbps, many ISPs are near the turning point where their OPEX starts exceeding their profit from pure traffic routing services.

Network Operations Center (NOC), maintaining the network, building new networks, expanding existing ones, and integrating new services according to the respective requirements of its client(s).

We note that besides the trend for investment in new services which are more complex to handle and provide, albeit much more profitable, there is a clear tendency towards the "flattening" of the AS topology graph [71]. ASes—especially smaller ISPs and enterprise networks—tend to get more and more inter-connected with each other, using private or public IXPs for exchanging traffic at the peering points, thus avoiding paying high transit costs to their upstream providers. This calls for sophisticated traffic engineering, in order to save money, but in parallel increases the associated complexity [170].

2.3.2 Complexity of Network Routing Management

High-level network services and clever traffic engineering, which have been described beforehand, depend on efficient routing and path computation. The art of routing encompasses though many more skills than the mere knowledge of how BGP or other routing protocols work. This includes the optimization of traffic flows via traffic engineering, correctly mapping SLAs to policies, coping with misconfigurations and scalability issues, while at the same time properly securing the network. Each of these oftentimes competing goals requires tuning several knobs in the routing protocol(s). Optimizing how packets are routed within an ISP to satisfy numerous operational and economic objectives is a difficult research problem. On the one hand, a number of advanced traffic engineering techniques have been proposed in the research literature, for example based on integer programming and multi-commodity flow optimization [235]. On the other hand, operators in practice may not have the required knowledge at hand to optimize their network utilization through advanced traffic engineering or to improve security, e.g., through deploying sBGP [140]. Operators are often satisfied with a network that is just running.

In addition, the router configuration code an ISP needs to develop, debug, and update is extensive, while the manual configuration of routers requires many administrator work-hours and is an error-prone process; routing misconfigurations are common and can be very costly.

The reader is referred to Cisco's website [109] where he/she can get an idea of the complexity of the Cisco IOS software for ISP-grade infrastructure and router management. In some cases even simple unintended errors took significant parts of the entire Internet down, resulting in major economic damages. An indicative example of such incidents is the Youtube hijacking performed by Pakistan Telecom in 2008 [40], demonstrating that a local unauthorized announcement can propagate to the entire Internet and cripple useful services because of improper configuration. Another example is the Brazil route leak incident [69] which fortunately did not have global impact because of the appropriate timely reaction of the upstream ISPs.

A formal analysis of the complexity of managing a network is provided by Benson et al. [30]; the authors develop a suite of complexity models that describe the routing design and configuration of a network in a succinct fashion, abstracting away details of the underlying configuration languages. Configuration errors are responsible for a large fraction of network outages, and the authors argue that as networks become more complex the risk of configuration errors increases. The follow-up work of Benson et al. [31] presents three key related findings especially for ISP networks. *(i)* The design of the core data plane is usually service-agnostic and simple, but the control planes for different services become more complex as services evolve. *(ii)* The configuration at the service edge inevitably becomes more complex over time, potentially hindering key management issues such as service upgrades and troubleshooting. *(iii)* There are key service-specific issues that also contribute significantly to the overall design complexity. Therefore, the high complexity could impede the adoption and growth of network-based services, if not properly addressed or streamlined.

We believe that the complexity issue regarding the management of a network as a distributed system within a huge ecosystem of other networks and financial players (peers, customers, competitors, etc.) will persist, even with the emergence of SDN, because it is inherently tied to the core of network engineering and operations. Debugging or precluding policy conflicts and misconfigurations will always be relevant to our research directions. Therefore, we believe that our motivation for making routing management simpler, more efficient and accurate through our approach (outsourcing) is valid. Of course, the complexity is never reduced: what we propose in Chapter 3 is rather to move parts

of it to logically centralized specialized entities, in order to simplify management at different layers of the network control and management plane.

2.3.3 Routing as a Service - Network as a Service

First, we refer the reader to the related work of Lakshminarayanan et al. [162]. The authors introduce the notion of RaaS, motivated by the resolution of tussles [67] between ISPs and customers over the control of end-to-end paths. The authors propose that ISPs should outsource route computation to customers in the context of virtual link stitching and end-to-end virtual path calculation across multiple domains. They claim that route computation should be offered as a service by third-party providers. According to them, outsourcing specialized route computation allows different path-selection mechanisms to coexist, and evolve over time. RaaS is limited to the export of virtual links by ISPs and allows source-based routing through the mediation of third parties. Instead, our work focuses on the benefits of outsourcing parts of the per-AS routing logic and combining the outsourced inputs from multiple ASes for improving inter-domain routing in general, not only for end-to-end QoS or limited use cases. We note though that this work inspired the use case of logically centralized path brokers and QoS mediators that we describe in Chapter 6. In addition, we investigate the advantages of using such mediators in practice, based on a concrete feasibility analysis (for the control and the data plane) and an evaluation of diverse online, offline and hybrid path stitching algorithms.

Gibb et al. [101] present an architecture which enables network functionality to be outsourced to external Feature Providers (FPs). The policies of the FP customers are expressed via service names, rather than locations, and an intelligent control plane maps the policies onto the underlying network. This architecture applies mainly to outsourcing network functions of enterprise networks. In a similar context, Sherry et al. [223] motivate, design, and implement APLOMB, a service for outsourcing enterprise middlebox processing to the cloud, thus benefiting from decreased costs, ease of management, elasticity and fault tolerance. Both studies suggest the export of client traffic which is routed in the data plane to Network Function Virtualization (NFV)-like providers, while we focus on outsourcing (parts of) the routing control logic of multiple domains via cumulative outsourcing. Moreover, these works

are centered around outsourcing in-line network functionality, such as middlebox processing, while we do research on routing control plane outsourcing, mostly with out-of-band mechanisms.

Network as a Service (NaaS), in the context of cloud computing and network infrastructures, is formally presented by Costa et al. [68] and Benson et al. [32]. Costa et al. [68] describe a framework that integrates current cloud computing offerings with direct, yet secure, tenant access to the network infrastructure. According to the authors, NaaS allows tenants to deploy custom routing and multicast protocols. Furthermore, by modifying the content of packets "on-path", NaaS can help implement advanced network services, such as in-network data aggregation, redundancy elimination and smart caching. Benson et al. [32] describe a cloud networking platform for enterprise applications. This platform enables tenants to leverage many of the network functions needed for production enterprise applications to run in IaaS clouds. By leveraging programmable network devices, the authors state that their platform supports features such as isolation, middlebox functions, and QoS. Such architectures are orthogonal to our approach, since they are based on outsourcing the network itself while we claim the need for outsourcing a specific set of network functions, i.e., routing and path computation. However, they also benefit from an economy of scale like our own approach; specifically they advocate an ecosystem composed of multiple clients and a few outsourcing contractors or providers. This potentially leads to lower costs for the clients and larger revenues for the contractors, depending on the employed pricing strategy. Furthermore, the proposed ideas can be utilized by our framework in the context of leveraging the "Cloud" and in general NFV solutions to host the contractor's platform and the routing computation infrastructure.

2.4 General Summary

In this section we gave a brief survey of inter-domain routing and BGP, starting from the basics, moving on to inter-domain policies that govern routing in today's Internet and focusing on the challenges and solutions related to BGP. Afterwards, we described the basic principles of SDN, the design and implementation proposals of centralized SDN-like routing control planes and the recent research on inter-domain SDN. The SDN paradigm may serve as one of the basic pillars for

the evolution of inter-domain routing, as currently reported in the research literature. Finally, we described the motivation and practices of network outsourcing, focusing on the complexity of managing network routing; this complexity can be offloaded to external specialized parties in the context of frameworks providing NaaS and RaaS. In general, we place our work within the intersection of three research areas: BGP and inter-domain routing, SDN, and NaaS/RaaS. Therefore, this chapter provided the necessary background on BGP, SDN and routing outsourcing, complemented with related research work. We will next use this knowledge to delve into the basic approach of this thesis in Chapter 3 and the associated use cases in Chapter 4 and Chapter 6.

Chapter 3

Architectural Approach for Inter-Domain Routing Centralization

Form follows function.

—Louis Sullivan

What new possibilities does inter-domain SDN-based centralization enable? Could the radical proposal of routing outsourcing across domains be realized in some form? Which are the main entities and components of such a framework and how do they interact with each other? In this chapter we analyze the associated trade-offs, starting from the promises of SDN within a domain in Section 3.1, and continue with the benefits and challenges that our approach entails in Section 3.2. We propose using cumulative routing outsourcing as a centralization mechanism in Section 3.3, while we summarize the approach in Section 3.4. This chapter is based on our Elsevier journal publication in the Special Issue on SDN & NFV [151], our HotNets paper [148] and our ONS extended abstract [149] supporting the case for the basic architectural approach for inter-domain routing centralization.

3.1 Centralizing Routing Within a Network Domain

The separation of the network control from the data plane and the consequent logical centralization of routing control promises to drastically simplify routing management within an AS [45, 85]—even using legacy APIs instead of OpenFlow—and to provide faster intra-domain routing convergence [94]. Operators can centrally express—e.g., using a query-like network programming language [90]—enforce [207] and check [138] routing policies using the global view that a SDN-based NOS provides. These policies can be dynamically compiled by frameworks such as the one presented by Monsanto et al. [186].

Moreover, if logical centralization and state distribution are performed with control plane resiliency in mind [146], an AS can benefit from scalable routing models while lowering the overall management complexity that it sees on the northbound service-facing interface [31]. Besides, having a central NOS which controls the AS's routing and forwarding elements simplifies the modification of routing applications, as this process can be achieved solely based on custom software. An open interface to the device layer enables to avoid lock-in in situations where vendor-proprietary software needs to be changed from scratch during a multi-year process, over which the network operator has no control. On-the-fly changes based on well-established agile software engineering practices are desirable for operators; centralized routing control platforms [45] can be used to dynamically implement such changes. We can thus have an evolving routing system for the respective AS since intra-domain routing protocols can change easier, according to the operator's service requirements.

Furthermore, today the control plane on the domain routers is extremely complex and comprises of multiple distributed network functions and protocols: OSPF for intra-domain shortest path routing based on tunable link weights, LDP for distributing MPLS path labels within a MPLS-enabled core, RSVP-TE for QoS-compliant resource reservation along a network path, iBGP for internal dissemination of BGP-learned IP prefixes, MP-BGP for dual IPv4 and IPv6 prefix information exchange, and many more which are not mentioned here for brevity. Each of these protocols comes with its own state distribution mechanisms. SDN can help build simpler control planes by centralizing

instead of replicating complexity everywhere, solving the problem of state distribution once at the NOS's side. It is also simpler to clone/emulate [164, 220] the control plane to safely deploy configuration changes, e.g., ensuring consistent state updates [211], or tune its redundancy properties [167].

Of course such systems need to deal with multiple challenges, such as scalability, resiliency, security and solve problems related to their design and implementation as centralized entities. In particular, centralizing the AS's decisions can lead to consistency challenges. As an example, we note that the non-negligible delay between the controlled network devices and the controller or between the controllers in a distributed setup may lead to outdated information being used for the routing computations, and creates consistency issues regarding the state that is stored on the controllers and the state that the network devices maintain. Moreover, when a controller fails, its state should be transferred to a backup controller; since this cannot happen "post-mortem" we need to keep the state in sync across multiple controller instances, whether they are replicas or independent controllers of parts of the domain. We will dive into some of these challenges from the perspective of a multi-domain routing control platform later in this chapter.

In the next section, we elaborate on the basic topic of this thesis, namely on how the benefits and challenges of intra-domain routing control centralization can be transfered to the inter-domain area. We will thus examine a new approach of routing centralization across domains with the associated advantages and disadvantages.

3.2 Centralizing Routing Across Network Domains

In this section we first describe the basic architectural approach of this thesis (cf. Section 3.2.1), and continue with some of its benefits in Section 3.2.2 and associated challenges in Section 3.2.3.

3.2.1 The Architectural Approach Proposed in this Thesis

Let us assume that we have formed a logically centralized control plane running routing-related processes within an AS. We propose to exploit the benefits of centralization beyond AS boundaries, using a multi-domain NOS that controls a cluster of ASes. With the term cluster[1], we mean a group of ASes which are served through the same NOS, regardless of whether they have specific bilateral agreements with each other; therefore clusters can be either contiguous or disjoint. One of the most interesting aspects of a multi-AS NOS is that as more ASes choose to use it for cross-domain routing, larger AS clusters are gradually formed. The advantages of the approach grow as the size of the clusters increases (horizontal scaling) as we will show in Chapter 4 and Chapter 6. Centralizing the routing control logic of many ASes can potentially benefit inter-domain routing in many ways as we will analyze later. The case for such a NOS is briefly introduced in our extended abstract [149]. The main ideas behind the basic approach, some economic considerations, a short implementation proposal and the associated open questions are described within our HotNets position paper [148]. In the following we delve into the details of the approach.

3.2.2 Benefits

Bird's Eye View over Multiple ASes. The central multi-AS NOS/controller is aware of (parts of) the policies, topologies and monitoring information of the ASes within the cluster it controls. It is therefore the natural point at which inter-domain policy conflicts and problems can be spotted and resolved, and routing paths can be optimized. Coordination beyond AS boundaries can yield efficient paths even if ASes have different policies and optimization criteria. This helps to improve routing stability and mitigate path inflation [178]. This can potentially benefit multiple ASes, even when they are not part of the cluster, as it will result in shorter and more stable end-to-end

[1] This term should not be confused with the term "super domain" which first appeared in the RFC 1478 [230] and whose context is entirely different. In our case, each part of the cluster maintains its policy and identity within the multi-domain setting, meaning that the view of each domain does not change from the perspective of the outside world.

paths, thus reducing network load on a larger scale. Thus, this helps even if the centralized controller cannot "see" all ASes as parts of its cluster; in reality, combining the default limited view of BGP outside the controlled cluster with the more informed view of SDN interfaces within the cluster can still lead to global optimizations. In this context, benefits in terms of convergence and stability will be substantiated in Chapter 4.

Even in the case where the ASes within a cluster are not adjacent, the global view of the NOS is still important for routing optimization. An example is the establishment of stable, multi-hop end-to-end paths with specific attributes, such as maximum latency or minimum bandwidth. Such a use case will be further analyzed in Chapter 6. Moreover, if detailed monitoring data are also exported to the multi-AS NOS, security and network troubleshooting can be further enhanced. For example, the global view can help pinpoint the source of a routing anomaly or failure by analyzing the information acquired by multiple parties, and correlating it with external sources—such as the CAIDA datasets [49] or BGP looking glasses [34]—for cross-validation. Such benefits can only be leveraged when aggregating information from multiple ASes, including detection of prefix hijacking [25] or DDoS counter-measures [105, 106].

Inter-Domain Routing Evolution. Based on the multi-AS NOS, new inter-domain routing algorithms and protocols can be adopted to govern routing between the members of a cluster. Classic distributed routing protocols can be used for fail-over purposes when the cluster-NOS communication is temporarily unavailable. Innovation inside the clusters can be accelerated, while legacy interfaces with the rest of the Internet (BGP) guarantee proper interoperability. For example, we can have lower convergence times and also decreased churn rates through centrally controlling the dynamics of intra-cluster routing, as we will show in Chapter 4. If we choose to work in parallel with BGP, using an overlay approach, we can potentially host new end-to-end services as we will show in Chapter 6. Additionally, hierarchical routing, which benefits routing scalability, is enabled at the inter-AS level thus allowing hierarchical routing schemes, such as HLP [232], to flourish along the NOS control chain. New BGP-like protocols can be defined between NOSes [29], or a NOS may be placed at locations of maximal inter-domain impact such as IXPs [117]. Rethinking BGP in the context of the communication between NOSes which control multiple ASes is one

possible avenue; this can lead to new routing paradigms. One example is the expansion of the reachability information exchange protocol to cover the generic notion of a "header space" [139] or "flow space", being propagated across ASes, instead of plain destination-based IP prefixes.

3.2.3 Challenges

Redundancy. Forming such a centralized NOS to control the inter-domain routing logic of multiple ASes comes with its own set of challenges. We discuss here the technical rather than the financial/political challenges; the latter ones will be analyzed in Section 3.3.3 under the prism of routing outsourcing. First, we need to have backup fail-over schemes in order to keep everything operational even if the AS-NOS communication fails. This can be achieved with NOS agents within the AS that can "think" locally and act when the global NOS is not available; these may be for example RCP-like controllers [45]. In general, we need a redundant architecture that provides resiliency in case of failures; NOS hierarchies and distributed controller fabrics can be a good direction for that purpose [146, 239]. Furthermore, if the NOS and the contractor are located somewhere in the Internet, then the communication between the contractor and the client also goes across the Internet. Thus, the following vicious dependency loop needs to be addressed: *(i)* there is a problem with inter-domain routing, *(ii)* the NOS would want to act to fix the problem, *(iii)* however, the problem also hampers or breaks the communication between the NOS and the client, and *(iv)* thus the problem cannot be fixed. This loop can be broken with sufficient redundant instances of the NOS at different locations of the Intenet, minimizing the probability that no instance at all can help the client ASes deal with the inter-domain routing problem. Another (but probably more costly) countermeasure would be the presence of a dedicated network infrastructure for the communication between the NOS and the client(s). Finally, AS-internal NOS agents could take over the routing decisions in a distributed fashion (on the AS level), while the inter-domain routing problem persists and disrupts NOS-client communication, as mentioned above. These agents operate only in the case of failures and do not conflict with the high-level control logic as long as it is up and running. There is though a potential for conflicts between agents after failures (e.g., in terms of state consistency or policy preferences). Such conflicts are resolved in the same way as BGP does

today; the agents may agree to provide minimal routing reachability while the multi-domain controller is out and log conflicts that affect other services. After recovery, any intermediate conflicts are investigated and resolved again centrally.

Hierarchies and State Distribution. Hierarchical approaches could generally be beneficial for scaling up the multi-AS NOS, since scalability is also an important challenge that we need to counter. The design and implementation of such systems is governed by the trade-offs between logical centralization and physical distribution [167] in terms of state management. For example, we have a trade-off between the "staleness" of distributed state and optimality of the performance of the centralized application logic. Furthermore, robustness to state inconsistency can be achieved by making the application logic (running over the NOS) aware of the state distribution underneath; this of course increases significantly the complexity of writing applications, especially considering a multi-domain, geographically dispersed setup. Moreover, eventual versus strong state consistency requirements may differ per inter-domain application. An example where strong consistency would be required is the provision of hard latency guarantees for real-time applications, such as telesurgery [120], while caching services [38] may be more robust to temporarily inconsistent state.

Security. Moreover, security and privacy for the communication between the served ASes and the NOS should be guaranteed. For a security analysis of the OpenFlow protocol as an implementation of the SDN paradigm, focusing primarily on the implications of reactive routing strategies, we refer the reader to our detailed collaborative work published in the IEEE NPSEC workshop [145] and the comprehensive survey of Scott-Hayward et al. [221]. Security and privacy could also be guaranteed using for example a multi-domain NOS that operates based on Secure Multi-Party Computation (SMPC), as proposed by Gupta et al. [118]. In addition, we note that proper northbound APIs for inter-domain services should be provided. One such interface can be based on the pathlet abstraction [108] using a mediator/broker architectural model that we further analyze in Chapter 6. The interface design should be based on a compromise between efficient routing optimization on the NOS's side, and information hiding on the served AS's side. For example, the NOS should act on the cumulative information received from the cluster ASes, but its actions should not divulge sensitive

information across, e.g., competing ISPs. The latter objective needs intelligent information management and protection mechanisms on the NOS's side, beyond the secure communication between the ASes and the multi-domain NOS.

In General. We note that the challenges associated with scaling up and securing SDN NOS systems as well as defining northbound APIs for SDN NOSes and controllers are cutting-edge research topics for the SDN community. Therefore, our work could potentially benefit from gradual improvements and developments in these areas. We refer the reader to the technical program of the HotSDN workshops of 2013 and 2014 for some recent works in these contexts.

3.3 Outsourcing as a Routing Centralization Mechanism

In this section, we present routing outsourcing across domains as a possible centralization mechanism. We start with the basic scheme in Section 3.3.1, and continue with the technical and financial benefits in Section 3.3.2. Challenges are described in Section 3.3.3, while we provide the outline of the proposed architecture in Section 3.3.4.

3.3.1 Basic Scheme

As we explained in Section 2.3.2, there is a clear motivation behind outsourcing (parts of) the network management to external specialized parties. To partially address the problems that we mentioned in that section (e.g., ever-increasing management complexity and costs), we propose that the routing control logic of a network could be outsourced to a contractor that specializes in routing management, including routing optimization, configuration, troubleshooting, and monitoring. This would constitute a new type of business relationship for technical routing optimizations and service policy enforcement, based for example on SLAs. We next discuss the benefits and challenges of such an idea, as well as a possible outline of the corresponding architecture.

3.3.2 Benefits

Technical Benefits and Associated Mechanisms

The contractor has extensive knowledge on routing and can therefore provide best routing policies tailored to the requirements of a client. Intelligent routing policies and optimizations can help to improve the reliability, performance, and security of a network. From the perspective of a client AS, outsourcing enables it to benefit from advanced traffic engineering, consulting about best routing practices, policy reconciliation, and network troubleshooting. The transition from today's domain-specific situation to an outsourced routing scheme can be handled smoothly by taking several steps. In a first stage, the contractor consults the client about best practices and together they arrive at a policy plan that satisfies the requirements of the client. This may be based on a routing policy specification language, such as RSPL [12]. Based on the agreed plan, the contractor takes over the handling of traffic and optimizes routing within the client's domain. For networks without intelligent traffic handling, this can yield direct performance benefits, e.g., in terms of network load or other performance metrics. Besides, the contractor monitors how network traffic in the client's network changes over time and enforces corrective traffic engineering actions.

Since routing is the basic function on which the main business logic of an ISP is built[2], a prospective client ISP can choose to outsource only specific parts of routing functionality as proposed by Gibb et al. [101]. In the context of our proposal, an ISP may choose to outsource only inter-domain routing interactions at a flow space granularity, e.g., on an IP prefix level for legacy compatibility or on an application traffic level, such as video traffic, for QoS service provisioning. This selective outsourcing may be based on network slicing, virtualization and isolation mechanisms [11, 224]. Obviously, the scheme can work without forcing the client to reveal internal topology information; as we will explain in detail in Chapter 6, the ISP may present itself as a set of abstracted pathlets that connect the edges of its domain and possess specific properties (e.g., QoS or reliability attributes). The technical solution of slicing and selective outsourcing is a counter-argument in the political

[2]However, providing basic or advanced services to the ISP's clients is more important; routing by itself only generates cost, no revenue.

discussion on the fact that ISPs are generally reluctant to outsource internal topology or routing policy information to a third party. We thus argue that this outsourcing can occur in a controllable fashion that satisfies both parties (contractor and client). Of course, reverse engineering of the internals of the ISP network, e.g., to uncover fate-sharing on intra-ISP links based on external observation of edge-to-edge paths, can be applied in all cases including the outsourcing one. This is not though a limitation of our approach which is based on a trusted third party, but rather a general methodology to externally infer internal ISP behavior—e.g., using causal inference [234]—independently from what we propose.

Furthermore, from the perspective of a client enterprise edge network—i.e., not a transit ISP—outsourcing can help offload the heavy tasks of routing (which is not the primary business of the client anyway) to a specialized party under an SLA contract. We note that AS-internal routing functionality could also be selectively outsourced using NOSes such as the ones discussed in Section 3.1; in this work, we are though more interested in the inter-domain routing interactions between the outsourcing clients and how centralization may help *on the AS level*.

Technically, the routing logic of the service contractor calculates the proper configuration of the control plane, updates the state of the network elements of the client AS, and deals with inter-domain routing with the rest of the Internet through BGP. In particular, as a client domain, we can choose to export the following information to the contractor using corresponding mechanisms to yield the aforementioned technical benefits. As we noted before, this information can be offered in part—as a specific abstracted view of our network—if we choose to follow the path of selective outsourcing to protect our privacy or for other reasons related to our domain's business logic.

Routing policies. These are policies of the client defined by the client AS administrators or derived based on requirements of SLAs between the client and other parties. They should be enforced and monitored by the contractor. Routing outsourcing does not impede other services offered by a client AS that may depend on routing. This is because the enforced policies are specified by the client AS during the consulting phase and either take the requirements of all offered services into account, or they operate within their own virtualized, isolated space that the contractor manages. In addition, the client may regularly

update its routing requirements and policies within a dynamic service environment; the dynamicity of these updates depends on the interface between the involved parties.

Network's state and monitoring data. The client exports selected topology, configuration, and measurement data, e.g., network utilization or bandwidth allocation. The contractor is a trusted third party that treats this data as well as routing policies that accompany the data confidentially. The model of a trusted third party, although it requires trust, has been very successful in practice for many modern services [166]. Also, SLAs can always specify the level of confidentiality and traffic visibility, while virtualization and slicing mechanisms implement the needed abstractions from a practical point of view [222].

eBGP sessions. The contractor handles the eBGP sessions and routing interactions between the client and other ASes. BGP messages can be redirected from the border gateways of the client AS to the contractor's routing control platform and vice versa. For internal dissemination of this information to the flow tables of the routers and switches, iBGP [45] or OpenFlow [183] may be employed.

Financial Benefits and Associated Observations

In the following, we elaborate on some qualitative observations regarding the economic benefits of routing outsourcing. Operators have traditionally viewed the network as their core business. However, declining profit margins have put them under pressure to reduce costs and to launch new, higher margin services. This situation has also pushed operators to streamline their operating expenses (OPEX) and has given rise to an emerging market of managed services, in which the operation and maintenance of the network is outsourced to a third party. In this context, we propose a new approach for network outsourcing, i.e., routing outsourcing, which enables the logical centralization of the routing control plane beyond AS boundaries. Financially, the contractor enjoys an opportunity for an economy of scale as the basic principles of routing optimization are the same across different networks. Economies of scale have been prolific in many computing contexts [166]. We believe that this also holds for routing management; we do not *claim* that this statement holds *always* in a heterogeneous Internet-scale environment, but rather that there are cases where outsourcing is reasonable as a practice. For example, outsourcing can reduce network-related OPEX

for the client, via streamlining. Moreover, outsourcing a low-margin service enables more effective use of human resources on higher priority services, which is the main goal of an ISP or an enterprise network. Finally, the price of outsourcing vs insourcing a certain network function, such as routing, should be also treated under the light of the benefit (added value); for example, the capability of the RaaS contractor for global optimizations may be worth the investment in such a practice. A particular example is the scenario of congestion between two ISPs; outsourcing their interactions to a centralized mediator may help resolve the problem.

Finally, we note that related privacy concerns are not investigated here at length; we rather discuss them within the next three chapters of the thesis in the context of the related use case (for example, see Section 6.2.3).

3.3.3 Challenges

The transition from current network setups to outsourcing-enabled environments is a multi-stage process; one challenge is the capability to backtrack or change to a better-suited contractor during this process. Therefore, in each of these stages, the client should be able to scrutinize the effects of the changes performed, both in terms of traffic management and expenses, and step back in case it is not satisfied. In Chapter 4 we will describe a framework where the client is abstracted as a big switch/router and outsources the management of inter-domain interactions to the contractor's controller. The client can then monitor how much this benefits the convergence times of inter-domain routing, and decide how satisfied it is with the resulting situation. A similar evaluation will be presented in Chapter 6 in the context of end-to-end path performance across domains.

On another note, policy conflicts between ASes can lead to tussles, which can create problematic paths or even depeering events. The bird's eye view enables the contractor to efficiently detect tussles [67] between its clients. The job of the contractor is to allow the tussles to unfold and run as today, based on the choices and policy requirements of each client. The main difference with the status quo is that the contractor can easily detect and mediate the resolution of routing problems, like dispute wheels [113], which may stem from these tussles. In addition,

the bird's eye view of the contractor can help find better solutions that meet the policies of each AS than when ASes act alone based on their limited local view. Moreover, new tussles between outsourcing service contractors are introduced. As the contractors start competing for clients, additional tussle dimensions arise, thus enabling a new game between the outsourcing entities; this game can be further investigated (see future work at Section 7.4). As a first step, in this thesis in Chapter 5 we develop a graph transformation algorithm which can be applied in a variety of contexts, including a policy-based multi-domain setup. A logically centralized contractor can run classic flow algorithms on the transformed graph to determine, e.g., the path diversity and bisection bandwidth obeying the policy choices of its clients on the inter-domain level.

3.3.4 Outline of the Architecture and Simplifications

Here we elaborate on a possible outline of the outsourcing-based architectural approach that we advocate, based on a 2-layer hierarchy: a high-level multi-domain routing control platform and a per-domain delegate controller to aid scalability, redundancy and to offer the required operational abstractions of the domain network. The outline is given in Figure 3.1. The aim of this section is not to present all the functional details of a RaaS controller architecture; we rather explain the high-level architecture, point out the most important involved components and interactions and describe how we apply its principles in this thesis[3].

First, in order to remain legacy-compatible, a routing outsourcing contractor may use legacy control sessions to directly access the Forwarding Information Bases (FIBs) and Routing Information Bases (RIBs) of the network components of each served domain. This access may be based on a partial, virtualized view, e.g., of the sliced flow tables of OpenFlow switches [224]. In particular, the routing control platform of the contractor can leverage iBGP sessions or Command Line Interfaces (CLIs) to control legacy components in the context of SDN as described by Caesar et al. [45]. Legacy APIs and high-level

[3]For example, we will describe a concrete implementation of the multi-domain routing controller in Chapter 4 in detail, based on a popular OpenFlow controller. Examples of an abstracted inter-domain data plane and the high-level control logic which governs it will be presented in Chapter 6.

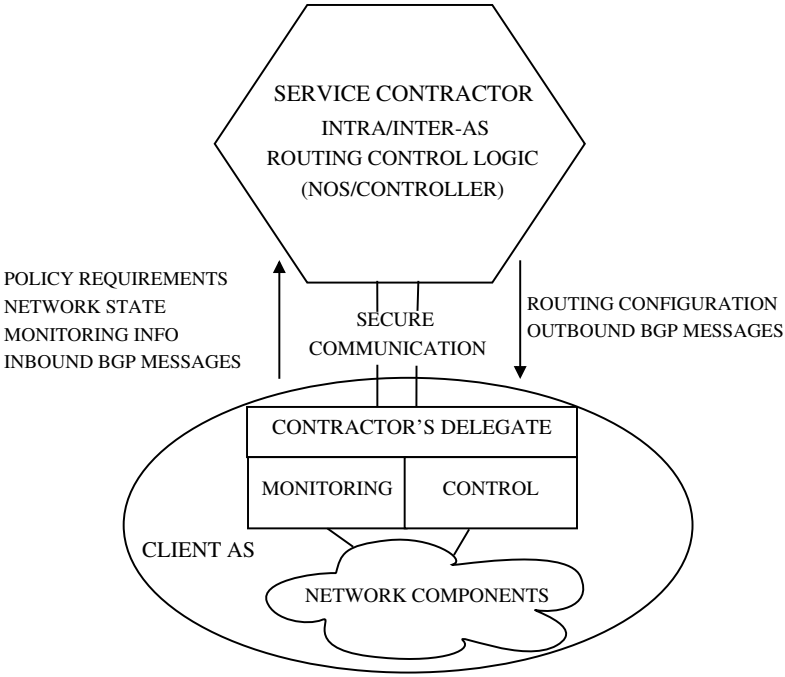


Figure 3.1: Outline of the architectural approach based on a 2-level control hierarchy. A RaaS contractor is responsible for parts of the intra-domain and inter-domain routing logic, while it can use its per-domain delegate for domain-local computations and control and monitoring management. Moreover, the delegate offers the needed abstractions of the client network to the high level control logic of the contractor. Secure tunnels may be used for the communication with the delegate, while the contractor can communicate over BGP with legacy ASes and over BGP or new protocols with other contractors (not shown in the figure for simplicity). The focus of this thesis is the multi-AS routing controller, implementing the RaaS contractor's logic, and the abstracted inter-domain data plane on which it operates.

SDN emulation [215] can help provide interoperability within hybrid networks which are gradually adopting OpenFlow. The management of the complexity of this process is the contractor's task. Compatibility

with the rest of the (legacy) world is guaranteed by using the BGP protocol and API, i.e., by accepting and generating BGP messages and maintaining eBGP sessions with external domain routers.

The logically centralized multi-AS routing control platform we propose can be physically distributed or centralized. Clearly the location of the different components affects factors such as resiliency, delay, costs, privacy and transmission overhead. Although the exact functional details of the control platform are outside the scope of this chapter, we note that a potential element of the architecture that can aid with tuning these factors is a domain-local delegate controller as shown in Figure 3.1. This component is useful to: *(i)* mitigate the delay overhead between the network elements and the contractor's platform, *(ii)* avoid multi-hop control sessions over the Internet and therefore increase resiliency to disruptions, and *(iii)* offer certain network abstractions for state control and management to the contractor. The component is domain-specific and is tailored to the requirements of the client.

The delegate communicates directly both with the contractor's platform and the internal elements of the network. For the elements, it acts as the routing controller: it terminates all control sessions, like OpenFlow, Netconf, etc[4]. For the contractor, it acts as a proxy for the control and management of the network elements. Besides handling the diverse routing configuration channels, it delivers monitoring data, related to the client network's state and utilization statistics. It is also the end-point of redirected eBGP sessions and relays BGP messages between the contractor and BGP gateways. The delegate can generally host parts of the routing logic of the contractor, acting as its stepping stone inside the client's local infrastructure. Additionally, the contractor can host physically outsourced control and management components in a private cloud infrastructure. In this way, large-scale routing computations can benefit from cloud computing characteristics, i.e., reduced costs, elasticity, scalability, multiplicity, availability and reliability for computing, communication and state storage resources.

The contractor's platform and the delegate communicate using an appropriate API and a secure communication protocol, over which the

[4] The delegate, or proxy, can act as a set of "virtual datapaths", receiving OpenFlow commands sent by the contractor's controller, interpreting them, and applying the corresponding actions on the real switches using SNMP, WebService or CLIs. With this process, VLANs for example can be set up on the fly, emulating tunnels between OpenFlow-enabled switches over a legacy infrastructure.

diverse channels between the client and the contractor are multiplexed. Multiple secure tunnels traversing diverse paths between the parties can be used for resiliency. In summary, the layout of our scheme is composed of the AS infrastructure, a domain-local delegate controller and the remote routing control and management platform of the contractor. The scheme is shown in Figure 3.1. We note that this scheme is proposed as a possible outline of the centralization architecture; this was the original schematic that we used as the basis of the architectures described in Chapter 4 and Chapter 6. For simplicity, we will focus on the multi-domain controller entity and the domain abstractions in both cases and we leave the engineering associated with the 2-layer contractor-delegate architecture for future work. Therefore, we treat the delegate as the client-specific component offering suitable abstractions of the network domain to the RaaS contractor; this can be achieved in practice via virtualization techniques [11], as well as hierarchical setups such as ONIX [146]. Each domain can select the appropriate way to implement this contractor-facing abstraction layer [222].

Figure 3.2 shows a simplified schematic of how the architectural principles of Figure 3.1 are applied in practice for the rest of the thesis. In particular, an inter-domain controller communicates via an SDN interface for control and monitoring with each client's domain. Each domain is abstracted as a big switch/router[5], using the delegate controller, and outsources its interactions with other domains on its perimeter. This allows the study of inter-domain routing on the AS level with the internals of the domain abstracted away. Internally, the domain can be controlled and monitored by its own controller, while the delegate can interface with the data plane using any available—legacy or SDN—interface. High-level multi-domain applications, such as BGP convergence acceleration or end-to-end service provisioning (in parallel to BGP) run on top of the RaaS controller. This simplified approach can then be emulated (cf. Chapter 4), simulated (cf. Chapter 6) or be used for theoretical reasoning of logically centralized optimization algorithms (cf. Chapter 5).

[5]This abstraction is explained in detail in Chapter 4, while an associated IXP-based abstraction is described in Chapter 6.

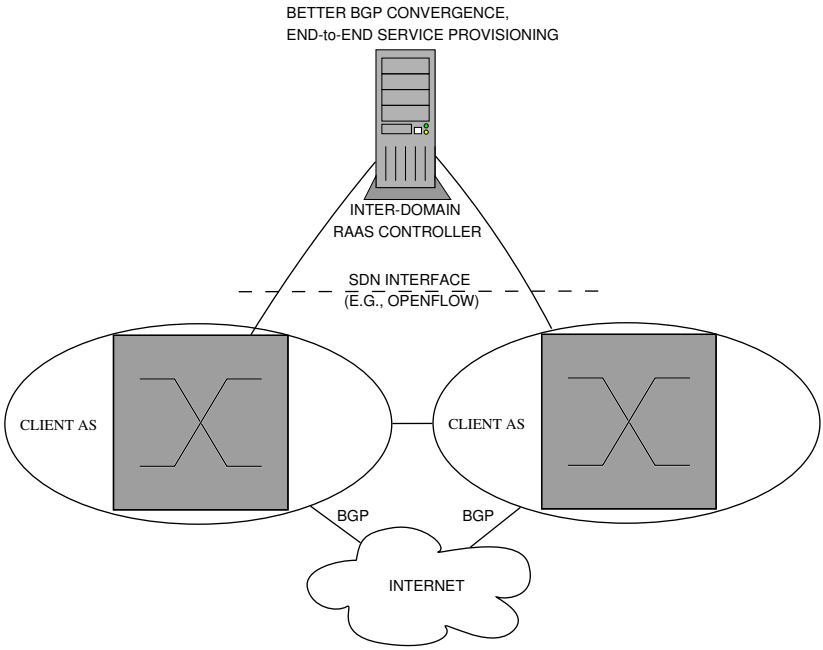


Figure 3.2: Simplified outline of how the architectural approach of Figure 3.1 is applied in this thesis. An inter-domain RaaS controller communicates via an SDN interface for control and monitoring with the client's domain. This domain is abstracted as a big switch/router, using a delegate controller (not shown on the figure), and outsources its interactions with other domains on its perimeter. Internally, the domain can be controlled and monitored by its own controller, while the delegate can interface with the data plane using any available—legacy or SDN—interface. High level applications, such as BGP convergence acceleration or end-to-end service provisioning run on top of the RaaS controller.

3.4 Summary of our Approach

In a nutshell, we propose a scheme where a contractor can control (parts of) the inter-domain routing logic of multiple ASes based on their policy requirements and network state. As more and more

ASes choose the same contractor, AS clusters are gradually formed. These clusters are the manifestation of gradual inter-domain routing centralization, and give us the footing to research the interplay between multi-domain SDN controllers and the rest of the Internet. Moreover, we envision multiple contractors competing for clients, and interfacing with each other over new APIs; a full overview of our proposal and the steps towards inter-domain routing centralization are shown in Figure 3.3. This figure summarizes the three primary steps of our approach; *(i)* centralizing and outsourcing the per-domain routing control plane to an external contractor (cf. Section 3.1), *(ii)* performing cumulative routing outsourcing in order to enable the provisioning of new multi-domain services (cf. Section 3.2), and *(iii)* scaling up the scheme to multiple contractors and client clusters that interoperate with each other (cf. Section 7.4).

Regarding financial aspects that go beyond the scope of this thesis, we note that the expenses of routing outsourcing vs insourcing should be weighed together with the added social welfare value that global optimizations on the contractor's side bring to the table. The bird's eye view for example may constitute important added value for potential RaaS clients, as a set of domains. Finally, we note that in this thesis we do not implement the entire proposed RaaS architecture, but focus on the high-level control, the inter-domain RaaS computations and the proper AS abstractions. In particular, we look at different instantiations—or slices—of the general architectural approach in the context of prominent use cases that are associated with inter-domain routing in the Internet.

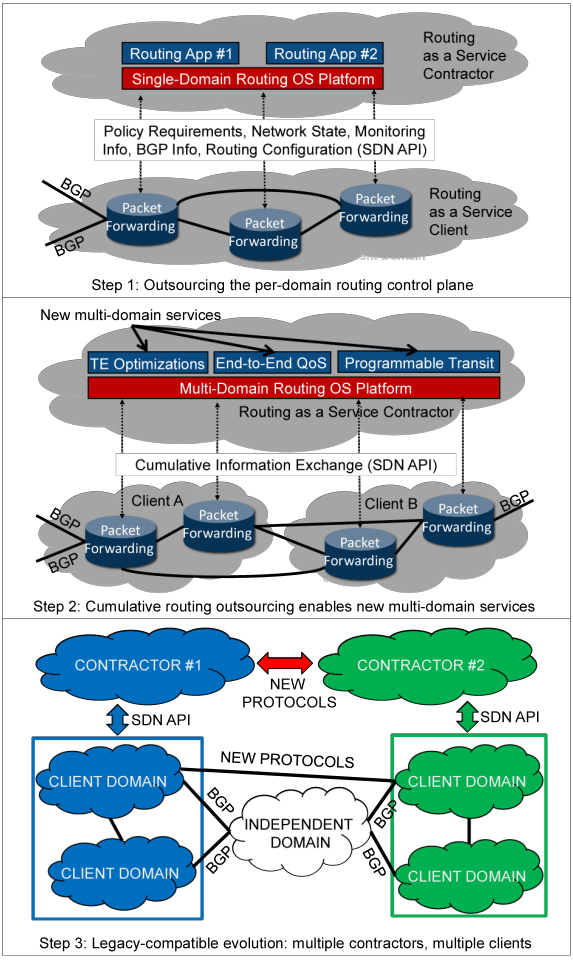


Figure 3.3: Overview of the general architectural approach and the steps towards scaling up to a hybrid multi-domain Internet ecosystem, with multiple RaaS contractors coexisting with the rest of the Internet.

Chapter 4

Use Case 1: Improving Convergence with BGP Compatibility

> "How will we get back up?" I worried. "I have a different route in mind for our return trip."
>
> — Karen Marie Moning, *Darkfever*

How would a SDN-based centralization scheme interact with classic BGP in terms of routing convergence, e.g., after a topological fail-over event? What is the impact of incremental deployment in this regard? In this chapter, we describe an implementation of the routing outsourcing architectural approach that we described in Chapter 3, based on the background study performed in Chapter 2. This is accompanied by an evaluation of an inter-domain routing convergence use case. We note that the main content of this chapter is based on our Elsevier journal publication in the Special Issue on SDN & NFV [151], while an initial implementation of the controller and framework that we use can be found at the SIGCOMM demo extended abstract presented by one of our collaborators [95].

4.1 Introducing the Controller and Framework

In this work, we elaborate on a logically centralized SDN controller and the associated emulation framework. The controller controls the routing computations on behalf of multiple domains, tailored to accelerating BGP convergence as a first use case. This is one use case and manifestation of the architectural outline sketched in Chapter 3, with the emphasis being placed on the global multi-domain controller rather than the domain-local delegate (cf. Figure 3.1, Figure 3.2). Each domain offers a virtualized view of itself—the "big switch/router" abstraction that we will analyze in detail later in this chapter—to the central multi-AS control logic. In particular, we make the following contributions related to the framework, the multi-AS controller and their use in terms of evaluating hybrid SDN-BGP interactions[1].

First, we develop a *publicly available emulation framework* for conducting *hybrid* BGP-SDN inter-domain routing experiments; this can be used in generic experiments also by other researchers. Second, we design and implement a *proof-of-concept multi-domain SDN controller* which controls AS clusters via OpenFlow (used as the southbound interface) and maintains legacy compatibility with BGP (used as the east-west interface). Insights on the development of such a controller are analyzed in detail, focusing on the implications of interacting over eBGP with legacy domains, while applying centralized path computation within the cluster. All BGP semantics (e.g., AS numbers) are maintained during and after the SDN-based clustering process. Finally, as a use case, we evaluate the *interplay between path-vector BGP and link-state SDN routing* in terms of convergence using the developed controller and framework. Our findings indicate that convergence times can be reduced with increasing SDN penetration in hybrid cross-domain networks, while churn rate reductions need relatively large SDN deployments to be tangible. The experimental results can be replicated by other researchers for verification purposes, as the software and scripts used are available for the community to use.

The rest of this chapter is structured as follows. Section 4.2 gives an overview of the hybrid BGP-SDN framework on which we run our

[1] These interactions are studied in the context of routing convergence times and update churn rates as a use case.

experiments. Section 4.3 describes the design goals and implementation insights gained during the development of a proof-of-concept inter-domain routing controller. Section 4.4 presents the results and insights acquired from the evaluation of routing convergence on synthetic hybrid BGP-SDN AS-level networks. In Section 4.5 we summarize the chapter's main elements and take-away messages.

4.2 A Hybrid BGP-SDN Emulation Framework

In this section we describe the design goals and requirements of the SIREN (Sdn-bgp Inter-domain Routing EmulatioN) framework, key features of its implementation and some examples on its potential use by researchers and experimenters. We note that the SIREN framework and controller were implemented in collaboration with Adrian Gämperli.

4.2.1 Design Goals of SIREN

The main objective of SIREN is to enable hybrid BGP-SDN routing experiments. In particular, SIREN was implemented with the following design goals in mind:

- Enable the emulation of BGP-SDN routing interactions, while making their evaluation simple and straightforward. The Routing-as-a-Service approach described in Chapter 3 was taken into account as an example use case that could be deployed over such a framework.
- Emulate multiple ASes, both from the legacy (BGP) and the new evolved (SDN) routing perspective.
- Use real router and switch software where possible in order to emulate the effects of production software suites.
- Simplify and automate experiment management, allowing researchers to focus more on the experiments rather than the tools and concentrate on the results and associated insights.
- Easily visualize and interpret the measurement results.

4.2.2 Implementation of SIREN

SIREN [96, 97] is a publicly-available, Python-based network emulation framework for conducting hybrid BGP-SDN experiments. It extends the Mininet [164] network emulator, which is a popular environment for running SDN experiments. Mininet—and consequently SIREN by inheritance—offer OS-level virtualization (based on Linux namespaces), which efficiently scales up to hundreds of emulated nodes and links, and comes bundled with the Open vSwitch [4] virtual OpenFlow switch. In SIREN, we combine Mininet with the popular Quagga [6] routing software[2], which implements BGP and also other routing protocols such as RIP and OSPF. For this work, we focused on Quagga's support for eBGP sessions between the border routers of ASes.

In Figure 4.1 we show the components of a sample SIREN setup, that corresponds to the routing outsourcing approach that we described in Chapter 3. On the left side, we see the legacy BGP part of the emulated multi-AS network—composed of BGP routers—whereas on the right side we illustrate a SDN cluster, composed of programmable OpenFlow switches. These switches are controlled by a logically centralized SDN controller. BGP routers and SDN switches can originate IPv4 prefixes (the latter entities via the controller's mediation). It is also possible to add hosts with IP addresses within a particular prefix for monitoring end-to-end connectivity with tools like *ping*, or end-to-end throughput with *iperf*. All BGP routers peer with a *BGP route collector/monitor*, which collects routing updates for monitoring purposes. Moreover, within the SDN cluster we have a special BGP speaker, called *cluster BGP speaker*, which relays routing information between external BGP routers and the SDN controller. This speaker is implemented using the ExaBGP Python library [1]. For every BGP peering there is a link from the speaker to the border SDN switch, in order to relay control plane information over the switches. The latter process effectively implements the "outsourcing" of eBGP sessions.

In SIREN, every AS is abstracted as one "big switch" (or router), i.e., it is emulated by a single network device/node. This abstraction is not fundamental to the framework—and can be extended in future versions—but is useful for use cases such as ours, for the following

[2]Quagga is regarded as one of the most used software for inter-domain routing on general purpose hardware, and its interplay with SDN controllers—over SDN switches—is what we study in practice while running hybrid BGP-SDN experiments.

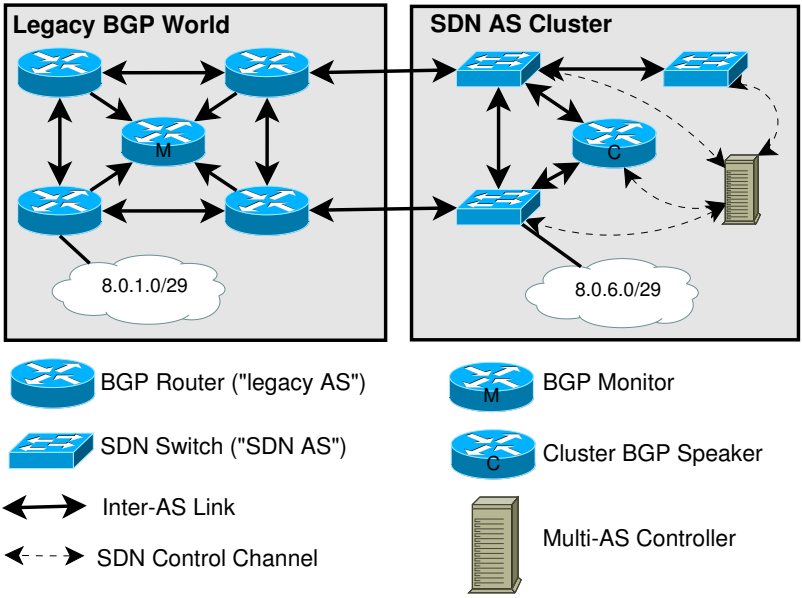


Figure 4.1: Sample network setup in SIREN: a multi-AS cluster, under the supervision of a logically centralized SDN controller, peers with the legacy BGP world. Inter-domain routing is affected by the interplay between classic BGP and the SDN RaaS controller.

reasons. First, we want to isolate the effect of inter-domain routing convergence and experimentally study its properties. Second, we assume that each AS is not willing to share its internal topology with the routing outsourcing contractor's controller for privacy reasons. Third, BGP is by design an information-hiding protocol: the legacy ASes appear on the routing update messages (e.g., AS path attribute) as abstracted single nodes. We note that this abstraction is aligned with our assumed simplification and is sufficient for a proof-of-concept implementation. We acknowledge of course that in reality an AS can be a complex interconnected fabric of end-hosts, switches, routers and middleboxes, but the management of the full spectrum of internal AS functionality is out of the scope of this work. For these reasons, the "big-switch/router" view is suitable for inter-domain routing management assuming of course

that the AS is consistent regarding its interactions with other domains at different peering points. This appears to be true in practice according to works such as the survey of Gill et al. [102]. In SIREN, a legacy AS is modeled by a Quagga BGP router, while a SDN cluster AS is modeled by an OpenFlow switch. A cluster AS can use non-SDN mechanisms for internal routing; this does not hinder the view of the AS as a centrally controlled switch/router on the inter-domain level. The proper abstraction can be provided by a local RaaS delegate controller, as we discussed in Chapter 3.

4.2.3 Experimenting with SIREN

SIREN experimental setups can be written in Python. The framework automatically assigns IP addresses and configures network devices using pre-defined templates. We extended Mininet with several BGP-specific commands to announce prefixes, wait until BGP has converged, etc. Additionally, the framework supports tools for automatic log file analysis, network graph creation, measurement of convergence time, routing update churn rates and packet loss, and route change visualization. For example, to facilitate experiments on routing stability, the framework detects when the network has converged and whether there is stable connectivity between all hosts. Other compatible tools can be added as Mininet is an extensible platform. Also, experiment batches can be distributed over multiple computing nodes using an *experiment manager*[3]. An example of live routing visualization is presented in Figure 4.2.

Generally, SIREN can be used in the following modes: *(i) Command Line Interface (CLI)*, like in Mininet, where the experimenter conducts classic terminal scripting. The main premise here is that the experimenter writes her Python experiment setup scripts and invokes them, while interacting with the framework via the CLI; in the end results are collected and interpreted. This mode is indicated for experiments of average scale. *(ii) Live Visualization* as described beforehand, where the experimenter visually interacts with the network while she can view routing-related info (e.g., convergence time) after her induced changes. This mode is indicated for experiments of small scale

[3]We note that while this entity is useful for managing multiple small to medium scale experiments (10s of nodes and 100s of links) across a cluster, there are hard scalability limits with respect to single larger scale experiments.

and for visual demonstrations. *(iii) Experiment Manager* as described beforehand, where the experimenter distributes batches of experiments to multiple nodes and exploits parallelism, e.g., in cloud environments. This mode is indicated for distributing large batches of small/average-scale experiments that lie within the scaling capabilities of the framework. More details on SIREN's implementation can be found at our published SIGCOMM demo extended abstract [97] and the MSc thesis of the student whom we advised and who co-authored the software [95]. For

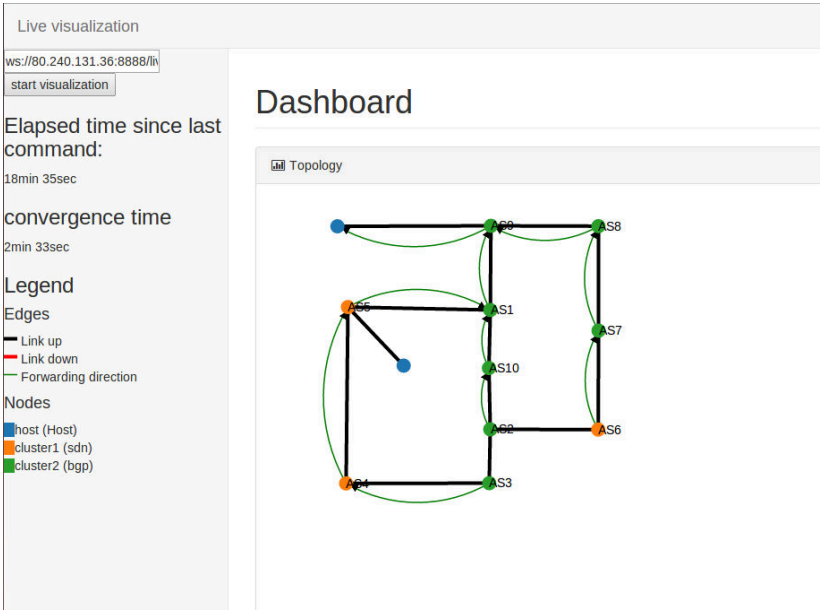


Figure 4.2: Live visualization in SIREN: forwarding is tracked towards the depicted hosts based on the routing configuration of the ASes on the end-to-end path. The user can visually interact with SIREN via bringing inter-AS links up or down, actively creating convergence triggers and monitoring the network's response. Green nodes correspond to legacy BGP ASes, while orange nodes are parts of a (disjoint) multi-AS SDN cluster. Traffic (green arrows) flows from all ASes towards the north-west end-host (blue node) over inter-AS links (black links). The other end-host is used for verifying end-to-end connectivity with ping.

completeness, we also note that MiniNext [220] is another hybrid SDN-legacy routing emulator based on Mininet and Quagga. However, while MiniNext aims at emulating operational environments and focuses on low-level APIs, our framework focuses on multi-AS inter-domain experiments for research and provides a high-level API for experiment orchestration tailored to such use cases.

4.3 A Proof-of-Concept Multi-AS Routing Controller

In this section we describe the design and implementation of a first multi-AS routing controller, tailored to improving BGP's slow convergence. We make the controller public as part of the SIREN software [96]. Our goal is not to build a general-purpose multi-AS NOS; addressing the full set of challenges in building such a generic system is beyond the scope of this work. In contrast, our objective is to improve BGP's problematic convergence as a proof-of-concept use case on which our inter-domain SDN routing approach may be applied. We use the controller to evaluate the interplay between centrally controlled routing within a SDN-enabled AS cluster and distributed path-vector BGP routing outside of the cluster in Section 4.4, focusing on convergence properties, such as total convergence times or routing update churn rates.

4.3.1 Design Goals of Multi-AS RaaS Controllers

The design goals and objectives of the multi-AS routing controller implementation are analyzed in detail in the following.

Logically centralize routing computations. We wish to benefit from logical routing centralization on the AS level in order to improve BGP's convergence time and reduce routing churn, leading to more stable routing overall. The intention is to help both ASes within the cluster and the outside (legacy/BGP) world.

Interoperation with BGP. As BGP is currently deployed across the globe enabling ∼50,000 ASes to mutually exchange routing information, it is crucial that the controller is fully compatible with the BGP standard [212]. In addition, the SDN cluster should be transparent to the outside world, i.e., legacy BGP AS border routers should treat

the cluster's BGP speaker as yet another BGP router of one of the cluster ASes, rather than part of a multi-AS SDN controller setup.

No cluster lock-in. The identity of the participating ASes (e.g., their AS numbers) should be preserved in the hybrid routing system. This prevents cluster lock-in, which could result if the AS numbers of the served cluster ASes were replaced with a "super-domain" identifier, being effectively aggregated. Instead of this, we wish to have groups of ASes which are visible as separate entities to the rest of the Internet and maintain their individual identities and policies. This approach also facilitates a smooth transition to the new system as existing mechanisms relying on AS numbers (e.g., access lists or BGP communities) do not need to be updated. Maintaining BGP semantics is yet another provision to make the AS cluster transparent to BGP and avoid a disruptive expansion of the sphere of influence of multi-domain Routing-as-a-Service controllers.

Disjoint clusters. As the transition to the new routing architecture will likely be gradual, clusters will probably not be contiguous, at least in the beginning. This means that AS paths may enter, exit and re-enter the cluster(s) at different points (e.g., IXP-facing PoPs); thus the controller should be able to calculate paths using the global view of a disjoint cluster of clients and the legacy BGP information that it learns through them. This also means that in case the cluster is internally partitioned due to an inter-domain link failure, it may not be partitioned on the global level since paths that join the two parts over legacy ASes can still be used. Thus reachability over disjoint clusters should be achieved in any case. Of course, operating on a group of dispersed client ASes may prevent other optimizations that can only happen over a contiguous controlled cluster, such as guaranteed end-to-end circuits (cf. Chapter 6). On the other hand, classic IP connectivity can be established as long as there are BGP paths—passing over external ASes—that enable this, while the convergence benefits can be exhibited in both the contiguous and disjoint cluster use cases (cf. randomized cluster formations in the evaluation of Section 4.4). Slow convergence (over external ASes) between the clusters might still be an issue is some cases, but the global view of the controller accelerates the process where applicable.

Hybrid routing. The controller knows the full inter-domain topology within the cluster, using southbound SDN-based interfaces

to acquire and control the (abstracted) view of the client ASes. Additionally, it receives external AS path announcements from the outside world via BGP; this communication takes place over the border routers of the cluster ASes. Therefore, inter-domain routing becomes hybrid path-vector and link-state [13] as follows. The controller can use an algorithm such as Dijkstra in order to calculate shortest paths over the known cluster topology. External AS paths learned from BGP can be attached as "extensions" to the cluster graph and be explored with Dijkstra. Selected paths can be then advertised to legacy BGP peers, making the controller a part of the outside BGP path-vector system and enabling the co-existence of both link-state and path-vector routing on the global inter-domain level. This hybrid routing and SDN-BGP interactions constitute a primary design objective of our implementation.

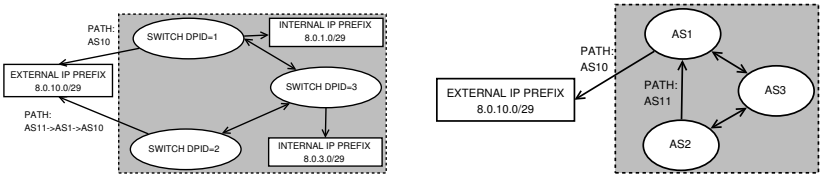
No loops. AS-level loop avoidance is essential in the new hybrid BGP-SDN setup both for routing efficiency and correctness. We note that naively using the same loop avoidance mechanism as BGP is not wise, as we will show later, due to the differences between BGP's distributed process, based on local per-AS view, and SDN's logically centralized process, based on the global multi-AS view.

4.3.2 Implementation of Multi-AS RaaS Controllers

The controller runs using the POX controller [5] mechanisms for OpenFlow-based interaction with the cluster switches, while it interfaces with external BGP routers through ExaBGP [1]. In particular, the controller speaks OpenFlow in the southbound direction within the cluster, and BGP outside of the cluster in the east-west direction. Applications, such as AS-level shortest-path routing, run as software modules on top of the controller using its northbound interface. POX-like cooperative multitasking is used for the event-based processing that happens on the controller. This approach is well-suited for rapid prototyping, therefore we can focus more on research questions pertaining to our proof-of-concept use case rather than deal with state consistency, scalability and concurrency issues [146]. To better understand the operation of the implemented path selection algorithm, we first introduce two graphs which represent the core state that the controller maintains.

Switch Graph. The *Switch Graph* is a simple directed graph that represents the physical topology of the multi-AS cluster combined with prefix connectivity information, as seen from the controller's perspective. This assumes that the participating ASes are willing to expose such information in order to benefit from the reduced convergence times and more efficient routing control. If they are not, they are simply treated as external entities that are not parts of the cluster. We have two kinds of nodes: switch nodes, which represent SDN switches, and prefix nodes. The presence of an edge means that data can be forwarded from the source to the destination of the edge (switch-to-prefix or switch-to-switch). The *Switch Graph* is built gradually: we add a directed edge between two switch nodes, when a switch node (and consequently its controller) detects a link in that direction. An edge from a switch to a prefix node is added, when the prefix is learned from BGP or the prefix is directly connected to that particular SDN switch. In the "BGP-learned edge" case, we only add the best path in terms of hop count and annotate the edge with the corresponding AS sequence. We save all paths that the cluster receives information about; the best (i.e., shortest) paths are then selected for the eventual routing of traffic across the ASes. The graph is properly updated when inter-domain links fail/come up and/or paths to learned prefixes are withdrawn/announced.

AS Graph and Loop Avoidance. In our hybrid link-state/path-vector setting, we need to cater for AS-level paths that leave and re-enter the cluster, after passing over legacy ASes. As we have already mentioned, this case is possible and should be taken into account. If such paths were naively marked as annotations to external prefixes as in the *Switch Graph* and used directly by Dijkstra, then we could get loops due to the multiple occurrences of the same cluster AS in the path calculations (abstracted as another hop count in the annotation). For this purpose, we break such paths into two parts: *(i)* a destination prefix attached to the last cluster AS in the path, and *(ii)* virtual links that connect cluster ASes over external paths. We incorporate these changes into a per-prefix *AS Graph* structure, which is a transformation of the *Switch Graph*. At the beginning of the transformation, all AS numbers of the cluster are added as nodes. The AS connections inside the cluster, which have been represented as edges between switch nodes in the *Switch Graph*, are also added to the new graph. This provides an AS-level view of the cluster to the controller. The transformation is restructuring the *Switch Graph* taking into account paths that cross the



(a) Example of a Switch Graph. Switches 1, 2 and 3 form a cluster. Switches 1 and 2 know a path to prefix 8.0.10.0/29, which they learned over BGP; these paths pass over external ASes. Note that the path known to switch 2 passes over switch 1 as an intermediate node. Switch 1 has a directly connected prefix, and so does switch 3.

(b) Example of an AS Graph. The switches have been transformed to their corresponding ASes. Note that the path known to switch 2 has been sanitized; we have added a virtual link that includes the external AS path that exits and re-enters the cluster (in this case over AS 11), ending at AS 1. AS 1 then knows the best path towards the prefix.

Figure 4.3: Example of Switch-to-AS graph transformation: paths to 8.0.10.0/29 are sanitized to avoid loops.

legacy world and leave and enter the SDN cluster at different points in order to avoid loops. For example, Figure 4.3a shows a sample *Switch Graph*. Figure 4.3b shows the derived *AS graph*: the switches have been transformed to their corresponding ASes, while paths that exit and re-enter the cluster including legacy ASes have been sanitized for loop avoidance. This process guarantees that if BGP does not induce loops until the first encountered cluster AS on re-entrance, then shortest path routing on the controller's side will result in AS-level loop-free paths.

Main Algorithm. Dijkstra's algorithm runs on the AS topology graph using AS path lengths as weights. This allows the controller to calculate the shortest paths towards each prefix learned either internally from the cluster ASes or externally over BGP; paths to external prefixes exiting and re-entering the cluster are sanitized for loop avoidance as explained before. While for our practical use case we implemented classic AS-level shortest-path routing, in theory we can apply any routing algorithm operating on the global view of the multi-AS controller. Examples include but are not limited to: multipath routing [257], mixed routing protocols [232], new WAN-based DDoS link-flooding mitigation techniques [105, 106], etc. Furthermore, algorithms that operate on the

global view of the cluster, taking into account the AS-to-AS policies for the calculation of AS-level path diversity or bisection bandwidth, might also be employed as we will explain in Chapter 5.

The Path Recomputation Problem. AS paths are only recomputed by the controller's logic when needed. A link change between switch nodes or a switch change in the *Switch Graph* results in a full recomputation for the paths associated with all prefixes currently known in the network. However, when only a path to a certain prefix is changed, e.g., when a second switch adds a new path to a specific prefix (learned over BGP), only paths leading to that single prefix will be recomputed. At this point we should note the following insight gained during the implementation process. The SDN cluster controller can receive multiple BGP updates per second stemming from external ASes; these are essentially the external updates that are directed to all the members of its cluster[4], since it controls the inter-domain routing interactions on behalf of several ASes, each one with multiple external peers. Each of these updates triggers changes in the switch and AS graphs, causing path recomputation throughout the cluster. We need to stress that this is an expensive process; path recomputation is equivalent to the underlying switch reconfiguration through manipulation of the—virtualized—flow tables. Installing all necessary rules on the associated switches can take 100s of milliseconds[5], while the problem becomes more intense if we consider the need for consistent state updates [211]. During this convergence period more BGP updates are received, stressing the process even further. Moreover, the controller's actions need to be advertised to external peers; that means that besides the traffic shifts caused inside the cluster due to the flow rule installation process, the instability will also propagate further outside of the cluster and cause further problems. This is an artifact of using a logically centralized process to control the whole AS cluster; however, there is a practical work-around for this scalability issue that we describe in the following.

Delayed Path Recomputation. To mitigate the aforementioned issue we added a simple mechanism for delayed recomputation of paths,

[4]Besides, we can have SDN link or switch change events related to the AS cluster topology; these events through occur orders of magnitude less frequently than BGP updates received from the cluster's borders.

[5]This is due to the latency between switch and controller, plus the processing time needed within the switch. The latter delay is exacerbated under conditions of heavy load in the control plane, while the data plane reaches eventual consistency.

based on a timeout value called Cluster Waiting Recomputation Interval (CRWI)[6]. This is different than the MRAI advertisement interval of BGP. After the CRWI timeout happens, the controller computes and installs locally the rules associated with the new paths via OpenFlow[7]. These paths are the result of queued recomputation requests, accumulated over the waiting interval; after the final routing changes are implemented, the controller then directly advertises the changes over BGP to the outside world. This strategy helps the RaaS controller avoid routing inconsistencies with neighbours due to outdated information, since the queued requests are sanitized in terms of age (i.e., the oldest ones are automatically deprecated). Of course, if there is a persistent oscillation of an advertised prefix instead of a usual convergence process that eventually ends, this strategy simply slows down the oscillation; in such cases further strategies like route flap damping might be needed to mitigate the problem. Secondly, our proposed strategy can help make the network more stable by "rate-limiting" the cluster controller, reducing the number of required path changes and leaving some temporal slack for the forwarding rules to be installed on the cluster switches. In our experiments, we found that a CRWI of 1 sec is sufficient to avoid any problems with routing inconsistencies and flow rule installation delays. However, there is not an ideal value for every possible topology and routing setup. On the contrary, the CRWI value should be experimentally determined in terms of its interaction with the BGP MRAI timer and other timing mechanisms and artifacts of BGP routers, SDN switches and SDN controllers. Similar issues are present with the current situation; there is not an ideal MRAI value for all possible BGP-based setups, but its tuning is rather a trial-and-error process.

Further Implementation Details. Besides the important implementation features mentioned above, the controller also has partial support for consistent state updates [211] during the reconfiguration of the cluster switches. In particular, the controller proactively installs the routes from the destination to the source switch in order to give some time to the routing flow rules to be properly installed in the reverse order, before forwarding traffic along the direct path. However, the controller does not wait until the SDN switch confirms (e.g., via OpenFlow barrier messages) each rule installation as we faced timing

[6] In practice, this timeout is used for "throttling" the controller.

[7] In general, any suitable routing/flow table manipulation mechanism offered during the routing outsourcing process may be employed (cf. Chapter 3).

issues; the inverse order installation is simply a compromise between state consistency and timing efficiency. Without this ordering we might get a possible policy violation, unintended paths or loops as transient artifacts with a higher probability. Furthermore, proxied control traffic (BGP) and direct data (ARP, IP) traffic are both handled via flow rules on the flow tables of the switches. Proposals on how to scale the state management to hundreds of thousands of IP prefixes on the controller's side can be found at Section 7.4.1, as steps of future work. More details on the implementation of the multi-AS controller (e.g., the cluster-wide topology detection mechanisms) can be found at the associated works [95, 97].

4.4 Evaluation of Hybrid BGP-SDN Routing Convergence

In this section, we deal with the evaluation of routing convergence within a hybrid BGP-SDN environment. We start with the description of our experimental setup (cf. Section 4.4.1), including the basic setup, some needed simplifications and assumptions pertaining to the emulation, and the AS-level topologies that we used. We continue with the results that we got on convergence times and churn rates (cf. Section 4.4.2), and we close the section with some interesting observations and insights on the yielded results (cf. Section 4.4.3).

4.4.1 Experimental Setup

Basic Setup

What is the effect of inter-domain SDN centralization on BGP convergence time and stability? As a proof of concept of our proposed approach, we evaluate the effect of SDN centralization using our multi-AS controller (cf. Section 4.3) running on top of the SIREN framework (cf. Section 4.2). We assume the following setup (cf. Figure 4.4).

In our experiments, a dual-homed AS loses its primary upstream connection and performs a fail-over to its backup link. Its two providers are selected uniformly at random from a core inter-network of ISPs connected to each other in different topologies and formations. To enforce the primary-backup setup, the client AS prepends its AS

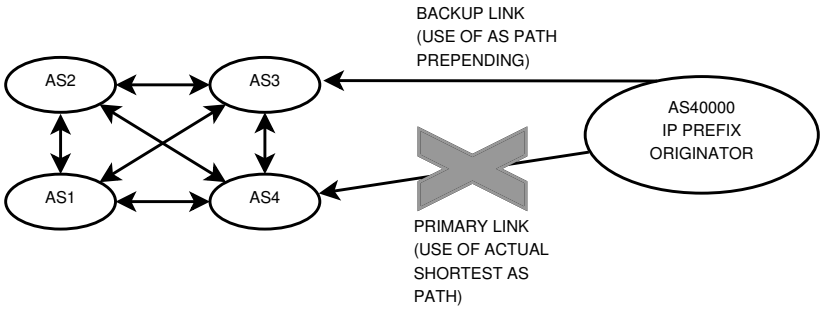


Figure 4.4: Link fail-over experiment used in the evaluation: the primary (preferred) inter-AS link goes down, and the path-prepended backup link takes over. All ISP ASes then recalculate the shortest path to the client AS 40000 (originating an IP prefix), while involved in a hybrid BGP-SDN path exploration process.

number multiple times in its prefix announcements propagated over the backup link. The link-down event on the primary link causes a wave of withdrawals throughout the network, accompanied by announcements of new—but not always valid—paths due to the path exploration process. The ISPs explore alternative paths to the client AS, taking into account the prepended route advertisements going over the backup—now active—link, as they converge to the shortest path. In this setting, we evaluate how gradual SDN penetration, in terms of increasing percentage of cluster SDN ASes, affects the convergence time and the average routing update churn rates. Concerning BGP timing artifacts, we note that we are using fast keep-alive and hold-down timers for both Quagga and ExaBGP, since we want to explore what happens after the link-down detection and not waste time discovering that the link is in fact down[8]. The values are selected empirically to be neither too large or too low, in order to avoid unnecessary delays or synchronization side effects, respectively.

[8]Essentially we focus on the path exploration phase rather than the topological failure detection phase. In practice, we bring the interfaces in both sides of the links down; therefore a Quagga router knows directly that a link is down, while the SDN-ExaBGP controller figures it out after at most 15 sec due to the decoupling between the controlled OpenFlow ports and the BGP session running over the ports.

Simplifications

We make the following simplifications in order to fit our use case and the properties of the data we have at our disposal (i.e., AS-level graphs). First, we assume one node per SDN-controlled or legacy AS; we refer to this simplification as the "big switch/router" approach as already described beforehand in SIREN's implementation. We understand that this is an important simplification as mentioned in the work of Willinger et al. [251], but it allows us to capture some basic properties of hybrid inter-domain routing even without knowing how an AS is structured internally. Also, according to the survey from Gill *et al* [102], ASes are usually consistent regarding their routing information export across their distributed fabric; therefore the abstraction makes sense from BGP's point of view and is aligned with the status quo. Moreover, the convergence times on the level of eBGP sessions between ASes are several orders of magnitude larger than the convergence times within an AS in terms of the iBGP and IGP (OSPF, IS-IS) protocols [160]. Thus, neglecting internal routing convergence times does not have any significant impact on our results on the inter-domain level. However, some intra-domain routing dynamics may propagate on the inter-domain level and affect global routing [237]; we plan to examine those cases as future work (cf. Section 7.4.1).

Second, in terms of policy-based routing, the controller calculates shortest paths based on classic Dijkstra-based algorithms. In BGP terminology, we are taking into account the AS path length as the primary metric, with the AS policy being "prefer the shortest path to reach a destination prefix". Besides AS path lengths, metrics such as congestion state or path latencies may be taken into account on the multi-AS SDN controller's side for the path calculations; we opted for path lengths in order to have a homogeneous routing behavior within the hybrid BGP-SDN core. We note that while there are ways of running centralized Dijkstra to find policy-compliant shortest paths (cf. Section 5.6.4 and the work of Bauer et al. [27]), obeying the valley-free conditions by Gao et al. [98], inter-domain policies are in reality much richer and more diverse than that. The main challenge that all researchers face in this context is that the policies are quite difficult to infer and are by nature commercial secrets of the ISPs; this results in lack of data regarding what policies ISPs implement, leaving only qualitative surveys [47] or surveys on small AS/ISP set samples [102]

to extract information from. Therefore, to simplify our experiments, we chose to explore shortest-path dynamics ignoring complex policies; future work may benefit from evolutions in policy inference to study richer BGP-SDN interactions on the control plane, policy-wise. The algorithm that we present in Chapter 5 is one step towards quantifying the effect of generic policies on the properties of such networked setups.

Emulated Topologies

Regarding AS-level topology emulation, we initially considered the CAIDA *IPv4 Routed /24 AS Links* dataset [50], providing snapshots of AS links derived from IP-level topology measurements. Due to the large size of the dataset in terms of AS nodes and links (∼tens of thousands), which goes way beyond the scalability limits of Mininet and consequently the SIREN framework[9], we did not run experiments on these graphs but used synthetic topology models instead, as we explain next.

According to the seminal work of Willinger et al. [250, 251], there is not yet a widely accepted model of the AS-level Internet topology; such inference requires a cumbersome reverse engineering approach based on domain-specific knowledge. Therefore, we took multiple different models into account [250]: cliques (full meshes), random graphs such as Erdos-Renyi (E-R), scale-free graphs of the preferential attachment type based on the Barabasi-Albert model (B-A), and small-world graphs using the Newman-Watts-Strogatz (N-W-S) approach. We then searched if common patterns were replicated across different graph types and scales, indicating interesting BGP-SDN interactions. We used the NetworkX graph generator [191] and selected its parameters such that the derived graphs achieve a compromise between fully connected ISP meshes, and sparse tiered environments. This can be observed in the node degree distributions of the studied graphs, as shown in Figure 4.5. Larger

[9] Our goal is to emulate a proof of concept rather than the entire AS-level Internet. Mininet can in practice support concurrent emulation of at most 150-200 switches and a number of links at the same order of magnitude; while the POX controller itself can sufficiently handle loads of tens of switches and about a hundred links. Down-scaling the original IPv4 AS link topologies based on such datasets [53] can partially mitigate the problem by achieving topologies reduced to 30% of the original size, but still the scale is prohibitively large to emulate. Sampling such topologies to reduce the size further distorts the properties of the graphs in question, something that we did not want to affect our experiments. Further up-scaling of our framework capabilities is the subject of future work (cf. Section 7.4.1).

Setup Parameters	Values
Experiment Type	route fail-over for dual-homed client
Topology Type	Clique (full mesh), Erdos-Renyi (E-R), Barabasi-Albert (B-A), Newman-Watts-Strogatz (N-W-S)
Topology Size [number of nodes]	8, 16, 32
Topology Size [number of links]	function(node_number, graph_type)
Clusters and Controllers [number]	one controller, one (contiguous/disjoint) cluster
SDN Penetration [%]	0, 25, 50, 75
SDN Cluster CRWI [sec]	1
BGP MRAI on Quagga [sec]	0, 30
Keep-alive timer (Quagga, ExaBGP) [sec]	5
Hold-down timer (Quagga, ExaBGP) [sec]	15
Reconnect timer (Quagga, ExaBGP) [sec]	5
Policy for BGP ASes, SDN ASes	prefer shortest AS path (hop count)
Client Policy for backup link	use 10-fold ASN path prepending

Table 4.1: Parameters used for the hybrid BGP-SDN inter-domain routing convergence experiments, and their respective values.

parameter values are closer to the first setup, while smaller values to the latter. The full set of experimental parameters and values explored is presented in Table 4.1. Note that for parameters such as the CRWI and BGP timers we select single values; these values were selected after extensive testing as the ones that have the minimum effect on the results. In practice, we selected the values with the lowest sensitivity in order to isolate the effects of other, more interesting parameters on the results. For example, a CRWI of 1 sec was ideal for our setup for different scales and graph types; smaller CRWIs lead to routing instabilities stemming from the controller's side while larger CRWIs significantly delay the whole convergence process without any practical benefits. Further study of the interaction between CRWI and MRAI timers is the subject of future work. The parameterized code and scripts, together with instructions on how to use the SIREN framework, are publicly available at the SIREN Bitbucket code repository [96].

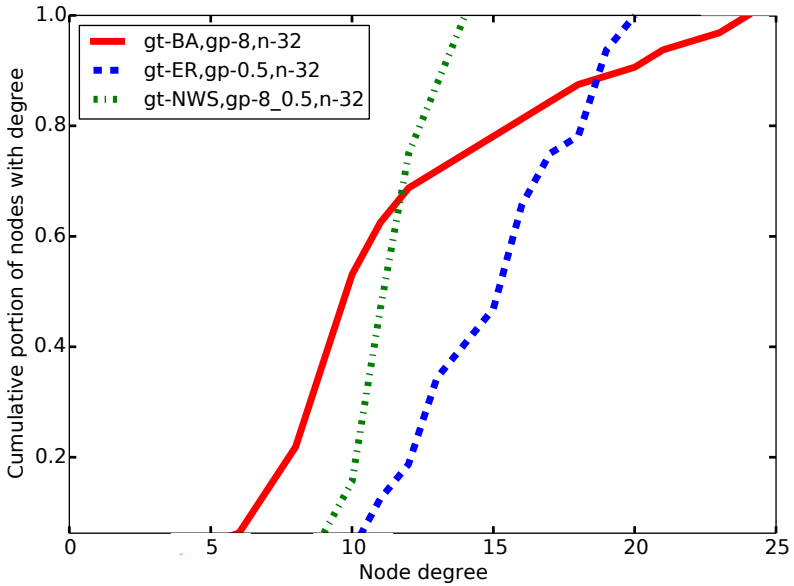


Figure 4.5: Node degree distribution for the 32-node graphs studied in the evaluation. Abbreviations: "gt"=graph type, "gp"=graph parameters, "n"=size of graph, "BA"=Barabasi Albert, "ER"=Erdos Renyi, "NWS'=Newman-Watts-Strogatz. The clique is omitted since all its nodes have a 31 node degree (full mesh). While NWS is quite homogeneous in terms of node degree, ER and BA have a larger variance, especially with a few nodes of BA (so-called "hubs") having much larger node degrees as compared to the rest of the nodes. These differences are important to understand later the effects of gradual SDN penetration within such topological formations. For example, taking over the hubs of a BA-like multi-AS network could benefit convergence much more than targeting poorly connected, low-degree AS nodes. Such selective ISP targeting strategies for maximal impact with mimimal deployment will be investigated as future work (cf. Section 7.4.1).

4.4.2 Results: Convergence Times and Churn Rates

Our main objective is to study how gradual SDN penetration on the inter-AS level, as described in Chapter 3, affects *(i)* routing convergence times, and *(ii)* routing update churn rates. The first metric allows us to estimate for how long the inter-domain routing control plane is unstable[10]. The second metric allows us to estimate the average load imposed on the control plane of the border routers of the ASes. First we describe how we got results related to these two metrics of interest; afterwards we will present and comment on these results.

Convergence Time. The time measurement starts at the time of the AS-level incident, e.g., a link-down event where the link between two ASes (such as between a client AS and its upstream ISP) is brought down. As explained beforehand, we make use of fast keep-alive and hold-down timers in order to focus on what happens *after* the detection of the link-down event, i.e., the convergence process. The time measurement ends when the last routing update is received, while we use proper safe-guard intervals to make sure that we have seen *all* updates that arrive at the BGP route monitor device. The total time spent processing routing updates within the BGP and SDN clusters, as well as between them (i.e., due to the *interaction* between BGP and the SDN controller), is the measured convergence time that we are interested in.

Churn Rate. For this metric, we first measure the total number of routing updates within the convergence interval; this temporal interval has been already described above. Afterwards, we divide this number by the duration of the convergence interval. This yields the average routing update churn rate during the convergence process.

The results from our experiments are depicted *(i)* in Figure 4.6, regarding routing convergence times, and *(ii)* in Figure 4.7, regarding average routing update churn rates. The figures and cumulative results can be found at the end of this chapter. The results are based on a BGP MRAI of 30 sec; our findings using MRAIs of 0 sec were very similar and are omitted from the presentation since they do not add any further insights. The cause for this similarity is that Quagga route withdrawals are not rate limited by the MRAI (in contrast to announcements), but

[10] Of course instability does not necessarily mean packet loss; although transient phenomena such as black holes or routing loops might occur that affect loss and packet delay and jitter.

are rather propagated directly with 0 delay across the domains. At the same time, they are the main triggers for the path exploration process. This process primarily affects the convergence results that are seen in the figures. Therefore, a first insight we gained was the importance of the withdrawals for path exploration and the indifference that the MRAI value has on the results. Further observations and insights follow in the next subsection. We note that all presented box plots correspond to 20 runs per emulation parameter set[11], while all y-axes have a 0-baseline. The x-axis represents the percentage of AS nodes that belong to the SDN cluster, chosen uniformly at random from the AS set at hand. Note that the variance in the box plots is both due to the non-deterministic behavior of BGP and its interplay with SDN, and the randomization of the experimental setup per run (e.g., IP prefix assignment, random topologies, race conditions in BGP session establishment, etc.).

4.4.3 Observations and Insights

The most important insights that we gained via our analysis and emulation experiments are the following.

How does the scale of the graphs affect convergence, taking into account varying levels of SDN penetration? Convergence times exhibited a small-gradient linear decrease at the 8-node scale, with comparable times in E-R (Erdos-Renyi), B-A (Barabasi-Albert) and N-W-S (Newman-Watts-Strogatz) graphs for different SDN penetration levels, with the most notable gains in the clique case. At the 16-node scale, we observed a high-gradient linear decrease of convergence time with increasing size of the SDN cluster. We could not explain this relationship mathematically due to the difficulty in producing closed formulas and models for studying the interplay between BGP and SDN; we rather measured this interplay using production software. Finally, at the 32-node scale, a negative sub-linear relationship between convergence time and SDN penetration is observable. In this case, at 25% and 50% SDN penetration, the reduction in convergence time is less pronounced. However, the convergence time drops rapidly between the 50% and 75% levels, where the time is essentially cut by more than half. The cause of

[11] In the case of the 32-node clique, the results pertaining to the 50% SDN penetration level were not yielded due to a scalability problem with the emulation platform. Such problems will be solved in future versions of the SIREN framework, combined with improvements in the Mininet software base.

the latter observation is that the SDN controller takes over the highest-radix nodes and a good percentage of all available ASes (cf. Figure 4.4), thus severely speeding up convergence due to centralization. We also note the very small width of the box plot at the 75% SDN penetration cases across all scales and topology types; this indicates small variance on convergence times due to a stabilizing effect of centralization.

Moreover, the absolute convergence times are effectively doubled as the topology doubles in size, a rather expected observation. The same rule seems to apply for churn rates; bigger scales translate to higher churn, as intuition also suggests in view of the increasing complexity of the inter-network. At the 8-node scale, the churn rate ranges between only 1 and 3 updates/sec without major changes with increasing SDN penetration. At the 16-node scale, the churn rate increases slightly to up to 12 updates/sec and SDN penetration shows more clear gains, although it might even slightly increase churn in some cases. Finally, for the 32-node size, the churn rate increases much faster to up to 60 updates/sec due to the increased complexity and size of the network topology; it is thus evident that the churn rate and the topology size are not linearly related. SDN penetration shows though consistent reductions in churn; at this scale the churn exhibits a sub-linear decrease with increasing SDN penetration, similarly to the convergence time patterns. This is because while the control plane state is propagated at a higher pace via the controller, the CRWI-based rate-limiting on the controller's side smoothens the convergence process.

How does the network graph type affect convergence, considering varying levels of SDN penetration? The behaviors of convergence time and churn rates were not significantly affected by the graph type, and the patterns we observed regarding SDN penetration were more or less preserved across diverse topologies. This is an indicator that the presence of SDN does not cause unpredictable behaviors based heavily on the type of topology in question; what matters more is the scale of the topology, as already explained beforehand. We note that the clique has only slightly different behavior regarding absolute numbers; in fact convergence times and churn rates were always elevated in contrast to the other topologies. This is expected since the clique is essentially the "worst-case" scenario for BGP convergence; we verified this fact experimentally. Setups that are sparser than the clique also seem to benefit from increasing SDN penetration in similar ways. These lead

though to faster and less "chatty" convergence, due to the less intense path exploration process within less dense graphs.

Are there any severe effects of convergence on data plane traffic? In our experiments we focused primarily on the behavior of the control plane during convergence, since the effect of convergence on the data plane, e.g., on packet loss, has already been investigated in detail by Kushman et al. [159]. Hybrid BGP-SDN convergence and its correlation with the data plane traffic is the subject of future work. However, the preliminary examination of the interaction between the control and data plane on some of our small-scale experiments yielded the following insights. *(i)* Delayed convergence seems to primarily affect the latency, jitter and ordering of the data packets; in the fail-over case packets usually travel upon the different explored routes (even in circles) until the final valid paths become available. *(ii)* In our experiments, we observed negligible packet loss during convergence. That means that all intermediate nodes always have fail-over paths towards a destination, and all the packets eventually reach it, albeit following different routes until stability is restored. Lastly, for statistics on the frequency of BGP instability events and the percentage of affected BGP prefixes, we refer the reader to the work of Huston [127].

What are the key take-away messages regarding the interplay between legacy BGP AS groups and SDN AS clusters? Gradual deployment of SDN and inter-domain routing centralization actually helps. Benefits in convergence times can already be seen with small penetration levels, while benefits in churn rates need larger deployments to be tangible. In our experiments, the critical mass that a Routing-as-a-Service contractor should acquire to improve the stability of a multi-domain network seems to be somewhere between 25% and 50%. This percentage might be reduced with selective ISP targeting as described in Section 7.4.1. We note that a percentage of 100% in this case corresponds to the whole hybrid multi-domain network under study; this can be a certain geographical area containing ISPs and their clients (e.g., Switzerland), or a larger part of the Internet, but not necessarily the entire Internet. The resulting benefits are present within the corresponding area of influence, as we analyze in the next paragraph. Between the 25% and 50% levels and at the 32-node scale, convergence times can be reduced by ∼20%, while churn rates by ∼15% *on average*. During this transition, if today's routers and controllers can

handle slightly larger or comparable churn to the pure BGP case, SDN-based centralization can *benefit convergence while interacting with BGP*. Finally, we observed that the use of logical centralization of routing control accelerates convergence because of two main factors. *(i)* The state propagation process is accelerated due to the central point where parts of the state are gathered and are then directly communicated outside. This acceleration benefits both client and non-client ASes, but may increase the associated churn in some cases. *(ii)* The controller has a global overview of its cluster and the inter-domain network overall; this view is efficiently used for informed decisions related to path exploration, based on the cumulative routing feedback.

So, should a single RaaS controller take over inter-domain routing on behalf of 10s of 1000s of ASes to improve convergence properties? First, such a proposal is obviously infeasible due to the exacerbated characteristics of the technical and political challenges that we explained in Chapter 3. Second, technically it is not a necessary requirement for a single controller to manage the routing for a large percentage of the entire multi-AS network in order to optimize convergence. In fact, multiple clusters which are controlled by different controllers may be formed and can expand independently, starting from small deployments, with BGP or even new protocols used for their mutual coordination[12]. The gradual centralization can then help improve convergence close to the borders of their respective sphere of influence. Benefits at Internet scales could then be yielded in a staged manner of targeted deployment (e.g., on well-connected ISPs), with many controllers stabilizing their respective parts of the inter-domain setting. This process could then have an actual impact on stabilizing global inter-domain routing, as a joint service of the RaaS contractors.

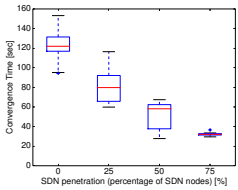
[12]Support for multi-controller and multi-cluster experiments in the context of the SIREN framework is future work (cf. Section 7.4.1).

4.5 Summary of Emulated Hybrid RaaS Gains

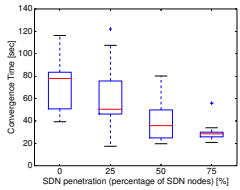
In Chapter 3 we proposed the gradual centralization of parts of the routing control logic of multi-domain AS-level networks. The goal was to improve general properties of inter-domain routing, such as the convergence behavior accompanying routing changes. The proposal can be technically applied using SDN mechanisms, while Routing-as-a-Service outsourcing frameworks may offer a financial basis for market adoption. In this chapter, as a use case, we evaluated the interplay between SDN-based routing centralization and classic BGP routing. To support that, we developed a hybrid BGP-SDN emulation framework and a multi-AS SDN controller running on top of it. Our fail-over experiments on hybrid graphs of diverse scales and types indicate that inter-domain routing centralization can improve convergence times even at small SDN penetration levels. Churn rates are comparable or slightly worse than pure BGP at small scales, with benefits shown at larger scales. Our work is another step towards extending the value proposition of SDN on the inter-domain level [29, 117], based on the radical idea of logically centralizing the inter-AS routing control plane (cf. Chapter 3). We note that this is one of the most challenging arenas for SDN to penetrate, due to the difficulty of making changes on how core routing works; politics and established practices can put a brake on novel technical approaches. Nevertheless, our current findings encourage further research along this direction of inter-domain SDN, specifically in the context of routing convergence.

Finally, we note that it is quite challenging to understand the dynamics between centrally controlled SDN clusters and the legacy BGP world, since this is a case of hybrid link-state and path-vector routing *across domains*, in a regime governed by different—sometimes conflicting—policies. Emulation may help researchers gather meaningful results in this area. With SIREN, we can experiment with real code, focusing on the experiment rather than the tool. For example, the research community may use SIREN to evaluate improvements to BGP induced by expanding SDN deployments, even if the context of their proposals is different from our initially assumed approach of logical RaaS-based centralization. Our future work agenda in the context of

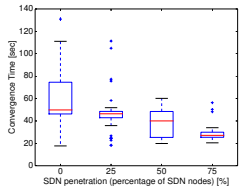
the SIREN emulation framework, controller and convergence-oriented experimentation is given in Section 7.4.1.



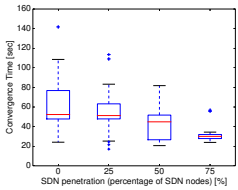
(a) 8-node clique (full mesh) graph



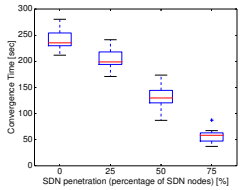
(b) 8-node Erdos-Renyi graph, p=0.5



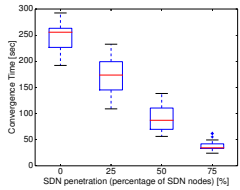
(c) 8-node Barabasi-Albert graph, m=2



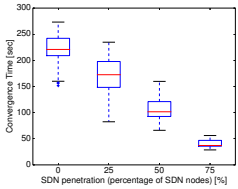
(d) 8-node N-W-S graph, k=2, p=0.5



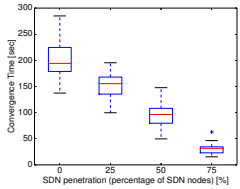
(e) 16-node clique (full mesh) graph



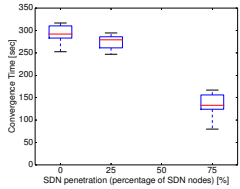
(f) 16-node Erdos-Renyi graph, p=0.5



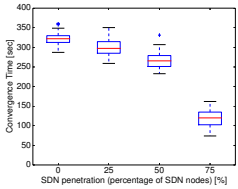
(g) 16-node N-W-S graph, k=4, p=0.5



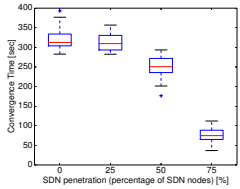
(h) 16-node Barabasi-Albert graph, m=4



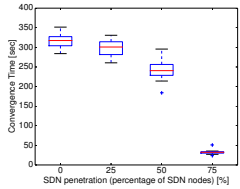
(i) 32-node clique (full mesh) graph



(j) 32-node N-W-S graph, k=8, p=0.5

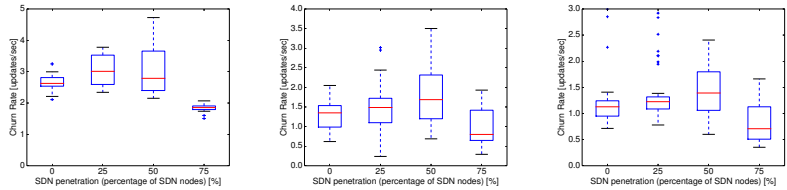


(k) 32-node Erdos-Renyi graph, p=0.5

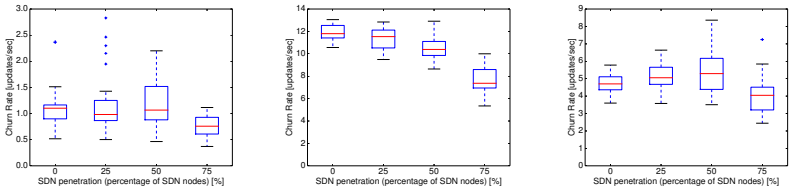


(l) 32-node Barabasi-Albert graph, m=8

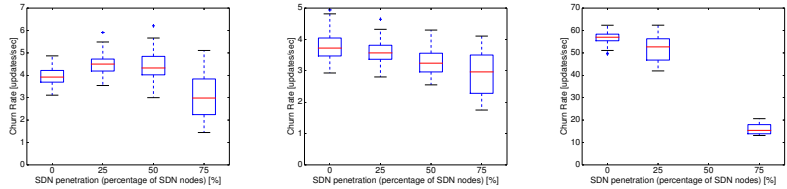
Figure 4.6: Convergence times [sec] (y-label) vs SDN penetration [%] (x-label) for the fail-over experiments: emulation results on graphs of different type and size, for BGP MRAI=30sec. Box plots correspond to 20 emulation runs. The reader is referred to the NetworkX website [191] regarding the explanation of graph parameterization (i.e., k, p, m).



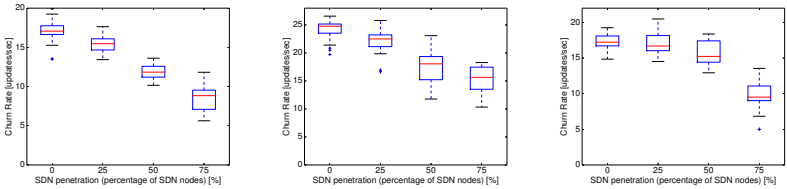
(a) 8-node clique (full mesh) graph (b) 8-node Erdos-Renyi graph, p=0.5 (c) 8-node Barabasi-Albert graph, m=2



(d) 8-node N-W-S graph, k=2, p=0.5 (e) 16-node clique (full mesh) graph (f) 16-node Erdos-Renyi graph, p=0.5



(g) 16-node Barabasi-Albert graph, m=4 (h) 16-node N-W-S graph, k=4, p=0.5 (i) 32-node clique (full mesh) graph



(j) 32-node N-W-S graph, k=8, p=0.5 (k) 32-node Erdos-Renyi graph, p=0.5 (l) 32-node Barabasi-Albert graph, m=8

Figure 4.7: Average routing update churn rates [updates/sec] (y-label) vs SDN penetration [%] (x-label) for the fail-over experiments: emulation results on graphs of different type and size, for BGP MRAI=30sec. Box plots correspond to 20 emulation runs. The reader is referred to the NetworkX website [191] regarding the explanation of graph parameterization (i.e., k, p, m).

Chapter 5

Centralized Path Computations and Min-Cuts on Policy-Governed Networks

> The same word we love and hate, leaves in different directions, taking different paths.
>
> —Dejan Stojanovic, *The Sun Watches the Sun*

A centralized RaaS platform, as the one described in Chapter 3, may implement multifarious routing policies on the inter-domain level. Metrics such as path diversity or policy-compliant path lengths and bandwidth are affected by such policies. *What is the effect of routing policy on path diversity on the AS level? How can we generalize the investigation of such effects for arbitrary networks and policies?* In this chapter, we introduce a general algorithmic framework for estimating the policy-compliant path diversity and bisection bandwidth of a network. In particular, we describe important use cases for policy-compliant min-cuts, including the use case of AS-level graph calculations used in the

context of Chapter 6, where the potential of overlay Internet graphs in terms of path diversity is evaluated. Afterwards, we give the idea and the algorithm for a custom graph transform to be used for calculating policy-compliant min-cuts on any network graph. The content is based on the parts that we contributed in our joint INFOCOM paper [144], together with Rowan Klöti, Dr. Bernhard Ager and Dr. Xenofontas Dimitropoulos.

5.1 Introduction

In this section, we first introduce the need for calculating metrics related to network resilience and availability on network graphs governed by policies in Section 5.1.1. We continue with a sketch of the use case of a RaaS contractor, operating on, e.g., a valley-free multi-domain network graph in Section 5.1.2. We end with the contributions of this chapter in Section 5.1.3, before giving its outline in Section 5.1.4.

5.1.1 Network Resilience and Availability Under Policies

Resilience is a desirable property for many networked systems and is often achieved through redundancy: when multiple paths exist between nodes, it is possible to route around a failed link. This resilience may be quantified in a graph-theoretic sense as the (edge-wise) *path diversity*. We thus adopt the following definition from related textbook literature.

Definition 1. *The* edge-wise path diversity *between two vertices in a graph is the number of edge-disjoint paths connecting them.*

The application of Menger's theorem [17] for edges allows the path diversity between two vertices to be calculated as the minimum cut, using an algorithm such as Ford-Fulkerson, with each edge having a unitary capacity. By adding non-unitary edge capacities, we can also calculate the *bisection bandwidth*. This metric is useful, e.g., for data center routing to optimize performance and robustness, or for quantifying the advantages of multipath intra-domain (e.g., Equal Cost Multi Path-ECMP) or inter-domain routing protocols like MIRO [257] in terms of achievable throughput. We adopt the following definition of the bisection bandwidth from related textbook literature.

Definition 2. *The* bisection bandwidth *between two vertices in a network is the maximum achievable flow between them.*

In practice though, networks do not permit all possible paths due to management policies. These can be the outcome of routing optimization techniques, security considerations or financial agreements [47]. For example, inter-domain paths in the Internet resemble a *valley-free* policy model [98], a simplified model of the business relationships between Autonomous Systems (AS). On the other hand, such policies substantially restrict which paths are permissible and constrain the effective path diversity. Thus, a rich—in terms of path choice—graph may not be fully utilized due to a restrictive policy (or set of policies) imposed over its paths. Therefore, calculating the *policy-compliant* path diversity and bisection bandwidth is desirable to answer questions such as: *(i)* how many *valid* edge-disjoint paths can exist between two nodes, or *(ii)* how much bandwidth[1] can be utilized between these two nodes before the network is overloaded, subject to network-wide policies. The goal is to understand the effect of network policy on network resiliency, availability and achievable throughput.

5.1.2 RaaS Contractor: Working on a Policy-governed Multi-AS Graph

The platform of an inter-domain RaaS contractor, based on the architectural approach described in Chapter 3—and evaluated via emulation in Chapter 4 and simulation in Chapter 6—provides an ideal vantage point on (parts of) the inter-domain graph. The important point here is that this platform (or NOS or controller) which provides outsourced routing services has a (semi-)*global* view of the multi-AS graph. Therefore, we can use it for a number of logically centralized calculations pertaining, for example, to shortest path routing or the discovery of the available path diversity for multipath inter-domain routing and other network services.

We note though that the graph that the contractor sees is actually governed by the policies of the individual participants that it serves, e.g., based on valley-free policies and relationships between peers,

[1] Policy-conformant bandwidth is always less or equal than the technically available bandwidth, due to the restraining effect of network policies.

customers and providers. The effect of routing policy on the properties of such (inter-)network graphs is not well understood; this knowledge is important for the correct operation of the RaaS contractor which combines the policy inputs from its clients. In this chapter, we thus develop an algorithm for transforming the policy-based graph in order to calculate policy-compliant min-cuts and estimate the available path diversity and bisection bandwidth within a policy regime. Classic off-the-shelf algorithms, such as Ford-Fulkerson for min-cuts or Dijkstra for shortest path routing, can run unmodified using the transformed graph and yield policy-compliant paths. Such algorithms can run on the RaaS platform on the (transformed) global view of the graph. The reader will see an example of policy-compliant path diversity calculations running on top of a simulated CXP platform in Section 6.3.5. These calculations will be used to estimate the potential of an Internet overlay graph, based on IXPs, in terms of path diversity. Here we explain how the general methodology works in detail and on which logical arguments it was based. The concrete contributions follow.

5.1.3 The Contributions of this Chapter

In this chapter, we introduce a general method for estimating the path diversity and bisection bandwidth of a network subject to policy constraints on the paths. We model the network topology as a directed graph with policy labels on the edges. We model network policies as regular expressions over these labels and require that all valid paths in the graph adhere to these regular expressions. Every regular language can be described by an automaton, specifically a Non-deterministic Finite state Automaton (NFA) [42]. This is a very important observation since policy-compliant NFAs, as visual graph objects, can be combined with a natural network graph to yield augmented graphs that encapsulate policies. In practice, using NFAs and the original graph, we develop a transformed graph that constrains paths to those accepted by the regular language. While path diversity calculations on the original graph under policies are hard, they become simple using general graph algorithms [89] on the transformed graph. For instance, classic graph algorithms can work on the transformed graph to find the policy-compliant max-flow or min-cut between two nodes as well as the paths achieving this flow. The price that is paid as a trade-off is that the graph needs to be augmented or extended to encapsulate the policies that govern it, with

the algorithms being agnostic to the change. Moreover, optimizing between constraining policies and the required resilience may be close to impossible in a traditional distributed setup. The overall view of a logically centralized RaaS contractor and the associated calculations can help with tuning such trade-offs.

In particular, we will show how the transformed graph can be used in order to obtain both upper and lower bounds on the path diversity or bisection capacity of the original graph. If the NFA fulfills certain criteria, the bounds are equal and therefore exactly the same as the actual value, i.e., the path diversity or bisection bandwidth are invariant under the transformation. Otherwise, we obtain upper and lower approximations that encapsulate the actual min-cut within their boundaries. The tightness of the boundaries depends on the complexity of the state transitions of the NFA, as we will explain later. We also show how constraints on traversed nodes may be imposed, including scenarios where both the nodes and edges are subject to separate constraints.

5.1.4 Outline of this Chapter

The rest of this chapter is structured as follows. Section 5.2 presents interesting use cases where path diversity and bisection bandwidth metrics—i.e., min-cuts—under policy compliance are required. Section 5.3 describes the basic ideas and the graph transform process; Section 5.4 substantiates this process using formal mathematical formulation and proving the needed claims. Moreover, Section 5.5 presents some proof-of-concept insights yielded from the application of our algorithm on Internet AS-level topologies, demonstrating our approach[2]. In Section 5.6 we report on related work in the field of network resilience and policy-compliant min-cuts. Finally, we conclude the chapter.

5.2 Use Cases for Policy-Compliant Min-Cuts

Calculating the policy-compliant path diversity and bisection bandwidth can be applied on a wide range of scenarios to quantify the resilience and achievable throughput of a network, including intra-domain and

[2]The source code can be found as part of the CXP repository [152].

inter-domain use cases. With our approach, the only requirement is that the network policy should be expressible with a regular expression; the form of the corresponding NFA dictates whether we can calculate the exact value or an approximation as we will describe in Section 5.4. Many network policies used in practice are expressible through regular expressions [227]. We identify the following use cases for policy-compliant min-cuts, with the ones pertaining to inter-domain networks being of greater interest to the RaaS approach described in Chapter 3.

5.2.1 Basic Use Case: Inter-Domain Routing

Classic Valley-free Policies

A basic use case is the calculation of the path diversity of a multi-domain network, i.e., an Internet-like setup, conforming to a simplified valley-free policy model [98]. Path diversity in this case may refer to the number of edge-disjoint paths between two ASes, where each edge connects two neighboring ASes together. Edges are labelled as *peer-to-peer* (`p2p`), *provider-to-customer* (`p2c`) or *customer-to-provider* (`c2p`) relationships. This topology and corresponding edge labels can be obtained from datasets like CAIDA [48,50]. We note that, in reality, such links correspond to multiple network layer links and even more physical (e.g., optical fiber) links; the calculated path diversity is therefore a *lower bound* of the physical path diversity[3]. We note that this tendency for underestimating the AS-level path diversity, due to the relationship-based "link" abstraction, has an unknown contradictory effect versus another limitation of the dataset; that is, certain seemingly edge-disjoint paths may in fact share optical fiber cables on layer 1 (e.g., transatlantic links). The mapping of the true AS-level Internet topology and the uncovering of these intricacies (for example, determining exactly the effect of this interplay) is out of the scope of this work. Nevertheless, despite its limitations, the use of the AS relationship dataset is valuable for demonstrating a direct feasible application of our algorithm that

[3]For example, the presence of a p2p relationship between two ASes implies that these ASes are connected at layer-3; this connection might in reality be multiple PoP-level connections that are hosted in public IXPs or private peering points. Also, it may correspond to a single or multiple redundant fiber cables; in any case the path diversity that is yielded based on the high-level relationship or layer-3 information is a lower bound of the underlying low-level diversity.

can be easily reproduced by other researchers (see application insights of Section 5.5).

In the valley-free model, the global inter-domain policy can be expressed with the following regular expression: `c2p`*`p2p?p2c`*. Paths can only go first uphill (`c2p`) and then downhill (`p2c`), while at most one `p2p` link can connect an uphill with a downhill transition forming a "mountain" with a `p2p` link on its "peak". The peak may also be sharper, with a direct transition from uphill to downhill. We will revisit valley-free path diversity in Section 5.5, where we provide some high-level insights on results yielded on the AS-level path diversity using the CAIDA AS relationship dataset.

Beyond Classic Policies - Negative Waypoint Routing

Valley-free is a basic family of policies that approximates the current market relationships in the Internet. Ideally, we would like to examine additional routing policies on top of the classic ones, across domains. Examples are *(i) waypoint routing*, i.e., forcing the traffic to pass over certain waypoints before reaching its destination, and *(ii) negative routing*, i.e., forcing the traffic to avoid certain nodes or links in the network. Such policies further perplex path diversity calculations but are interesting for specific real-world scenarios and use cases.

Consider the following (slightly contrived) example which encapsulates waypoint and negative routing policies. We assume a valley-free Internet, in which each inter-AS edge is directed and is annotated with a tuple label: *(relationship_type, next_AS)*, where *relationship_type* is `p2p`, `p2c` or `c2p` and *next_AS* is the edge-terminating AS. A government organization in AS `A` wants to send traffic to one of its embassies in AS `B`, located in another country. The traffic from `A` to `B` needs to pass over a special encryption middlebox; there are two clones of this middlebox in AS $\mathtt{W}_1$ and AS $\mathtt{W}_2$ for redundancy. The original traffic can pass through any AS before it reaches waypoint $\mathtt{W}_1$ or $\mathtt{W}_2$, except for AS `X`, which is governed by a rival administration. After either of the waypoints is traversed, traffic can go through any AS in the world—including `X`—until it reaches its destination `B`, where it is decrypted on the premises. This policy corresponds to a regular expression (omitted here for brevity) which can be mapped in turn to an NFA that accepts the expression, as depicted in Figure 5.1. The resulting policy-compliant NFA is then a composition of the valley-free and the negative waypoint routing NFAs.

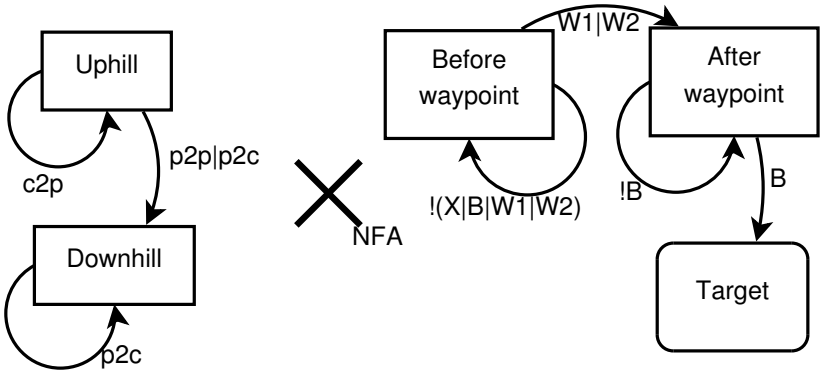


Figure 5.1: NFAs for the negative waypoint routing use case: valley-free (left side) and waypoint routing policies (right side). The "Uphill", "Downhill" and "Target" NFA states are terminating.

In this particular case, the path diversity metric can help the traffic sender determine how many inter-domain links could be brought down until the organization-to-embassy communication is crippled, e.g., in the event of a cyber-war launched by the rival country. We view this as a "negative waypoint" inter-domain routing use case, since we aim at approximating the multitude of edge-disjoint paths that *avoid* a certain node or edge in the inter-AS graph and pass through certain waypoints. The problem cannot be solved by simply removing AS `X`, pruning its corresponding edges and calculating the diversity on the pruned graph using the sender-waypoint and waypoint-receiver node pairs. As the policy dictates, AS `X` *can be traversed* after the traffic has been processed by *either* AS $\mathtt{W}_1$ or AS $\mathtt{W}_2$, but not sooner. The NFA encompasses both this stateful routing decision process and the underlying valley-free conditions, and allows us to encompass a complex policy in a simple graph object. Having such NFAs at hand, we will explain later how the policy-compliant diversity can be calculated.

5.2.2 Other Use Cases: MPTCP and DDoS Attacks

MultiPath TCP (MPTCP)

MPTCP [121] is a proposed extension to TCP from the IETF [88], allowing TCP connections to use multiple paths to increase resource utilization, redundancy and availability. This is especially useful when multiple wireless channels with different properties are available, or for better utilizing dense data center topologies, to exploit the large bisection bandwidth offered. Consider the data center scenario, where the operator has full control over the endpoints and switches and can route subflows individually (intra-domain flow routing use case).

Here, the number of disjoint available paths that MPTCP can send traffic over is useful information for the MPTCP implementation. Path diversity calculations can yield an approximation of the maximal number of distinct flows that MPTCP can push in the network, without these flows contending with each other simultaneously for bandwidth. The approach for calculating the effective path diversity can take into account domain-specific policies which have been, e.g., expressed via Merlin [227]. In addition, it enables data center operators to compute the policy-compliant available bandwidth between two areas of a network. This can help to estimate the transmission time that a MPTCP bulk data transfer between two server clusters may require, or whether the network is utilized properly during the transfer.

DDoS Link-Flooding Attacks

Estimating the bisection bandwidth of a data center can lead to an approximation of the attack budget that a DDoS link-flooding attack against data center core links, such as Crossfire [136], may require. The recent attack against Spamhaus [205], reaching a cumulative bandwidth of 300 Gbps, indicates that such information is valuable both for the attacker and the defender.

On the one hand, the attacker tries to find weak links which he can deplete at minimal cost, isolating entire domains from the Internet [136] by crippling their upstream connectivity. On the other hand, the defender tries to increase the cost for the attacker via suitable network and traffic engineering [62]. Knowing how much bandwidth needs to

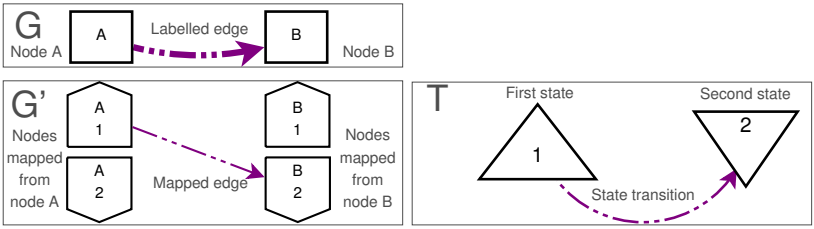
be depleted to cut off a network from the rest of the Internet can help an operator perform an informed risk assessment of a possible attack, while taking into account the routing policies imposed over the network.

This process can also be extended towards inter-domain defense mechanisms to fend off large-scale link-flooding attacks that deplete inter-AS core links, such as Coremelt [231]. For example, the networks under attack (being transit ISPs or enterprise networks) may collaborate with their upstream providers in order to perform an informed multi-AS risk assessment and countermeasure placement; for example, choosing together where to place more capacity based on their inter-domain routing policies. The algorithm that we propose is actually oblivious to the graph type that it operates on; the operators may choose to use low-level network graphs (switches, routers and layer-2/3 links) or work on high-level abstractions (ASes as nodes, inter-AS "links"). The proper abstraction is determined based on the nature of the use case (privacy preferences, issues with complexity, etc.).

5.3 Explanation of the Custom Graph Transform

First, we assume that we have a centralized entity at our disposal; this entity sees a network graph (e.g., nodes and links) and knows the policies that govern it. Using this information, it can perform path computations—such as min-cut calculations—related to the network, under policies. The abstraction of this operation (e.g., the graph abstraction) depends on the corresponding use case.

Our objective is to transform a directed labelled graph G into another graph G', such that: *(i)* only policy-compliant paths exist in G', and *(ii)* the transformation does not distort the minimum cut. The minimum cut may represent either the bisection bandwidth or, by choosing unitary edge capacities, the path diversity (cf. Menger's theorem [17]). We define a *policy-compliant path* as any path whose *path string*, i.e., the string resulting from concatenating the labels of the edges within the path, belongs to some regular language L. Here is how we accomplish this. Every regular language L is represented by a finite state automaton M, and vice versa. Therefore, upon traversing an edge in the graph, we



(a) Original→Transformed Graph (b) NFA

Figure 5.2: Graphical example of the tensor product. For each label $s \in \Sigma$, the tensor product of the edges in G labelled with s and the edges in T labelled with s (representing the state transitions in M when s is entered as an input) yields the set of mapped edges in G'. We represent the edge label s symbolically via the colour and style of the edge. The "pentagon" nodes in G' are derived from the "triangle" nodes of the NFA and the "square" nodes of the original graph G. All NFA states are assumed to be terminating.

need to accordingly change the state in the automaton, as if the edge label had been given to the automaton as an input.

The first idea is to use the *tensor product* of the graph G and the policy-checking automaton M (Figure 5.2 gives an example). From M we define the transition graph T, which has a node set consisting of the states of M and an edge set representing the permitted state transitions of M. The edges in G and T are labelled with policy-related symbols that result in the state transitions represented in T.

For each of these symbols $s \in \Sigma$, we form the subgraphs G_s and T_s consisting of all the edges labelled with s. We then form the *tensor product* of these subgraphs, which is defined as follows: if v_1 and v_2 are nodes in G and q_1 and q_2 are nodes in T, then G' contains the nodes (v_1, q_1), (v_1, q_2), (v_2, q_1) and (v_2, q_2). Likewise, if there is an edge (v_1, v_2) in the subgraph G_s and an edge (q_1, q_2) in the subgraph T_s, then the tensor product contains the edge $((v_1, q_1), (v_2, q_2))$. The union of all of these tensor products is the transformed graph G'. This allows us to move both between the nodes v_1 and v_2 in G and the nodes q_1 and q_2 in T (and therefore the states q_1 and q_2 in M) at the "same time". We note that we define M to be a *Non-deterministic* Finite

state Automaton (NFA) because an NFA is typically smaller than a corresponding *Deterministic* Finite state Automaton (DFA)[4].

This transform—based on the tensor product—gives us part of the solution, but it does not guarantee that the minimum cut calculated over G' represents accurately the corresponding minimum cut of G. A single edge in the original graph G may be mapped to a set of parallel edges in the transformed graph G'. To demonstrate this, consider the possibility of a transition from one state $\{q_1\}$ to two states $\{q_2, q_3\}$ over an edge (v_1, v_2). This will be mapped to two edges in G': $((v_1, q_1), (v_2, q_2))$ and $((v_1, q_1), (v_2, q_3))$, even though there is just one edge in G. Apparently, this constitutes an inflation of the minimum cut.

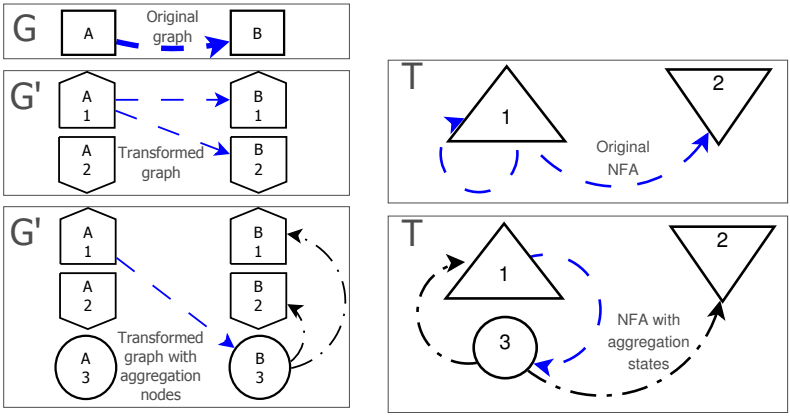
Therefore, the second idea is to *force* the transform to maintain the minimum cut. This may not always be possible, as we will explain later. If it is not, the result is still an approximation of the minimum cut, for which we will give lower and upper bounds in Section 5.4.4. The core idea is to add *aggregator* states to the NFA and consequently to G' so that the min-cut paths between two nodes must traverse at most the same number of parallel edges as in G, which limits the minimum cut in G' to the same value as in G.

To preserve the structure of the NFA—and thus the policy it describes—we utilize ε-*transitions*. ε-*transitions* can be thought of as "free" transitions: they do not consume a symbol during traversal. In our case, this means that we do not need to traverse an edge between nodes in G in order to traverse an ε-transition.

Thus where there is a chance of the min-cut being inflated, we add an aggregator node (for each edge label) and use ε-transitions to "channel" all of the paths through this node. Each node in G has corresponding aggregator nodes in G' *as needed* and where *it is applicable*; the goal is to avoid unnecessary inflations of the min-cut.

There are several possible cases for the to-be-aggregated transitions: *(i)* from a single state to another single state (*one-to-one*), *(ii)* from a single state to multiple states (*one-to-many*), *(iii)* from multiple states to a single state (*many-to-one*), or *(iv)* from multiple states to multiple states (*many-to-many*). In the first case, the min-cut is

[4]While a DFA may be constructed from any NFA by power-set construction, it has the same expressive power as the corresponding NFA, but potentially exponentially more states. Also, it is not compatible with the NFA concept of ε-*transitions*, which we will heavily use later.

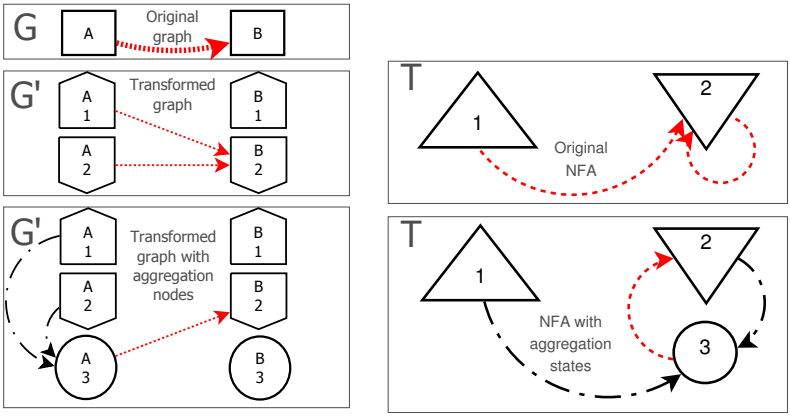


(a) Original→Transformed Graph (b) NFA

Figure 5.3: The use of aggregation nodes (represented with circles) for a *one-to-many* state mapping. The two blue dashed transitions of the NFA are aggregated using one aggregation node. ε-transitions are represented by black dotted dashed lines. All NFA states—except for the aggregator states—are assumed to be terminating.

always invariant; no inflation can occur. The latter cases are depicted in Figure 5.3, Figure 5.4, and Figure 5.5, respectively. In the *one-to-many* and *many-to-one* cases the addition of aggregation nodes leads to the correct min-cut value, while the *many-to-many* case requires careful consideration, for the following reasons.

In the latter case, we can only maintain the accurate min-cut using aggregation states if the set of state transitions can be expressed as the Cartesian product of two subsets of the set of state nodes. This results in complete bipartite subgraphs on the transformed graph G' with aggregatable transitions, as shown in Figure 5.5. Therefore, if this is not the case, we need to break the transition down to n disjoint state transition sets of the cases *(i)* to *(iv)* and perform the transform for each one of them; each set introduces one (*1-to-1, 1-to-N, N-to-1*) or two (*M-to-N*) extra aggregator states. We analyze the nature and complexity of this process in detail in Section 5.4.5.

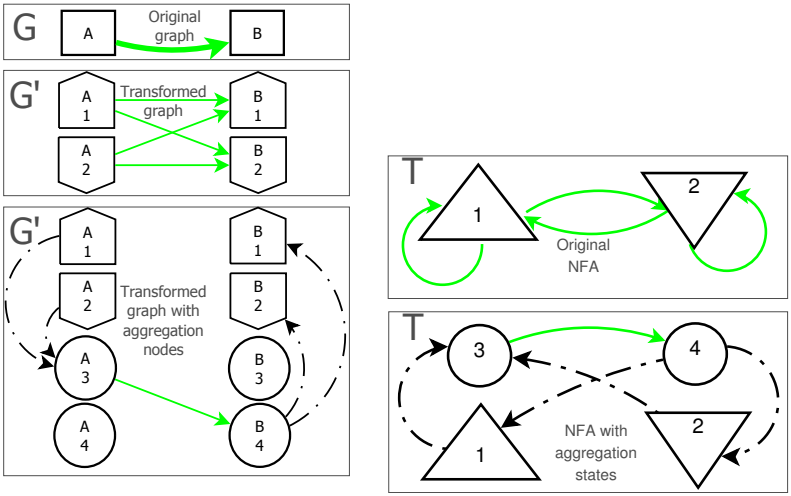


(a) Original→Transformed Graph (b) NFA

Figure 5.4: The use of aggregation nodes (represented with circles) for a *many-to-one* state mapping. The two red dotted transitions of the NFA are aggregated using one aggregation node. ε-transitions are represented by black dotted dashed lines. All NFA states—except for the aggregator states—are assumed to be terminating.

Furthermore, regarding edge capacities we provide different values to yield upper and lower bounds on the min-cut, as we will show later. To calculate the min-cut bounds with conventional algorithms, we need to have a single terminating node in G', which implies a single terminating state in M. If we have more than one terminating state, then prior to proceeding we need to create a virtual terminating state and copy all of the transitions to the previous terminating states to the new one.

Finally, if we want to consider constraints on nodes as well as edges, we can split the nodes under consideration in two halves as follows: all of the incoming edges connect to one half and all of the outgoing edges connect to the other. We then add a labelled edge between those halves, allowing the edge to represent the node and encompass its constraints.



(a) Original→Transformed Graph (b) NFA

Figure 5.5: The use of aggregation nodes (represented with circles) for a *many-to-many* state mapping. The four green transitions of the NFA are aggregated using two aggregation nodes. ε-transitions are represented by black dotted dashed lines. All NFA states—except for the aggregator states—are assumed to be terminating. Note that in this example the NFA graph T is mapped to a complete bipartite graph on the transformed graph G'.

5.4 The Math behind the Transformation Algorithm

Here we present the required definitions, algorithmic steps, proofs of correctness, and complexity of the proposed algorithm.

5.4.1 Notation and Definitions

For the rest of this chapter, we will use the following mathematical notation and definitions:

- $G = (V, E)$ is a labelled directed graph with nodes V, edges E, a labelling function $l : E \to \Sigma$ that maps edges in E to corresponding labels in Σ, and a capacity function $c : E \to \mathbb{R}^+$ that maps edges to edge capacities. For path diversity calculations, we choose $c(e) = 1$ for all $e \in E$, in contrast to bisection bandwidth calculations.
- $L \subseteq \Sigma^*$ is a regular language. We require that for any path $(e_1, ..., e_n)$ in G, the corresponding path string $l(e_1)...l(e_n)$ formed by the edge labels be in L.
- $M = (Q, \Sigma, \Delta, q_0, F)$ is a NFA with states Q, input symbols Σ, state transitions $\Delta \subseteq Q \times \Sigma \times Q$, terminating states $F \subseteq Q$ and a starting state $q_0 \in Q$. M accepts L. ε-transitions are allowed with $\varepsilon \in \Sigma$.
- $T = (Q, \Delta)$ is the directed graph derived from M.
- $G' = (V', E')$ is the final transformed graph: its formation is such that the policy constraints are met and the minimum cut is not inflated, if possible.

5.4.2 Single Virtual Termination States

In order to have a single terminating node in G' for use with Ford-Fulkerson (or any other flow algorithm), the set of terminating states F in M must be no larger than one. If $|F| > 1$, we add a new termination state q^*. A state transition to q^* is added from every other state which previously had a transition to a terminating state, and F is updated accordingly as follows:

$$Q := Q \cup \{q^*\} \tag{5.1}$$

$$\Delta := \Delta \cup \{(q', s, q^*) \mid (q', s, q) \in \Delta \wedge q \in F\} \tag{5.2}$$

$$F := \{q^*\} \tag{5.3}$$

5.4.3 Tensor Product Transform: Steps

The algorithm comprises of the following steps:

1. We form subgraphs $G_s \subseteq G$ consisting of all the edges labelled with $s \in \Sigma$:

$$G_s := (V, E_s) \tag{5.4}$$

$$E_s := \{(v_1, v_2) \mid (v_1, v_2) \in E \wedge l((v_1, v_2)) = s\} \tag{5.5}$$

2. We form subgraphs $T_s \subseteq T$ of all the state transitions for an input symbol $s \in \Sigma$:

$$T_s := (Q, \Delta_s) \tag{5.6}$$

$$\Delta_s := \{(q_1, q_2) \mid (q_1, s', q_2) \in \Delta \wedge s' = s\} \tag{5.7}$$

3. We augment Δ_s with aggregator states, if necessary.
   (i) We require that Δ_s has the form $Q'_s \times Q''_s$ with $Q'_s \subseteq Q$ and $Q''_s \subseteq Q$.
   (ii) If this is not the case, we decompose Δ_s into n_s disjoint sets $\Delta_{s,k}$ such that each subset has the form $Q'_{s,k} \times Q''_{s,k}$ and repeat the following for every k:

   (a) We add a new aggregator state q'_s if $|Q'_s| > 1$ and q''_s if $|Q''_s| > 1$:

$$Q := \begin{cases} Q & |Q'_s| = 1 \\ Q \cup \{q'_s\} & |Q'_s| > 1 \end{cases} \tag{5.8}$$

$$Q := \begin{cases} Q & |Q''_s| = 1 \\ Q \cup \{q''_s\} & |Q''_s| > 1 \end{cases} \tag{5.9}$$

   (b) If we added a q'_s in the previous step, we connect the preceding states to it with an ε-edge. Likewise for q''_s and succeeding states:

$$\Delta_\varepsilon := \begin{cases} \Delta_\varepsilon & |Q'_s| = 1 \\ \Delta_\varepsilon \cup (Q'_s \times \{q'_s\}) & |Q'_s| > 1 \end{cases} \tag{5.10}$$

$$\Delta_\varepsilon := \begin{cases} \Delta_\varepsilon & |Q''_s| = 1 \\ \Delta_\varepsilon \cup (\{q''_s\} \times Q''_s) & |Q''_s| > 1 \end{cases} \tag{5.11}$$

   (c) Finally, we connect states q'_s and q''_s with the aggregated state transition edge. If we did not use aggregating nodes on either side because there was only one state in Q'_s or Q''_s, we use that single state instead:

$$Q'_s := \begin{cases} Q'_s & |Q'_s| = 1 \\ \{q'_s\} & |Q'_s| > 1 \end{cases} \tag{5.12}$$

$$Q''_s := \begin{cases} Q''_s & |Q''_s| = 1 \\ \{q''_s\} & |Q''_s| > 1 \end{cases} \tag{5.13}$$

$$\Delta_s = Q'_s \times Q''_s \tag{5.14}$$

4. For each pair of G_s and T_s that we derived in the previous steps, we calculate the *tensor product* $G'_s = G_s \times T_s$. The nodes of G'_s are given by:

$$V' := V \times Q \tag{5.15}$$

The edges of G'_s are given by:

$$\begin{aligned} E'_s := \{&((v_1, q_1), (v_2, q_2)) \\ &\mid (v_1, v_2) \in E_s \wedge (q_1, q_2) \in \Delta_s\} \end{aligned} \tag{5.16}$$

Furthermore, ε-transitions are mapped to edges in G' that are effectively located within a single node in G:

$$E'_\varepsilon := \{((v, q_1), (v, q_2)) \mid v \in V \wedge (q_1, q_2) \in \Delta_\varepsilon\} \tag{5.17}$$

5. G' is the union of G'_s for each $s \in \Sigma$ (including ε):

$$G' := \bigcup_{s \in \Sigma} G'_s = (V', \bigcup_{s \in \Sigma} E'_s) \tag{5.18}$$

6. For the edge capacities $c' : E' \to \mathbb{R}^+$ in G', we have different values for the upper and lower bound of the minimum cut. (5.19) and (5.20) give the capacities c'_{upper} and c'_{lower} for the upper and lower bounds of the minimum cut, respectively.

$$c'_{upper}(((v_1, q_1), (v_2, q_2))) = \begin{cases} c(v_1, v_2) & v_1 \neq v_2 \\ \infty & v_1 = v_2 \end{cases} \tag{5.19}$$

$$c'_{lower}(((v_1, q_1), (v_2, q_2))) = \begin{cases} \frac{c(v_1, v_2)}{n_s} & v_1 \neq v_2 \\ \infty & v_1 = v_2 \end{cases} \tag{5.20}$$

Where $s = l((v_1, v_2))$ and n_s is the number of disjoint sets $\Delta_{s,k}$ that Δ_s is decomposed into in step 3.

5.4.4 Correctness and Min-cut Bounds

We next prove that the transformed graph contains only valid policy-compliant paths (cf. claims 1 and 2), that capacities c'_{lower} and c'_{upper} yield lower and upper bounds of the min-cut (cf. claims 3 and 4), respectively, and that we obtain an exact value for the policy-compliant min-cut as long as certain conditions hold, pertaining to the form of the NFA (cf. claim 5).

Claim 1. *Given a path P in G, if the string formed by the concatenation of the edge labels of P is not in L, then no corresponding path P' exists in G'.*

Proof. The edge labels of P form a string. This string contains at least one edge e with a label $s \in \Sigma$ which results in the string no longer being in L. This implies that there is no outgoing state transition from the preceding state to any other state in the NFA. The tensor product of the NFA with the edge e is thus empty. Therefore, there is no edge to the next node mapped from P and no P' can be formed in G'. □

Claim 2. *If a path P exists in G and the string formed by the concatenation of the edge labels of P is in L, then there exists a corresponding path P' in G'.*

Proof. If the edge labels traversed by P form a string in L, then that string represents a sequence of valid state transitions in NFA M (respectively the NFA graph T) from the starting to a terminating state. The tensor product for each $s \in \Sigma$ gives us a connection between two nodes if the edge corresponds to a valid state transition in M. As the string is in L, we know that all of the edge transitions are valid and therefore that all of the nodes mapped from P to P' are connected. Thus a valid path P' can be formed in G'. □

Claim 3. *Let v_1 and v_n be nodes in G. Let q_0 and q_t be the starting and terminating states in the NFA. Let the edge capacities of G' be c'_{lower} of (5.19). Then the minimum cut between (v_1, q_0) and (v_n, q_t) in G' is less than or equal to the minimum cut between v_1 and v_n in G, taking into consideration only those paths whose edge labels form strings in L.*

Proof. From Claim 1 and Claim 2 we have that any path in G' corresponds to a valid path in G, and vice versa. For each pair of adjacent nodes v_k and v_{k+1} in the path in G we have the capacity $c((v_k, v_{k+1}))$ for the edge that connects them. The edge (v_k, v_{k+1}) is mapped to n_s edges in G', each having a capacity of $\frac{c((v_k, v_{k+1}))}{n_s}$, where n_s is the number of disjoint sets $\Delta_{s,k}$ that Δ_s is decomposed into in step 3. All of these n_s mapped edges in G' therefore have a cumulative capacity of $c((v_k, v_{k+1}))$, the same as between the pair of nodes in G that they were mapped from. Hence the minimum cut between each pair of nodes in the path in G' is at most as large as that in G, while it may also be

smaller due to the Δ_s decomposition. By induction, this applies to the path as a whole, and by generalization to all paths in G'. Therefore, the minimum cut will not be overestimated and the calculated value in G' is a lower bound of the actual min-cut in G. □

Claim 4. *Let v_1 and v_n be nodes in G. Let q_0 and q_t be the starting and terminating states in the NFA. Let the edge capacities of G' be c'_{upper} of (5.20). Then the minimum cut between (v_1, q_0) and (v_n, q_t) in G' is greater than or equal to the minimum cut between v_1 and v_n in G, taking into consideration only those paths whose edge labels form strings in L.*

Proof. From Claim 1 and Claim 2 we have that any path in G' corresponds to a valid path in G, and vice versa. For each pair of adjacent nodes v_k and v_{k+1} in the path in G we have the capacity $c((v_k, v_{k+1}))$ for the edge that connects them. The edge (v_k, v_{k+1}) is mapped to n_s edges in G' which all have the capacity $c((v_k, v_{k+1}))$, where n_s is the number of disjoint sets $\Delta_{s,k}$ that Δ_s is decomposed into in step 3. Hence the capacity of an edge e in G and of any edge e' in G' that e is mapped to is the same. All valid paths in G' therefore have at least the same minimum cut as the ones in G from which they are mapped, while there may be several corresponding parallel paths in G' due to the Δ_s decomposition. Therefore, the minimum cut will not be underestimated and the calculated value in G' is an upper bound of the actual min-cut in G. □

Claim 5. *The lower bound of claim 3 and the upper bound of claim 4 coincide and the min-cut calculation is exact if the following condition holds:*

$$\forall s \in \Sigma : \exists Q'_s, Q''_s \subseteq Q : \Delta_s = Q'_s \times Q''_s \tag{5.21}$$

Proof. If (5.21) is true, then n_s is equal to one for every $s \in \Sigma$. Accordingly, $c'_{lower}(e) = c'_{upper}(e)$ for every $e \in E'$. This means that the lower and upper bounds of the minimum cut are equal. Since the actual min-cut lies between these values, it must therefore be equal to the lower and upper bounds and the calculation is exact. □

5.4.5 The Maximal Biclique Generation Problem

The number of disjoint sets $\Delta_{s,k}$ that Δ_s is decomposed into in step 3 of the graph transform (cf. Section 5.4.3) determines how far off the lower

and upper min-cut bounds are from the actual value in the worst case. Each $\Delta_{s,k} = Q'_{s,k} \times Q''_{s,k}$ should be expressed as the Cartesian product of two subsets of the state node set Q. The tensor product of each $\Delta_{s,k}$ with a link $(v_i, v_j) \in E_s$ is equivalent to a *biclique* in the transformed graph, i.e., a complete bipartite graph.

This process can be thus reduced to finding the minimal number of complete bipartite graphs—or bicliques—that cover the transformed subgraph corresponding to the initial labelled link (v_i, v_j). The found bicliques can then be used inversely to determine the k $\Delta_{s,k}$ sets, i.e., to determine the Cartesian product decompositions of the NFA state transitions (each associated with a symbol s). This observation helps to estimate the complexity of the decomposition problem, though a complete formal analysis is outside the scope of this work.

Finding maximal bicliques is generally NP-Complete [200], and we refer the reader to existing literature for proposed solutions [35, 137]. If looser bounds are acceptable, a heuristic solution could be used. We typically only need to decompose small subgraphs corresponding to simple NFAs and this only needs to be performed once per Δ_s—i.e., per symbol— when transforming the graph. The time costs are significant only in cases where the NFA contains very large and complicated transitions. For many common scenarios, including *1-to-1*, *1-to-N*, *N-to-1*, and *M-to-N*, the decomposition is trivial (see Figure 5.2 to Figure 5.5). For example, the transitions of valley-free and extended valley-free with multiple p2p links (cf. Chapter 6) fall in the latter category.

5.4.6 Algorithmic Complexity

Next, we consider the spatial and temporal complexity of the proposed algorithm, according to the steps presented in Section 5.4.3.

Spatial Complexity

Here we consider the space complexity, which depends on the number of nodes and edges in graph G and the number of states and transitions in NFA M. There are $|Q|$ states in the NFA. We may need to add $O(|\Delta|)$ states for aggregation (at most one per transition). Therefore, applying

the tensor product and taking into account the aggregation states gives us a total node complexity of:

$$|V'| = O(|V|(|Q| + |\Delta|)) \tag{5.22}$$

Typically, one edge in G will be mapped to one edge in G', plus some ε-edges. In the worst case, we may need $O(|\Delta|)$ edges to map between two nodes (if many need to be decomposed into disjoint subsets in step 3). We may also need $O(|\Delta|)$ ε-edges for each node, yielding a total edge complexity of:

$$|E'| = O(|\Delta|(|V| + |E|)) \tag{5.23}$$

Temporal Complexity

To obtain G_s, executing (5.5) requires $\Theta(|E|)$ steps (the nodes are maintained), while obtaining the T_s requires executing (5.7), requiring $\Theta(|\Delta|)$ steps. Step 3 will be executed in the worst case n_s times, where n_s is the number of disjoint transition sets that Δ_s may be broken into, which cannot be larger than $|\Delta|$. Additionally, it demands t_{dec} time, which is the amount of time required to actually decompose state transitions in M. The latter depends on Δ, and may be of non-deterministic polynomial complexity. However, t_{dec} is generally negligible in practice for many common scenarios (i.e., $|V| >> |Q|$ and $|E| >> |\Delta|$). 3a and 3c require only constant time, since they add a single object to a set. 3b requires $O(|Q|)$ steps, giving us $O(|Q||\Delta|)$ total time for step 3. For step 4, we have $O(|V||Q|)$ for executing (5.15), $O(|E||\Delta|)$ for executing (5.16) and $O(|V||\Delta|)$ for executing (5.17). Finally, (5.18) requires $O((|V| + |E|)|\Delta|)$ steps. Thus, the total time complexity is:

$$t = O(|V||Q| + |\Delta|(|V| + |E| + |Q|)) + t_{dec} \tag{5.24}$$

In practice, the total running time is dominated by the min-cut calculation on the transformed graph—e.g., via Ford-Fulkerson or Edmonds-Karp—rather than by the graph transformation process itself. The spatial complexity of the transformed graph in terms of the sizes $|V'|$ and $|E'|$ is the most important factor for the min-cut run-time. We demonstrate this observation in the following section.

Back-of-the-Envelope Calculation: Complexity of Transform on AS-level Internet

According to Willinger et al. [251] "today's AS-level Internet is made up of some 30,000-40,000 actively routed ASes and an order of magnitude more links". Therefore, let us perform a back-of-the-envelope calculation of the complexity of the graph transform applied on a simulated AS-level topology. We assume for simplicity the valley-free NFA of Section 5.2, with $|Q| = 2 = c_q$ (constant) and $|\Delta| = 3 = c_d$ (constant). In the following notation, we will use the form c_{var_X} to denote a constant related to a variable *var_X* (we will use an abbreviated form of the variable's name for brevity). We make the following observations regarding the associated algorithmic complexity.

The original spatial complexity is equal to:
$|V| = c_v \cdot 10,000 = c_v \cdot 10^4$ nodes, where c_v is a constant, and
$|E| = c_e \cdot 100,000 = c_e \cdot 10^5$ edges, where c_e is a constant.

The spatial complexity of the transformed graph is equal to:
$|V'| = O(|V|(|Q| + |\Delta|)) \Rightarrow |V'| = c_v \cdot (c_q + c_d) \cdot 10,000 = c_{v'} \cdot 10^4$ nodes,
$|E'| = O(|\Delta|(|V| + |E|)) \Rightarrow |E'| = c_d \cdot (c_v + c_e) \cdot 100,000 = c_{e'} \cdot 10^5$ edges.

The graph we need to process contains thus some 10s of 1000s of nodes and 100s of 1000s of edges. If we assume 32 bits=4 bytes per node (for its ID), and 3*32 bits=12 bytes per edge (source node ID, destination node ID and edge ID in case of multigraphs), we need in total: $c_v \cdot 10^4 \cdot 4 + c_e \cdot 10^5 \cdot 12 \approx c_b \cdot 10^6$ bytes, or otherwise some Mbytes for storage. Even with an increase of one order of magnitude in the NFA number of states $|Q|$ in parallel with an analogous increase in the complexity of the NFA transitions $|\Delta|$, such a graph may be stored efficiently in any modern computer system since a Gbyte-RAM is apparently not the limiting factor[5].

The temporal complexity of the transform equals to[6]:
$t_{trans} = O(|V||Q| + |\Delta|(|V| + |E| + |Q|)) \Rightarrow$
$t_{trans} \approx c_{trans} \cdot 100,000 \approx c_{trans} \cdot 10^5$ time units.

Let us now focus on the spatial complexity in order to estimate the time t_{EK} that Edmonds-Karp would require for a single-pair path

[5] This assumes an optimized implementation for storing the graphs; in the case of the Python-based NetworkX library for example, the memory requirements may be one or two orders of magnitude larger due to the use of Python *dict* structures, but still acceptable for modern computer systems.

[6] Ignoring t_{dec} if we assume exact-form NFAs, such as the valley-free one.

diversity estimation on the transformed graph, and whether this time actually prevails against the pure graph transform time t_{trans}. We find that: $t_{EK} = O(|V'||E'|^2) \Rightarrow t_{EK} \approx c_{EK} \cdot 10^{14}$ time units, with $t_{EK} >> t_{trans}$. Assuming an optimized implementation of the algorithm, and a single CPU core with a frequency of $c_f \cdot 10^9$ Hz, each CPU time unit corresponds to about $1/c_f$ nsec. Assuming that a unit computation of the Edmonds-Karp implementation takes c_{cpu} cycles, we need about $c_{EK} \cdot (c_{cpu}/c_f) \cdot 10^5 = c_{EK\_tot} \cdot 10^5$ sec in total for running Edmonds-Karp in order to calculate the path diversity between a single pair of AS nodes on the transformed graph of the AS-level Internet. That is about 30 hours, i.e., approximately a day.

We observe that the latter time is quite large for a single pair-wise calculation, making for example an all-to-all calculation between all node pairs infeasible. To deal with this problem, we can first take the non-transit stub ASes out of the calculation, since they do not contribute to the max-flow/min-cut paths as non-transit nodes. These ASes are still used as endpoints for the min-cut calculations but do not contribute to any flow as intermediate nodes. This reduces the node set to less than 20% of its initial size, reducing the node complexity by one order of magnitude to approximately $c_v \cdot 10^3$ nodes. This brings t_{EK} down to a value of $c_{EK} \cdot 10^{13}$ time units, requiring $c_{EK\_tot} \cdot 10^4$ sec in total for running Edmonds-Karp per AS pair, i.e., approximately 3 hours. The complexity of the edge set does not change due to the nature of the stub AS connectivity; i.e., they are the contributors of a negligible percentage of the total edges, thus we cannot exploit any quadratic reductions at $O(|V'||E'|^2)$.

In order to deal with the all-to-all calculation, we can next perform a random selection of AS pairs and run the calculations on the sampled set. For example, in all our experiments we chose 10K pairs of ASes uniformly at random, and observed that a sample of this magnitude is quite representative of the actual behavior of the full graph. 10K pair-wise path diversity calculations require almost $10^4 * 10^4 \approx 10^8$ sec, i.e., approximately 3 years; this is still a prohibitive duration.

Therefore, as our final optimization, we distributed the pair-wise calculations in multiple 100s of cores using our institute's server cluster. This is feasible since each calculation is an independent operation that can be conducted in parallel with others. The parallelization brought the total time for the 10K pair estimation to a manageable period of

about 11 days, close to 1.5 weeks. We note that this time may be more elevated than the ideal one due to the use of Python and the corresponding graph management libraries. Moreover, if we use one order of magnitude more complex NFAs, the time needed is elevated in an analogous manner and more optimizations may be required to perform the calculations within a reasonable time frame. Scaling down the number of edges, besides the number of nodes, may bring the most notable reductions due to the 2-exponent in the Edmonds-Karp temporal complexity ($t_{EK} = O(|V'||E'|^2)$).

Furthermore, we note that a pure version of Ford-Fulkerson may also be used in order to calculate the maximum flow and the minimum cut, with the promise of faster calculations. In fact, when the capacities are integers, the run-time of Ford-Fulkerson is bounded by $t_{FF} = O(|E'|f)$, where $|E'|$ is the number of edges in the augmented graph and f is the maximum flow in the graph. However, this bound is dependent on the maximum flow that can be achieved between the candidate pair of nodes used for the calculation. Therefore, modern graph libraries such as NetworkX [190] use in practice the Edmonds-Karp algorithm, as a variation of the Ford-Fulkerson algorithm with guaranteed termination and a run-time independent of the maximum flow. Our analysis thus focused on this version that we also applied in this chapter and the investigation of the potential of CXPs (cf. Chapter 6).

5.5 Proof-of-Concept Application and Insights

Our algorithm can be used in different application contexts (cf. Section 5.2). One of these applications is the calculation of the policy-compliant inter-domain path diversity. In this context, we refer the reader to the application section of our associated INFOCOM paper [144], using the CAIDA AS relationship datasets [48] in order to estimate lower bounds on the physical inter-domain path diversity. Due to the form of the NFAs that were used in all cases, the results of the paper are exact and not simple approximations. We warrant though caution regarding the limitations of this dataset, as described in Section 5.2.1. In the following, we provide a summary of the high-level insights that were gained from a demonstration of this application in

the INFOCOM paper. We would like to note that the implementation of the experimental software and associated calculations were mainly performed by our fellow colleague Rowan Klöti with our collaboration.

First, we observed sizable customer cones[7] for the tier one ISPs, in the order of some 1000s of ASes. This observation, combined with the lack of diversity between those customer cones—due to valley-free policies that restrict route choice—leads to the following conclusion: any tier one depeering, i.e., a voluntary or accidental interruption of their peering relationship and connectivity, has the potential to cause major disruption for the direct customers. This is known to have occurred already [41, 240]. For large tier two ISPs, due to the rich peer-to-peer interconnectivity resulting in large path diversity, such depeerings seem not to be harmful. Actually, we have seen that the limiting factor for path diversity is often the number of peering and provider connections that the ISP itself maintains, rather than the Internet topology at large; this points to a densely connected Internet [71, 161]. An ISP which wishes to improve its connectivity can therefore either establish a business relationship with another upper tier ISP, or expand its peering. Many evidently choose the latter [57], with the proliferation of public IXPs [79] and open policy environments (see later paragraphs) lowering the barriers to entry for establishing new peering relationships.

Second, adding extra peering links—extracted from IXP peering datasets—makes little difference to the diversity of the valley-free scenario. More liberal scenarios, such as allowing the traversal of multiple peering links, instead of at most one, result in a considerably larger path diversity and profit a little more from the extra peering links.

Third, the peering policy of the ISPs (*Open, Selective or Restrictive*) matters. We observed that especially *Open* peering links increase the mean path diversity by about 10%. This is not surprising given that: *(i)* these are the most numerous, and *(ii)* the most likely to be missed by BGP route collectors—relationships between the largest ISPs are most likely to be faithfully represented in the AS relationship dataset. *Restrictive* and *Selective* links—between IXP members with the same respective policy—have smaller impact on path diversity.

[7] A customer cone is defined as the set of ASes that an AS—in this case a tier one ISP—can reach using customer links.

Fourth, depeering events involving tier-1 ISPs can cause an approximately 10% decrease in path diversity. On the one hand, if we allow the traversal of multiple peering links (see CXP model at Chapter 6), the addition of extra peering links boosts path diversity by about 20%. On the other hand, the valley-free model does not benefit at all from the addition of extra peering links. Furthermore, for the multiple peering link model we did not observe a significant drop in path diversity due to the depeering. Allowing multiple peering links can therefore potentially increase resilience in the face of a depeering. This lies in accordance with the insights of other researchers studying the relaxation of peering policies, e.g., on IXPs, such as Hu et al. [123].

We note that this approach was also successfully used in Chapter 6 in the context of generalized policies over IXP-based overlay graphs, as we will show in Section 6.3.5.

5.6 Comparison with Related Work

In this section, we compare our algorithmic approach to related work in the field of tensor products (cf. Section 5.6.1), network resilience (cf. Section 5.6.2) and policy-compliant min-cuts (cf. Section 5.6.3). We further introduce algorithmic extensions pertaining to shortest path calculations by generalizing related work efforts (cf. Section 5.6.4). Generally, the main contribution and point of differentiation of our work is the graph transformation algorithm as a simple approach having multiple potential applications and use case contexts.

5.6.1 Tensor Products

Soulé et al. [227] use a similar process to the one presented in this work in the context of network management. Their goal is to enforce bandwidth allocation policies subject to path constraints represented by regular expressions and consequently by DFAs/NFAs. They use tensor products in a different context than our approach, since we focus on path diversity and the implications arising in its preservation across transformations. In particular, we further propose the addition of aggregator nodes, acting as inhibitors to min-cut inflation.

5.6.2 Network Resilience

Previous research on resilient networks [55, 60] considers the network as a set of nodes and links, annotated with geographical properties. Consequently, researchers can calculate min-cuts, path distances or shared fate link groups [256] that are affected in a correlated fashion during a disaster. In reality, networks are run as policy-compliant administrative domains [47]. Therefore, the choice of paths that traffic can traverse is constrained in practice. The stricter—i.e., more limiting—the policy, the less maneuverability is available to the routing protocol in terms of discovering and using valid backup paths [226]. The view of the network as a geographical map cannot capture this behavior. Thus, we argue that network resilience [252] should also be estimated within a policy-compliance regime *besides* a location-based framework. Our approach enables exactly that, allowing to run vanilla min-cut calculation algorithms like Ford-Fulkerson on the transformed graph, with tight min-cut approximations under certain conditions related to the form of the policy-dictating NFA.

5.6.3 Min-cuts with Policies

Sobrinho et al. [226] describe a model for understanding the connectivity provided by route-vector protocols in the face of routing policies. Erlebach et al. [77] study valid s-t-paths and s-t-cuts in the valley-free model and prove on the one hand the NP-hardness of the vertex-disjoint min-cut problem. On the other hand, they prove that the edge-disjoint version can be solved in polynomial time for valley-free policies; we have verified this statement also in our framework[8]. Both works focus on specific aspects of the general problem (route-vector protocols and valley-free policies respectively), while we are delving into a more general methodology for min-cut estimations. Sobrinho et al. [226] examine the dynamics of a routing protocol with their work, while we focus on a general method to understand the effect of stable network policies on path diversity, ignoring for example the dynamics of routing convergence. For the latter, we refer the reader to our BGP-related work in Chapter 4.

[8] We have verified the last statement since the valley-free NFA can be mapped to a complete bipartite graph on G'; this leads to exact estimations of the policy-compliant min-cut in polynomial time (cf. Section 5.4.6), since the biclique decomposition in this case is trivial (cf. Section 5.4.5).

Teixeira et al. [236] study vertex- and edge-disjoint paths in undirected Internet topology models, but without taking routing policies into account for their investigation.

5.6.4 Extensions for Shortest Path Calculations

The graph transformation algorithm can be seen as a generalization of the special transformation used by Bauer et al. [27] for the calculation of valley-free shortest paths. We note though that our main contribution is the addition of aggregator nodes for yielding the correct values of the min-cut; extensions to shortest path calculations is just an additional benefit that we can exploit. In practice, our algorithm can be used for finding policy-compliant shortest paths as follows.

1. Extend the attributes of the edges E, by adding a delay function $d : E \to \mathbb{R}^+$ that maps edges to edge delays/latencies.

2. Perform all the remaining steps of the graph transform (cf. Section 5.4.3) without any changes; this will yield the transformed graph G', with the corresponding capacities. We can already run any min-cut calculation algorithm on this graph.

3. Properly set the delays on the edges of G', so that we can also run any shortest path calculation algorithm on this graph. We thus consider the following edge delays $d' : E' \to \mathbb{R}^+$ of G':

$$d'(((v_1, q_1), (v_2, q_2))) = \begin{cases} d(v_1, v_2) & v_1 \neq v_2 \\ 0 & v_1 = v_2 \end{cases} \tag{5.25}$$

   We note that ε-transitions (i.e., when "$v_1 = v_2$") should induce 0 delay so that they do not inflate the cumulative path delay.

4. Select any two mapped nodes in the transformed graph G' and run a classic Dijkstra-based algorithm to find the shortest path between them. Mapping the found path from G' to G is straightforward, as we can simply take out the state information from the nodes. The presence of aggregator nodes does not play any role in the overall calculation, since the path length is never distorted when we traverse them; ε-transitions leading to or stemming from them within the original graph nodes induce 0 delay. The correct

delay is taken into account when traversing an edge between two distinct graph nodes in G' (i.e., corresponding to different nodes in the original graph G). The traversed paths are always policy-compliant, as we have already proven in Section 5.4.4.

5.7 Summary of the Graph Transform

Path diversity and bisection bandwidth are useful metrics to describe how resilient or rich a network is. Network policies, imposed by network administrators and applied via routing protocols or network configuration, can constrain the natural path diversity of a network graph. This applies to both intra-domain and inter-domain cases; diverse client policies on the inter-domain level can for example affect path calculations on the global view offered to a multi-AS RaaS contractor.

With this work, we described and proved the correctness of a generic methodology for min-cut computations in arbitrary graphs, assuming policies that can be formulated using regular expressions. Our approach can be applied in a variety of scenarios, some of which are briefly showcased in this work. These include the investigation of Internet topology and alternative policy models in the Internet, effectively studying Internet-wide resilience and the effects of inter-AS connectivity on path diversity. We see further potential for our approach in the analysis of MPTCP flow path availability in data center networks, and path selection optimization in multipath flow routing applications. Achieving tighter bounds for the min-cut is another topic of interest. A detailed account of future steps in the context of the graph transformation algorithm can be found in Section 7.4.2.

In the following chapter, among other topics, we will delve into a practical application of the algorithm for the estimation of the AS-level path diversity of overlay IXP-based multigraphs.

Chapter 6

Use Case 2: Control eXchange Points and IXP Multigraphs

> Eventually everything connects—people, ideas, objects. The quality of the connections is the key to quality per se.
>
> —Charles Eames

How could the routing centralization approach be applied in the context of guaranteed end-to-end services over the current Internet, across domains, working in parallel with BGP? In this chapter, we present another use case for the architectural approach of Chapter 3, namely the provisioning of guaranteed end-to-end services using inter-domain mediators deployed on IXPs. In particular, we reduce the level of outsourced information that ISPs may provide to this new class of RaaS contractors, by abstracting themselves as sets of partial paths, with certain properties. We then study the associated IXP-based multigraph that is created from such paths connecting IXP-facing Points of Presence (PoPs). We investigate its potential as the data plane for inter-domain services in terms of client reach, path diversity, incremental deployment—both in space and time—and the performance of QoS-enabled path stitching algorithms using real peering datasets. The

content is based on our Technical Report submitted to ETH Zurich (D-ITET department, TIK laboratory) [153], our SIGMETRICS extended abstract [154] and our ONS extended abstract [150]. We also include some relevant parts that we contributed to the accuracy discussion of our joint work with Klöti et al., which is currently under submission [143].

6.1 Introducing Control eXchange Points

In this section, we first introduce the need for better end-to-end (e2e) Internet services across domains under certain guarantees, in light of ongoing changes in the Internet landscape (cf. Section 6.1.1). We continue with the basic contributions of this chapter in Section 6.1.2, and we give its outline in Section 6.1.3.

6.1.1 Motivation: Better Service, Across Domains

Modern Internet applications have increasingly tighter requirements for bandwidth, latency and/or availability [266]. For example, real-time HD video streaming [188], telemusic [73], remote control of critical infrastructure, such as power plants [44], or even telesurgery [120] are emerging or envisioned applications with strict network requirements. In fact more and more applications, such as the interconnection of stock exchange markets, need to operate under such guarantees stressing the limits of the original design of the Internet further and further.

Presently, ISPs are able to provide certain QoS guarantees [228] only in intra-domain settings based on technologies such as leased circuits and VPN tunnels, e.g., over MPLS (MPLS-VPNs). Such services provide an important revenue stream for ISPs [43,61,228]. However, despite several research and standardization efforts, providing QoS guarantees at the inter-domain level has seen very limited success so far [28,168,253,261].

During the last decade, an increasing number of proposals coming from diverse angles advocate centralized or distributed inter-domain routing brokers [78,148,162,229,241,263] as an approach to enable ISPs to cooperate and provide e2e guarantees for Internet services. In these schemes, ISPs provide QoS-enabled pathlets [108], which are stitched together by an inter-domain routing mediator. This mediator can be, e.g., a bandwidth broker [241], if bandwidth is considered as the primary QoS attribute to guarantee. Moreover, the stitching can take place either

using a distributed setup of multiple per-domain logical entities [229, 263], or a logically centralized setup of multi-domain entities [148, 263]. Commercial initiatives may be based on such frameworks [2, 132].

In a similar context, aggregate resource managers are currently investigated for multi-domain testbed experimentation. Examples of such entities are presented in the US GENI initiative [24]. Moreover, EU projects such as PACE [129] currently examine the transition from software-defined [183] concepts and mechanisms to standards, interoperability and deployment in the context of Path Computation Element (PCE) [82, 243] architectures. These architectures can serve as the basis on which such brokers and controllers are developed.

However, a necessary step before deploying such systems in the wild is a suitable analysis and characterization of the current Internet substrate (or overlay network) and its potential; this process should carefully take into account current ISP and Internet peering practices [168]. Therefore, in this work, we explore logically centralized inter-domain QoS routing mediators in light of the evolving Internet ecosystem.

We thus focus on public Internet eXchange Points (IXPs), which continuously rise in number and size [7, 57]. Presently, hundreds of IXPs around the globe connect thousands of ISPs together [79, 199]. As a result, the Internet becomes denser and more flat [71, 110, 161] over time, as the peering ecosystem expands in size. In parallel, the paradigm shift towards network virtualization [11, 224] and SDN [183] introduces new possibilities in network management and innovation, also in the context of IXPs. An example combining these approaches is the Software Defined eXchange (SDX) [117], advocating programmable IXP fabrics controlled by SDN controllers, with which the peering ISPs can communicate and exchange routing and policy information in a flexible manner. While SDX aims at achieving maximal impact starting from IXPs as individual entities, we believe that multi-site IXP-based deployments are natural next steps in the transformation of the peering ecosystem. Our main proposal is that a rich, flat, programmable overlay inter-IXP fabric opens interesting traffic engineering flexibilities which can be exploited, e.g., for inter-domain QoS. For example, as we will show later, the multiplicity of available paths on such a graph could be used for the provisioning of better availability upon fail-over events, more throughput in combination with multipath routing, or a larger range of available latencies depending on the application requirements.

6.1.2 The Contributions of this Chapter

Contribution 1: Stitching Paths over IXPs via CXPs

We propose performing inter-domain QoS routing over a novel abstraction of the Internet topology, in which vertices are IXPs and edges are virtual links connecting two IXPs over an ISP. The use of these edges can be used to extend valley-free policies, which dictate passing over at most one IXP per path. We call this abstraction the *IXP multigraph* because two IXPs can be connected with multiple edges over different ISPs. This abstraction hides the internal details of an ISP (including the diverse technologies that can be used to provide intra-domain QoS [263], such as leased circuits and MPLS [228]), and serves as a clean separation of concerns between intra-domain and inter-domain QoS routing that is consistent with the status quo. More importantly, the choice of IXPs as switching points exploits their rich connectivity, enabling high path diversity and global client reach with a deployment of only a few well-placed elements. We call the routing brokers that are deployed over IXP multigraphs, Control eXchange Points (CXPs); CXPs can generally use any switching point between ISP domains, with IXP-facing PoPs being the initial locations of maximum impact.

In comparison, current inter-domain routing is based on BGP, known to use suboptimal inter-domain paths[1], as they typically cross up to one peering link in an IXP due to valley-free routing policies [98]. For example, assume that an international corporation wants to connect two of its branches located in different continents over multi-domain connections, with high availability being the primary objective. Path redundancy and rich route choice are means to achieve this availability. Provisioned CXP paths could cross multiple IXPs if needed to ensure this. Whereas the use of multi-IXP links is not technically prohibited by BGP, paths violating the valley-free model are not common [102, 104]. CXPs enable the utilization of this additional path diversity, with this utlization being further enhanced by using multipath routing practices, escaping from the single "best path" model of BGP. In parallel, ISPs are motivated to apply their core competence of creating and maintaining guaranteed intra-domain tunnels to enhance the achieved diversity.

[1] Approximately 60% of all Internet paths today are subject to triangle inequality violations [16, 175], with unpredictable performance.

Contribution 2: Understanding the IXP-based Multigraph

We make a first characterization of the Internet's IXP multigraph and highlight its special properties, namely high path diversity and massive client reach. In this context, we analyze ISP membership data for 229 IXPs using a snapshot of the Euro-IX dataset [79]. We show that by placing CXP deployment points within a few large IXPs, we can access a significant part of the IPv4 address space. In particular, 40% of the announced IPv4 addresses are accessible via only 5 CXP data plane elements in well-connected IXPs. This increases to 91% if we also consider the 1-hop customers of the IXP members. Second, we find 49K unique inter-IXP pathlets, i.e., inter-connections of type: $\mathrm{IXP}_A, \mathrm{ISP}_X, \mathrm{IXP}_B$. These are virtual links which are likely not used currently as parts of e2e paths, since such paths do not conform to valley-free conditions. We show that by relaxing valley-free constraints on such links and enabling their active use, CXPs can greatly increase path diversity by up to *29* times, as opposed to the current setup. Path availability and access to clients can thus benefit highly from IXP-based deployments; we also highlight additional reasons for the suitability of IXPs in this regard. Furthermore, we corroborate our findings on the edge-wise path diversity of the Euro-IX-based multigraph and its properties by performing a temporal analysis of PeeringDB [199] over a 7-year period. We find that our observations hold *over time*, with the edge multiplicity of the IXP multigraph consistently leading to an order of magnitude larger edge-wise path diversity over inter-IXP links.

Contribution 3: Path Stitching Algorithms

We present algorithms to efficiently exploit the high path diversity observed in the IXP multigraph. In particular, we study the problem of maximizing the number of concurrently embeddable paths, subject to bandwidth and latency constraints. We describe online as well as hybrid online-offline algorithms which sample feasible paths efficiently (i.e., in polynomial time). These algorithms achieve different trade-offs between optimal acceptance ratios and fast online computation, with the hybrid approach realizing a balance between the two goals by reallocating paths in the background based on an optimal offline algorithm. Using simulation, we show that our algorithms scale to the sizes of the measured graphs and derive insights on which variant(s) should be leveraged to

serve diverse request mixes. The relevant performance metrics that we examine are acceptance ratios, substrate utilization and embedding runtimes. This algorithmic framework can serve as the application logic of logically centralized inter-domain brokers, operating on the global view of IXP multigraphs. We note that the full simulation framework, the multigraph processors based on the peering datasets, and the analysis code for determining properties of these multigraphs (such as path diversity) are publicly available as open-source software [152].

6.1.3 Outline of this Chapter

The rest of this chapter is structured as follows. Section 6.2 provides the needed background on inter-domain QoS brokers, CXPs and the motivation behind our IXP-based approach. Section 6.3 maps the IXP multigraph of the Internet, as derived from a Euro-IX dataset snapshot, and characterizes its high path diversity and client reach for inter-domain QoS and guaranteed e2e services. Furthermore, the path diversity characteristics of the multigraph are studied over time using PeeringDB snapshots, supporting the stability of our claims. Section 6.4 presents algorithms for embedding flows in paths crossing dense IXP multigraphs. Section 6.5 evaluates these algorithms based on a custom simulator using real IXP multigraphs. Section 6.7 bridges our IXP-centric analysis and methodology with emerging use cases, such as remote telesurgery over IP. Section 6.6 presents related literature and the comparison of our work to these research efforts. Finally, we provide some outlooks and conclusions related to the CXP concept in Section 6.1.2.

6.2 Service Brokers, IXPs and CXPs

In this section, we first give an overview of previous research on centralized path and resource brokers for guaranteed inter-domain services (cf. Section 6.2.1). Second, we discuss why IXPs are suitable locations for deploying the data plane elements of such brokers (cf. Section 6.2.2). Last, we describe in detail the properties and potential implementation mechanisms of our proposed IXP-based path brokers, which we call Control eXchange Points (CXPs), in Section 6.2.3. CXPs are another instantiation of the architectural approach described in

Chapter 3, and work independently from the BGP-compatible RaaS controllers presented in Chapter 4.

6.2.1 Network Service Brokers

The community has performed research on bandwidth brokers for mediating the concatenation of multiple tunnels with guaranteed bandwidth among ISPs (e.g., MINT [241]), or for scaling up the provisioning process for guaranteed bandwidth services within an ISP network (e.g., the work of Zhang et al. [263]). Similar initiatives have created bandwidth markets and commercial brokers, such as InvisibleHand [132] or Geant's multi-domain Bandwidth-on-Demand (BoD) service [2]. Other proposals have introduced "route bazaars" between ISPs and end-users [78], where pricing mechanisms and interactions between demand (end-users) and supply (ISPs) directly affect routing and path establishment. Routing-as-a-Service controllers [162] that are based, for example, on the idea of outsourcing routing functionality [148], have been proposed as potential broker models and corresponding implementations. Such controllers can deal with end-to-end path stitching using their bird's eye view over the participating domains. Dynamic traffic management applications can then operate on this global view, based on appropriate interfaces for the programmable control of the network.

Moreover, centralized routing controller platforms based on the Path Computation Element (PCE) architecture [82, 243] have been evaluated in the context of QoS routing schemes for high capacity optical networks [100]. We note that the initial multi-domain intention of the PCE initiative was to help coordinate path establishment requests, and to be able to compute an end-to-end path using cooperative per-domain PCEs. Systems like PCE are highly relevant for the implementation of brokers and routing controllers, e.g., applied on IP/MPLS domains [229], and are backed up by IETF standardization efforts [82, 243].

In this work, we investigate deployment possibilities for service brokers/controllers, assuming an underlying system's base such as PCEs and client routing elements, or SDN controllers and programmable switches. We thus look at potential locations for placing the data plane elements, investigating relevant properties of the overlay graph, and the resulting client reach and impact for inter-domain services.

6.2.2 Deploying Service Brokers on IXPs

Brokers and controllers for guaranteed e2e services across domains need to exert inter-domain control through programmable data plane elements, such as OpenFlow switches[2]. We call these elements *anchors*, since they "anchor" inter-domain traffic switching to specific locations, decoupled from the traffic management within, e.g., ISP domains. The ideal anchor is adjacent to multiple geo-diverse ISPs, is provisioned for high bandwidth and availability, and is independent from a single ISP. We observe that IXPs have all these properties and thus provide ideal starting points for deployment.

One aspect that needs to be clarified early at this point is whether IXPs would be truly open to host such devices in order to provide new services to their members. According to Chatzis et al. [57], IXPs are presently the hubs of multiple services involving their members; these services surpass their initial goal of pure layer-2 switching fabrics for handling the offloaded traffic of ISPs. Hosting route servers for ease of BGP-based peering [14,214], mobile peering with 3G providers for traffic convergence [15], or adjusting and evolving their control and data plane towards SDN approaches for new inter-domain applications [117]—such as application-specific peering—are just a few examples.

Modeling IXPs as vertices and inter-IXP pathlets as edges, the resulting topology is a dense multigraph: two IXPs can be connected via multiple ISPs. This is quite common because many ISPs are present at multiple different IXPs in parallel (cf. Section 6.3 for details). We base our case study on this simple yet powerful observation, enabling us to build a novel IXP-centric abstraction of the Internet topology. Endpoints can connect to this topology via pathlets offered by their access providers towards adjacent IXPs (see Figure 6.1). The topology can be built and measured based on available peering datasets, such as Euro-IX [79] or PeeringDB [199], as we will show in detail in Section 6.3.

6.2.3 CXPs

Following the observation that IXPs provide ideal locations for data plane anchors, we introduce *Control eXchange Points* (CXPs), i.e.,

[2]These devices can be based on simple *match-action* primitives. For example: "Match on incoming traffic flow space", "act on the associated packets of the flow (forward, drop, rewrite, encapsulate and put in tunnel, etc.)".

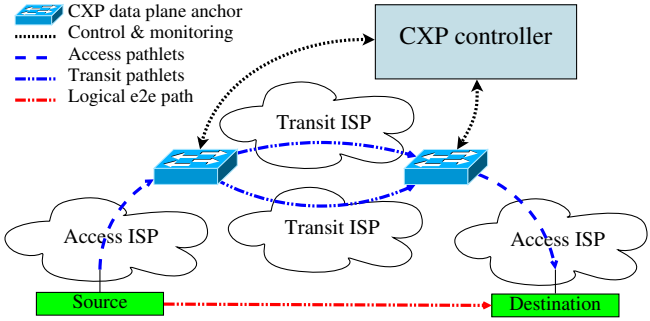


Figure 6.1: The CXP stitches QoS-enabled e2e paths over an overlay network of data plane anchors (e.g., IXPs), and peering ISPs.

control points which enable stitching of pathlets across multiple administrative domains to construct global paths. These pathlets are provided by ISPs; these ISPs allocate the pathlets at their own capacity within their domain, but exchange control with the CXP for efficient path stitching at the inter-domain level. In this section, we discuss which existing or new technologies a CXP could rely on for implementation, both in the control and the data plane. A complete implementation is though out of the scope of this work; here we describe the engineering practices that could be employed for a potential CXP-oriented project.

Basics

A CXP is a logically centralized entity, applying inter-domain control over how parts of Internet traffic are routed. In this context, it can, for example, provide e2e QoS or support multicast services by selecting (a multitude of) appropriate paths that satisfy the service requirements. A CXP works in parallel to traditional routing and can control parts of traffic independently from BGP, e.g., utilizing flow space isolation mechanisms [11,224]. CXPs use programmable data plane anchors which switch traffic, such as SDN switches [183]. Recent advances on scalable packet classification may be applied on these devices for achieving the required scale for traffic processing [248].

Furthermore, Software Defined Internet eXchanges (SDX) as proposed by Gupta et al. [117] could constitute an IXP-based deployment possibility; the transition from single-site SDXes to multi-site SDXes involving several IXPs could be accelerated based on the CXP concept. CXP control planes may also be built using PCEs [82]. PCEs can reduce the required inter-domain signaling, enforce traffic access policies and hierarchically manage multi-technology domains. Moreover, a potential cooperation between IXP Route Servers [214] and PCEs, via well-established IETF protocols such as PCEP [243], could enable CXPs to respond dynamically to changing needs over a set of IXP-mediated inter-domain connections. Besides public IXPs, which are very useful as starting points of maximal impact, CXP anchors can be deployed at private peering points for augmenting geographical coverage, if required.

Providing Pathlets

Between CXP data plane anchors, traffic is shipped on virtual links which are parts of e2e paths and act as pathlets [108]. Pathlets are provided by ISPs and are annotated with specific properties, such as bandwidth and latency guarantees if QoS is to be supported, with simple connectivity as the baseline. When a client requests an e2e path, the CXP has to find a suitable sequence of pathlets that meet the client's QoS requirements e2e across the involved domains.

Pathlets can be provided by ISPs with existing tunneling techniques, such as MPLS, GRE and VPNs, or emerging SDN approaches based on flow space allocation along a network path [11, 183]. Inside the ISP, QoS guarantees are provided via traffic engineering and prioritization techniques [22, 263], such as the ones employed for DiffServ [83]. The reader is referred to the work of Hunt et al. [125] on QoS over IP-based networks using MPLS-TE for a comprehensive review of the associated mechanisms in the context of the MPLS protocol.

On the one hand, the ISP is responsible for providing cross-traffic isolation internally, keeping its management policies confidential. On the other hand, the CXP provides traffic isolation on the data plane anchors. An ISP may provide multiple pathlets between two data plane anchors with different properties for service differentiation or fail-over. We note that CXPs do not have control over how *physical pathlet redundancy* is achieved within the ISP. Availability properties (e.g., for telesurgical applications [120]) should therefore accompany the

ISP-originated pathlet advertisements. One way to achieve this is by annotating pathlets with Shared Risk Link Group (SRLG) IDs [256].

The incentive for ISPs to provide pathlets is the revenue generated by their use for e2e services; any ISP can be a provider. This is an important observation on which we will later capitalize, in order to explain why and how we can relax the classic inter-domain valley-free policies [98] using CXP-mediated paths. As shown in Figure 6.1, the ISPs of the source and the destination endpoints offer access pathlets to connect them to ISP-adjacent data plane anchors (i.e., the peering IXP switches), while the intermediary ISPs offer transit pathlets over their domains, between anchors. This partly resembles the status quo (access and transit providers), with the important difference that the liberation from BGP constraints[3] and the extension of classic policies may provide much larger path diversity and route choice, as we will show in Section 6.3.

CXP Tasks

The CXP: *(i)* handles new requests for QoS-enabled paths (admission control), *(ii)* computes and sets up suitable paths (embeddings), *(iii)* monitors pathlet availability and compliance with QoS guarantees, and *(iv)* performs re-embedding, if required. A client negotiates her request directly with her access ISP, which selects a suitable CXP for establishing the inter-domain route out of a set of available CXPs. The ISP forwards the client's request to the chosen CXP which in turn computes a suitable e2e path. The CXP reserves capacity on the selected pathlets and then configures the respective data plane anchors. Accordingly, the client's ISP has to configure its network such that the quality sensitive traffic is sent via one or more pathlets to the correct data plane anchor(s). A CXP monitors the bandwidth, latency and availability of a path for the duration of the client's reservation [194, 204, 247]. If the client's requirements are violated or a pathlet becomes unavailable, the CXP chooses and configures an alternative path for the affected part(s) of the traffic; this can even be a pre-configured "hot-standby" backup path carrying traffic duplicates (cf. Section 6.7). Besides, the CXP may choose to better

[3]For example, CXPs can route on a flow space granularity as opposed to the classic prefix-based BGP routing.

utilize the available pathlets by re-embedding paths and defragmenting the substrate resources, such as the allocated bandwidth.

Technical Challenges and Trade-offs

Regarding the proper addressing of the CXP tasks, we mention the following challenges and associated trade-offs. *(i)* Controller distribution and placement [122]; the RTT between the data plane anchors and the centralized CXP controllers is a lower bound of the reaction times to failures or state updates [211], while full distribution induces state consistency and concurrency challenges [167]. *(ii)* Forming an accurate, real-time monitoring infrastructure for supervising pathlet guarantees and measuring the performance of QoS-constrained flows is a challenging task in its own right [194]. Nevertheless, CXPs need to control just a handful of IXP anchors around the world, which is a promising starting point. Also, the complexity of pathlet formation and state monitoring is delegated to the ISP. For example, physical link failures that affect pathlets are first handled locally within the ISP and then propagate on the inter-domain level only if the failure needs to be known to the CXP to be handled via e2e re-routing. *(iii)* Path embeddings need to be protected against failures via CXP controller and anchor redundancy.

Regarding the security of a potential system deployment, we identify the following attack vectors and possible solutions. *(i)* Clients could request high-QoS paths, without actually needing them. This does not hurt ISPs since clients have to pay for what they request; moreover adaptive pricing i.e., high prices on scarce, valuable pathlets would disincentivize such clients. *(ii)* An attacker may try to reserve bandwidth in a coordinated fashion aiming at depleting the substrate. This is a difficult attack to carry out, since the rich path diversity of the substrate (cf. Section 6.3) requires high attack budgets. *(iii)* An attacker may request many hard-to-compute embeddings and overwhelm the controller, possibly keeping the allocations only for a short time to save money. This attack becomes less attractive by asking for a per-request fee or demanding a minimal request duration. *(iv)* Further Denial of Service attacks against the controller need to be mitigated using control plane redundancy or protected CXP-anchor channels.

We stress that similar topics are active within the SDN research community and that the CXP model can benefit from developments in these areas (e.g., distributed controllers and NOSes [146]). For the rest

of this chapter, we will capitalize on a simplified simulation of a CXP system focusing on two primary axes: *(i)* the data plane on which it may operate, and *(ii)* the application logic (i.e., routing/path stitching algorithms) deployed over the system. We refer the reader to future research and engineering directions in terms of deploying such a system in the wild in Section 7.4.3.

Privacy Concerns

The CXP scheme is based on the advertisement of ISP-originated pathlets with certain properties to a trusted entity, so that this entity can stitch end-to-end paths to serve client endpoints at the edges of the multi-domain network. However, there are certain privacy concerns that are related to this scheme. In the following, we describe some cases where privacy may be violated due to the use of CXPs.

One concern is that an attacker can explore the CXP system behaviour over time and may thus combine knowledge learned from different demand matrices. For example, assume that an ISP advertises a pathlet between IXP_A and IXP_B, with maximal latency L sec and minimal bandwidth B Gbps. A malicious end-user can infer the following information: *(i)* The ISP has PoPs at both IXP_A and IXP_B and is a member of these two IXPs. *(ii)* The peering ports at IXP_A and IXP_B have at least B Gbps of capacity. *(iii)* By correlating the offered latency with the speed-of-light latency between IXP_A and IXP_B the end-user can deduce whether the offered pathlet is subject to an ISP-internal triangle inequality violation, and whether there is any congestion within the ISP or at the peering ports that may affect the level of the guaranteed latency. The external user can then observe this information over time and for multiple ISPs at different peering points, as well as the stitched end-to-end paths that the CXP chooses to use for forwarding his/her traffic under certain guarantees, and infer detailed information about the IXP-based overlay graph and its properties.

Moreover, reverse engineering of the internals of the ISP network, e.g., to uncover fate-sharing on intra-ISP links based on external observation of the advertised edge-to-edge pathlets, can be potentially applied in the CXP case. Such reverse engineering can be based on general methodologies to externally infer internal ISP behavior—e.g., using causal inference [234]—but is not a limitation of the CXP case in particular.

We note that dealing with the aforementioned privacy-related challenges is the subject of future work, associated to the technical implementation of CXP systems and to political decisions related to the operation of such systems in the wild (cf. Section 7.4.3).

6.3 Understanding the IXP Multigraph and its Potential

In this section, we measure and characterize the inter-IXP multigraph, i.e., the overlay substrate on which CXPs may operate. This analysis may be particularly useful for prospective SDN-based CXP deployments, since it is a necessary step to understand where inter-domain control could be applied, as well as the potential for incremental deployment and the resulting impact. This is orthogonal to research related to scaling up CXP-like control planes [33, 146] or investigating the trade-offs involved in logically centralized controllers [167]. We start with a brief analysis of the peering datasets that we use, both in terms of accuracy and completeness (cf. Section 6.3.1). We next map the IXP multigraph based on Euro-IX data [79] in Section 6.3.2, and investigate its properties in Section 6.3.3. Afterwards, we answer the following questions: *(i)* how many IXPs need to participate so that CXPs can provide guaranteed services to a large population of the Internet (cf. Section 6.3.4), assuming that their member ISPs would offer the required pathlets, and *(ii)* how much path choice and diversity we can gain on the AS level, compared to classic BGP routing practices (cf. Section 6.3.5). We highlight this because currently, due to valley-free routing [98] and the prevalence of peer-to-peer links at IXPs [7], Internet paths typically cross at most one IXP. CXPs simplify the use of paths that cross multiple IXPs (e.g., for e2e QoS provisioning), thus providing the motivation for studying the associated multigraph structure and its potential. Finally, we perform a brief temporal analysis of the IXP multigraph based on available PeeringDB snapshots (cf. Section 6.3.6), in order to verify that our observations pertaining to the structure of the multigraph (e.g., number of nodes and edges), as well as aspects of its potential (e.g., edge multiplicity and edge-wise path diversity), are valid *over time*.

6.3.1 Datasets and Artifacts

Which Data we Used in This Work

In this work, we use four datasets to map the inter-IXP topology and the IPv4 address space: *(i)* the Euro-IX [79] and *(ii)* PeeringDB [199] databases, from which we obtained IXP membership data, *(iii)* the CAIDA AS relationship data [48, 174] for inferring policies on AS-AS links, and *(iv)* the CAIDA RouteViews AS-to-prefix data [51] for mapping AS-based organizations to IPv4 prefixes. Primarily we report results for Euro-IX, which also provides geographic coordinates of IXPs (used to determine distances between IXP locations in Section 6.5) in contrast to PeeringDB. Analysis on PeeringDB data further corroborates our findings; we use the richness of PeeringDB snapshots at our disposal for performing a temporal analysis of the edge multiplicity and path diversity of the IXP multigraph at both a monthly and yearly granularity in Section 6.3.6, deriving insights about its behavior over time. Next, we comment on the available peering datasets and the problems we faced while investigating them.

Available Peering Data Sources

We are aware of five different sources for acquiring data related to the Internet peering ecosystem, such as membership information: *(i)* Euro-IX [79], *(ii)* PeeringDB [199], *(iii)* Packet Clearing House (PCH) [198], *(iv)* the websites of the IXPs themselves [14], and *(v)* BGP dumps—e.g., BGP summaries for IPv4 prefixes, including log files—from looking glass servers [34] and route collectors [198] scattered across the Internet.

Euro-IX, PeeringDB and PCH

As a first step, we focus primarily on three sources and content providers of peering information, namely Euro-IX, PeeringDB and PCH. A detailed characterization, mapping and comparison of these datasets can be found in the work that we co-authored with Klöti et al. [143]. In that work, we performed an analysis and cross-comparison of the three datasets. In particular, we *(i)* compared three rich IXP datasets in order to assess their strengths and weaknesses, and *(ii)* combined them in order to improve the completeness of the publicly available IXP data. The combination was based on the union of the datasets,

after they were properly linked—i.e., their entities were mapped across datasets—based on their identifiers and associated information. We will see later that estimating the accuracy and completeness of such datasets is a very difficult challenge, while the unified linked dataset is not yet ready to be used for arbitrary experimental research on IXPs.

The results show that the three datasets have similar geographical coverage, with PCH having many more IXPs, but also many inactive ones, while being relatively poor in terms of available peering information[4]. In addition, PeeringDB seems to have an AS-centric bias in terms of peering information characteristics, while Euro-IX has an IXP-centric bias due the nature of the self-reporting methodologies used by the two providers. Besides, PCH includes very little information about IXP members as mentioned before in comparison. Furthermore, we showed that the datasets have partially common as well as rich complementary information; none is though complete. With respect to complementary, we showed for example that by linking the datasets we can increase the number of IXP records by ∼40% compared to using solely PeeringDB. Even more complementary information is available for IXP memberships, which previous studies have shown to be incomplete in PeeringDB [171]. For more details on the characteristics and implications of the unified (i.e., mapped and linked) dataset, we refer the reader to the corresponding work [143], which is currently under submission. We will examine later the problem of data accuracy, related to both the individual and combined datasets.

IXP Websites

We leave the fourth source of information (IXP websites) out of our study for the following reasons. First, the ratio between manual labor required for detailed investigation and verification, to the value of the corresponding reward (i.e., partial ground truth), is quite low. We note though that it has been used in other research works [20] albeit in a limited context, while its use does not scale for performing, e.g., a time analysis across multiple snapshots (as we do in Section 6.3.6). Moreover, many IXP websites might not be used as an independent verification method due to the fact that in some cases, like the ones

[4]The number of provided *(IXP,ASN)* peering links was about 65% less than the other two datasets.

of DE-CIX and LINX, the websites are automatically updated via PeeringDB, inheriting the associated bias. In other cases, such as the rising IXPs in Brazil and Argentina, information is available only in local languages (see also Asian-Chinese IXPs), which further complicates any verification efforts. We note also that not all IXPs provide or maintain websites. For example, out of almost 500 IXPs in PeeringDB, more than 80 don't provide a website at all, or the provided URL cannot be reached. Even if the website is reachable, finding a membership list requires overcoming the language barrier, if such a membership list is provided at all, as explained beforehand. Finally, it is difficult to automate this process; in some cases, membership lists do not include AS numbers, but just organization names—which are extremely hard to match automatically. And last but not least, the web crawling approach cannot give guarantees on completeness either, because some IXPs may not be known to exist. We thus warrant caution when working with such data; current initiatives spearheaded by Euro-IX aim at tackling these challenges using a single federated worldwide database [3]. Finally, some of our collaborators' ongoing work on partial verification using data available on IXP websites shows promising results in terms of accuracy, both for the biggest IXPs and for IXPs that are randomly selected from the combined pool of available IXPs.

BGP Data and the Quest for Accuracy

The fifth source of information (looking glasses and route servers) can be used for extracting peering data, but only for verification in a limited number of cases. This is because the vast majority of peering links does not appear on BGP paths collected on the route collectors except if the collector communicates directly with the route server of the IXP; the location of the vantage point is crucial. The problem of finding the "missing links" is well-known within the research community and its importance was pronounced also in the work of Ager et al. [7][5]. *Complete* public data from IXP route servers is generally not available, since the relevant BGP dumps can be found for only a small number of IXPs [34, 198] (∼15% of the IXPs available in Euro-IX and PeeringDB datasets). Nevertheless, BGP data can be taken into account for verifying accuracy

[5]The authors used closed data from a big European IXP to show that the number of p2p links, which appeared within this IXP, exceeded the total number of p2p links that was speculated to exist within the entire Internet at that point in time.

properties of the used datasets, when applicable. In fact, we have performed such a verification as an independent contribution in the work of Klöti et al. [143], with respect to a snapshot of the datasets.

We note that previous research has evaluated the accuracy of specific attributes of the PeeringDB data. In particular, Snijders [225] used 256 BGP session summaries submitted by network operators as ground truth to evaluate the accuracy of the peering IP addresses reported in PeeringDB. They could parse 96.5% of the IP addresses and 99% of these were correct. In addition, they found that the completeness of the IP addresses with respect to the ground truth was 75%. Lodhi et al. [171] used data extracted from the websites of the 20 largest IXPs to evaluate the completeness of the IXP participant information in PeeringDB. They found a median of 80% completeness, with significant differences though between regions; e.g., the completeness was only 25% for the Moscow IX. These studies indicate that although the studied PeeringDB data are not complete, they are mostly accurate. This is also indicated by the fact that PeeringDB is actively used by many network operators. In the case of Euro-IX though, we have received the appropriate assurance regarding the accuracy of the membership information by direct communication with Euro-IX staff.

We note that in general, accuracy encompasses completeness, i.e., to what extent IXPs and their members are represented in the dataset, and correctness, i.e., to what extent the IXP and membership information in the dataset corresponds to reality. Evaluating accuracy is very challenging because of the lack of ground truth for the different information about IXPs in the datasets and also due the diversity of the IXP attributes (and their interpretation) that the datasets include. In the following we report on our main results in the context of IXP membership.

Accuracy: Comparing with BGP Data

We extracted IXP participant information from BGP summaries collected by PCH at 77 of their route collectors [198] to compare and evaluate the completeness of the participant information in all datasets, including the linked one (i.e., the union of Euro-IX, PeeringDB and PCH; see also Klöti et al. [143]). The BGP data include information about established sessions with BGP peers over the IXP, in contrast to the partially self-reporting origins of the other datasets. PCH tries to

openly peer with all other IXP participants. Still, the data may miss participants who do not select to peer with PCH. We assume that all peer ASes seen by the IXP route collector, peer over the IXP fabric. To verify this, we manually scanned the next hop IPs and ASNs within the summary records to determine which ASN is actually peering at the related IXP. We used BGP data collected on the 19th of Sept 2014, i.e., the same date as the other datasets, and linked the IXP identifiers of the 77 PCH BGP route collectors with the IXP identifiers in the other datasets using AS membership and IP address information. The latter was done in order to have a basis for comparing these datasets. The route collectors contain location information in their name (typically an airport code), which we utilized for further verification.

In Table 6.1 and Table 6.2 we report the number of IXP-to-ASN links by dataset for the 77 IXPs with BGP route collectors, and the Jaccard and overlap similarity between the reference BGP data and the four other datasets, respectively. First, we find that approximately 72% of the BGP IXP-to-ASN tuples are reported in the linked dataset, while the corresponding figure for the individual datasets ranges between 61% and 73%. These numbers are close to the completeness figures reported by Snijders [225] and Lodhi et al. [171] for (solely) PeeringDB using different datasets and methodologies. The validation datasets used in the different studies are subject to selection bias, i.e., bias due to the IXPs or/and ISPs that provide useful information for validation, which could explain the small differences. Indeed, looking at our own validation set of 77 IXPs we find that PeeringDB, PCH and Euro-IX datasets are in larger agreement for this validation set than for the overall comparison (cf. Klöti et al. [143]).

Moreover, we find that in particular Euro-IX and PeeringDB include many IXP-to-ASN tuples, which are not present in the BGP data. This indicates that the BGP data is not complete, either. In particular, the route collectors report only approximately 56% of the membership contained in the databases. The underlying reasons include the fact that not all IXP participants may be willing to peer with a route collector, leading the collector to miss some of the peering links, and that the databases may contain stale data. We conclude that the figures presented on dataset completeness are only an upper bound of the true membership data completeness in the 77 IXPs. This indicates that although linking can significantly increase the completeness of the

Number of links				
BGP	UNI	EIX	PDB	PCH
6,425	8,121	6,087	5,749	3,547

Table 6.1: The number of IXP-to-ASN links by dataset for the 77 IXPs with BGP route collectors; Union (UNI) denotes the linked dataset containing PeeringDB (PDB), Euro-IX (EIX), and PCH.

BGP/UNI		BGP/EIX		BGP/PDB		BGP/PCH	
J(%)	O(%)	J(%)	O(%)	J(%)	O(%)	J(%)	O(%)
46.1	71.5	42.2	61.0	45.1	65.8	35.3	73.4

Table 6.2: The Jaccard (J) and overlap (O) indexes between each dataset and the ground truth IXP-to-ASN links extracted from the 77 BGP route collectors. Union (UNI) denotes the linked dataset containing PeeringDB (PDB), Euro-IX (EIX), and PCH.

available data, there is still a long way to go to completely map the IXP ecosystem of the Internet and great caution is required.

Finally, due to the large number of deprecated IXPs in the PCH dataset, and its small membership information, we focus on the Euro-IX and PeeringDB data for the rest of our work. We examine them separately due to the unverified accuracy and completeness of the combined dataset (or "union"), as described beforehand. The formal verification of the combined dataset is ongoing work. In general, none of the provided datasets is complete. This means that the results the reader will find in the rest of the chapter should be treated under the light of a lower bound approximation of the true potential of our proposal. Forming an accurate, complete view of the peering ecosystem is the subject of other ongoing initiatives [3].

6.3.2 Mapping the Inter-IXP Multigraph

Using a snapshot[6] of the Euro-IX peering database [79], we extracted membership data for 6,542 ASes in 277 IXPs. After ignoring IXPs

[6]The Euro-IX snapshot was taken on 9/4/2014.

which were ambiguous in terms of naming[7], had no members or had only members which advertised no IP prefixes, we have 6,122 ASes in 231 IXPs. Two further IXPs which have no connections to others are discarded. The final (connected) graph consists of 229 IXPs and ∼49k edges between IXPs, crossing ISPs that peer concurrently with these IXPs. This graph is the largest connected component of the full (disconnected) graph. We note that this pre-processing step is important since some IXPs intentionally form highly localized ecosystems (e.g., in countries like Vietnam as we observed), with their membership base consisting of ASes that peer only at these IXPs and with noone else. Such IXPs and their participants cannot contribute useful nodes and pathlets to the CXP multigraph. We are thus looking at graphs that are *connected* based on AS-traversing inter-IXP pathlets, in compliance with the CXP model.

We derive simple graphs by collapsing multi-edges to single edges, annotated with the initial edge multiplicity. At this point, we consider undirected edges/pathlets since we simply wish to study the properties and connectivity of the multigraph rather than the flow embedding problem; later in our algorithmic simulations (cf. Section 6.4 and Section 6.5) we will use single-directional pathlets that carry one-way traffic stemming from a source endpoint directed to a destination endpoint. The use of directed vs undirected graphs in our case simply affects the scaling of the edge multiplicity and path diversity results by a factor of 2, since one undirected edge is mapped on two single-directional pathlets.

Ideally, the CXP multigraph should contain all possible IXPs and their participants for maximum coverage and service capabilities. This cannot happen at one go; in practice, incremental deployment is essential. We thus scale down the extracted full inter-IXP topology assuming that a CXP does not have all the IXPs at its disposal, but gradually recruits IXPs to maximize the IP address space it can serve. Each new IXP provides access to more client address space served by its member ISPs. We will elaborate more on the impact of incremental deployment in terms of IXPs and served addresses in Section 6.3.4. We determine a suitable order based on a greedy heuristic, starting with the IXP having

[7]Initially the names of the IXPs were used for indexing; in later versions of our processing and analysis software we have integrated id-based indexing to avoid problems with ambiguities stemming from the names.

the largest address space coverage and in each iteration adding the IXP which yields the greatest number of non-overlapping addresses. This allows us to study the evolution of the multigraph *in space*, and to examine different snapshots at different levels of CXP penetration. We assume that whenever we add a new IXP, all its member ISPs would host pathlets that: *(i)* connect their edge clients to the new IXP (via *access* pathlets, cf. Figure 6.1) , and *(ii)* connect the new IXP to other CXP-enabled IXPs at which these ISPs are present in parallel (via *transit* pathlets, cf. Figure 6.1).

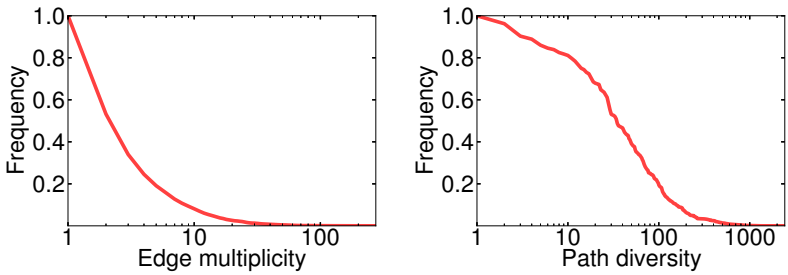
We make the latter assumption, since our goal is to investigate the potential of an IXP-centric multigraph for CXP deployment, as the CXP approaches more and more IXPs. Each IXP is associated with an ISP membership base, which we want to examine in full. The dynamics of the pathlet market will eventually determine which IXPs and ISPs will participate, which pathlets they will advertise and which clients will choose to connect under diverse QoS guarantees (e.g., bandwidth or latency levels). For such market analyses, investigating pathlet pricing and ISP participation, we refer the reader to works such as MINT [241] or RouteBazaar [78]. We also argue that the underlying market is interesting enough for many ISPs to participate in, as supported already by proposals investigating the economics of transit pathlet markets [241]. We thus leave the examination of gradual CXP penetration in terms of ISP attraction for future work, and focus on the incremental deployment of CXPs from the perspective of the recruited IXPs.

6.3.3 Properties of the Inter-IXP Multigraph

Table 6.3 gives an overview of the properties of the inter-IXP multigraph at diverse scales, according to the multigraph formation process described in Section 6.3.2. The scale-down factor 32 corresponds to a small potential CXP deployment on 7 IXPs, while a factor of 1 involves all the 229 IXPs of Euro-IX. We first observe average shortest path lengths between 1 and 1.92 edges. This observation, combined with the high clustering factors, suggests small world properties. Furthermore, multi-edges result in very high average node degrees, e.g., of 160 even in just the 7 IXP topology. Figure 6.2a shows the Complementary Cumulative Distribution Function (CCDF) of the edge multiplicity, i.e., the number of parallel ASes that connect pairs of IXPs, in the full (unmodified) topology. We observe that a few pairs of IXPs are

	Scale-Down Factor (SDF)					
Property	1	2	4	8	16	32
Node count	229	115	57	28	14	7
Edge count	49k	29k	15k	6.5k	3.9k	1.1k
Diameter	5	5	3	2	2	1
Av. node degree	220	250	260	230	280	160
Av. edge multiplicity	4.3	6.0	8.3	12.	25.	26.
Av. shortest path len.	1.9	1.6	1.4	1.3	1.1	1.0
Av. clustering coeff.	0.80	0.82	0.85	0.87	0.93	1.0

Table 6.3: Properties of the graphs generated from the Euro-IX dataset with various scale-down factors (SDF); large SDFs correspond to small CXP penetration, while SDF=1 involves the full multigraph.



(a) Edge multiplicity. (b) Path diversity.

Figure 6.2: CCDFs of inter-IXP edge multiplicity and path diversity.

interconnected by over a hundred distinct ASes, each of which is in a position to offer one or more pathlets between each pair. Between the largest IXPs, which form the most likely targets for an initial deployment, hundreds of pathlets may be available; these pathlets connect IXPs directly over the shared member ISPs.

Figure 6.2b shows the CCDF of *path diversity*, which is the number of *edge-disjoint* paths between each pair of IXPs, computed with the minimum cut. These paths can cross multiple IXPs and may be composed of multiple pathlets used in sequence. Conceptually, the cut provides the minimum number of pathlets which would have to

be removed so that no path *at all* is found between these IXPs. We note however, that a failure inside a single ISP (e.g., related to internal routing) can affect many pathlets offered by this ISP. As Figure 6.2a and Figure 6.2b show, the path diversity is much higher than the direct connectivity i.e., edge multiplicity between pairs of IXPs. Thus even when *all* direct ISP pathlets between an IXP pair fail, multiple indirect paths crossing other ISPs and IXP anchors may be used to replace the lost connectivity and recover availability. Furthermore, the most prolific peers (such as Hurricane Electric and other big ISPs) which can contribute hundreds of inter-IXP links due to concurrent peering, are usually domains with worldwide coverage that are extremely unlikely to be partitioned and disconnect from the multigraph. We will further investigate how edge multiplicity and edge-wise path diversity evolve over time in Section 6.3.6, in order to understand whether our observations are valid across different time points.

6.3.4 Reaching Clients with a Handful of IXPs

To be useful and consequently successful as a system, reaching a large client base is important for a CXP. Therefore, we address the question of how much of the IPv4 address space[8] can be reached from IXPs and their members. Figure 6.3 depicts the IPv4 address coverage versus the number of participating IXPs, assuming a greedy deployment strategy maximizing IPv4 address coverage by attracting the required IXP peering base. We show results both for directly adjacent IXP members as well as those connected over a single intermediate ISP (one AS hop away from the IXP[9]). We observe that we can serve over 1 billion IPv4 addresses via only 5 CXP anchors in well-connected IXPs for directly connected customers, which is 40% of the announced IPv4 addresses in the Internet. This increases to 2.4 billion IPv4 addresses (91% of announced addresses) if we also consider the 1-hop customer cone of the IXP members. With 20 IXPs, more than 1.5 billion IPv4

[8] We examined only IPv4 address coverage due to the universal deployment of IPv4 and the relatively low adoption of IPv6, even at the time when this thesis was written: http://www.worldipv6launch.org/measurements/

[9] In this case, both the intermediate ISP—which is a direct member of the IXP—and its customer ISP—which peers with the member ISP potentially at a private peering point—should offer one pathlet each in order to connect the IPv4 endpoints with the IXP anchor and the rest of the multigraph.

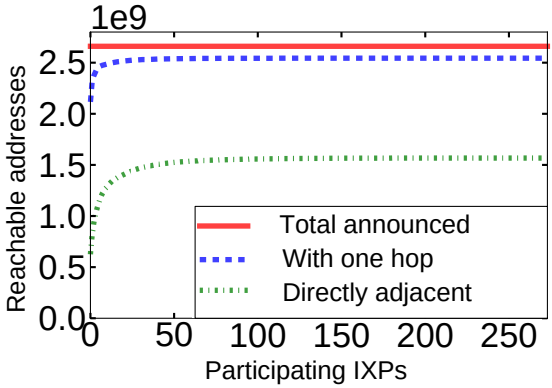


Figure 6.3: Cumulative coverage of IPv4 address space originated by IXP members vs. participating IXPs.

addresses (>50% of announced IP addresses) can be reached directly. This allows an initial deployment of just a few IXPs to serve large parts of the IPv4 address space and enables efficient incremental adoption of inter-domain QoS-enabled services. Further use of private peering points might selectively augment the required coverage, where applicable; in this work we report only on the public peering points that we can see in our available datasets. We believe that the coverage results are very important for future work and for parallel works such as the investigation of the potential of remote peering [54]. Moreover, one could do research on strategies that treat IP addresses with different weights; e.g., instead of maximizing a randomly selected IP address space by adding more and more IXPs, one could maximize coverage for addresses related to popular services (like Youtube, Amazon Web Services, etc.). We followed the random selection approach for reasons of fairness and to avoid any corresponding biases.

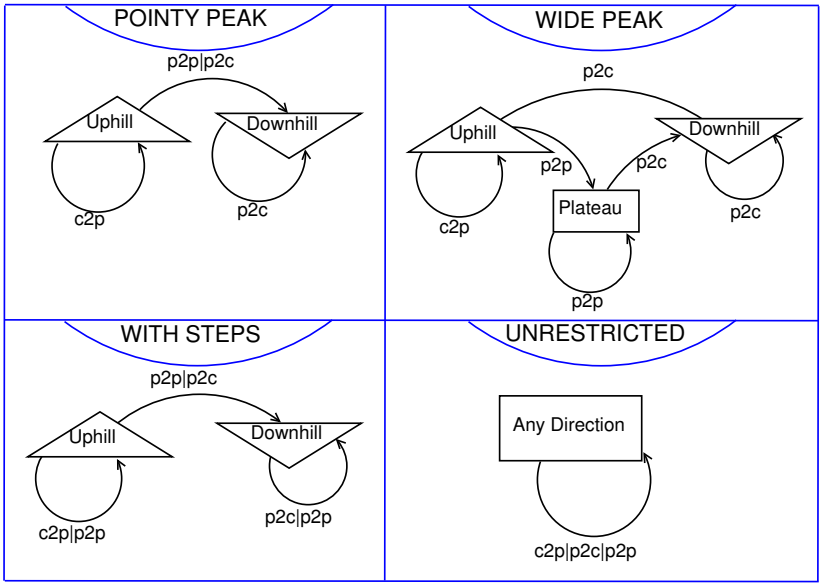


Figure 6.4: The four NFAs corresponding to the four different policy scenarios that we study. The state nodes are labelled using appropriate mnemonic names, while the state transitions are labelled based on the type of inter-AS relationship. All NFA states are also terminating.

6.3.5 Rich Policy-Compliant Path Selection

We next evaluate the increase in path diversity gained when using a CXP-enabled IXP multigraph with relaxed peering policies, as compared to classic valley-free routing on the AS-level topology. We cover a spectrum of possible routing policies. The most constrained policy corresponds to the traditional valley-free model [98] (scenario Pointy Peak[10]); this allows the sequential composition of an uphill path (over customer-to-provider c2p links), then at most one peer-to-peer (p2p) link, and a downhill path (over provider-to-customer p2c links), resembling a mountain with a rather narrow peak. The upper bound on

[10]Note that we choose mnemonic names for the scenarios which roughly describe the type of paths available in each scenario.

path diversity is achieved with the unrestricted policy scenario (scenario UNRESTRICTED[11]). We investigate two additional scenarios by gradually relaxing the valley-free conditions. *(i)* The WIDE PEAK scenario extends valley-free routing by allowing an arbitrary number of p2p hops between the uphill and the downhill path, instead of at most one. This represents a scenario where there is exactly one CXP-mediated path traversed, passing over multiple IXPs and using the associated inter-IXP links. This corresponds to a mountain with a wide plateau instead of a narrow peak. *(ii)* The WITH STEPS scenario allows an unlimited number of p2p links anywhere in the uphill path, and also in the downhill path. Any number of CXP-mediated paths can be traversed either while climbing uphill or descending downhill; this results in a step-wise setup, i.e., a mountain with potentially wide plateaus at different altitudes. We remind the reader that in pathlet markets [78, 241], such as the one that CXPs may form, the aforementioned scenarios are sensible since everyone has an incentive to offer pathlets; what differs among pathlet providers is the technical capability of the ISP and the alignment with its current business logic. For the sake of completeness, we visualize the four different inter-domain policies that we model here with the use of NFAs in Figure 6.4; as explained in Chapter 5, these NFAs can be combined with our view of the AS-level graph for min-cut calculations.

To address the known deficiency in detecting p2p links using the current methodology to find AS-level links [7], and to investigate the effect of more extensive peering on the Internet topology, we augment the AS relationship graph with p2p links derived from IXP membership datasets [79]. A given percentage of the derived links (cf. Table 6.4) is added to the graph, chosen uniformly at random; gradual addition is depicted with increasing percentages[12]. We use the approach that we co-authored with Klöti et al. [144] to estimate the AS-level policy-compliant path diversity[13]. We note that this approach yields upper and lower bounds on the path diversity, with exact values obtained depending on the form of the policy. In our case, the values of the

[11] The UNRESTRICTED scenario doesn't represent our envisioned model, but is rather an upper bound reference for the achievable AS-level path diversity. In particular, it allows paths to transit customer ASes, which in practice happens extremely rarely [102].

[12] Percentages $> 50\%$ were not investigated due to the memory limitations of the state-of-the-art min-cut implementations of the NetworkX Python library.

[13] All the needed details of this approach have been provided in Chapter 5.

		Perc. of added p2p links					
		0 %		25 %		50 %	
Scenario	Description	μ	M	μ	M	μ	M
Pointy Peak	Valley-free	2.9	2	3.2	2	3.3	2
Wide Peak	+ multiple peering links	10.	2	43.	3	70.	3
With Steps	+ unconstrained peering	19.	3	68.	4	104.	4
Unrestricted	No restrictions	42.	5	108.	7	143.	7

Table 6.4: AS-level policy models and their mean (μ) and median (M) path diversity, with different percentages of added p2p links.

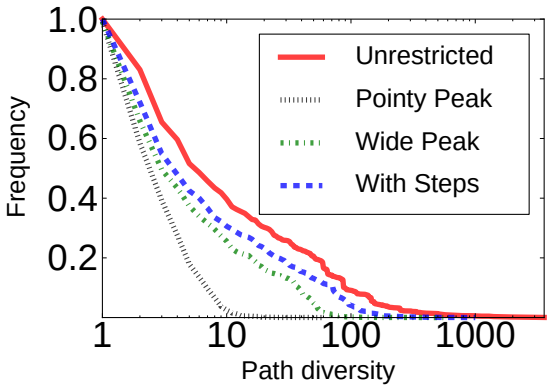


Figure 6.5: CCDF of AS-level path diversity by scenario (with no added p2p links from IXP membership, to isolate the effect of policy).

Pointy Peak, Wide Peak and Unrestricted scenarios are exact, while the values of the With Steps scenario are a 2-approximation of the actual path diversity[14]. We use a sample size of 10K pairs of AS endpoints, selected uniformly at random, with each AS weighted by the number of IPv4 addresses it announces over BGP. This sample was sufficient for our needs.

Table 6.4 shows the mean and median path diversity observed for the various models and amounts of added p2p links, to evaluate the

[14]The reader is referred to Section 5.4.5 for an explanation of this statement.

effect of policy on moderately "rich" substrates. Figure 6.5 shows the distribution of path diversity for the models without added p2p links, in order to vizualize the effect of different policies in isolation from the added links. We observe that transitioning from Pointy Peak to Wide Peak greatly increases the path diversity, even without added p2p links. Wide Peak clearly has an advantage over Pointy Peak even when the latter has many new links added and the former does not. This is true for the mean, but also the median, which is less affected by the highly skewed distribution; for example, for tier-1 and large tier-2 ISPs we see an increase by up to a factor of *29*. The With Steps scenario has more modest gains in median path diversity and lies within a factor of two of Unrestricted, which is the upper bound. After examining the data, we observed that the advantage of Unrestricted and With Steps over Wide Peak stems mainly from a relatively small number of very well connected nodes. We therefore conclude that *(i)* relaxing constraints on peering policy greatly increases path diversity, more so than simply introducing new p2p links, and *(ii)* further relaxations of the model yield relatively modest benefits.

Last but not least, the small world properties of the Internet AS-level topology graph, corroborated via the IXP multigraph abstraction (cf. Table 6.3), and our analysis of shortest path lengths[15] show the following. Since the Internet is densely connected on the AS level, with the number of interconnections growing within a valley-free regime, relaxing the policy constraints does not yield *shorter* paths but simply allows us to use *more* paths. We observed average lengths within 3-4 hops irrespective of policy, in agreement with other reports [158]. Our observations of path lengths subject to different policy models without added p2p links verify our intuition that policy changes do not have any measurable effect on AS path lengths.

6.3.6 Temporal Analysis of PeeringDB Multigraphs

In this section, we use available snapshots from the PeeringDB database, complementary to the Euro-IX snapshot-based analysis, in order to verify that our observations regarding the properties of the projected

[15] We omit the visual presentation here since the relevant information of the figure can be summarized in the few lines of this paragraph.

CXP multigraph are valid *over time*. We use PeeringDB due to: *(i)* the availability of the associated data snapshots, and *(ii)* the partly complementary nature of its information as compared to Euro-IX, which allows us to do a similar analysis and check the agreement of the yielded insights. We note that this analysis is not intended to be exhaustive, but rather an indicative demonstration of the temporal evolution of the peering ecosystem and the associated IXP multigraph, on which CXPs may operate. By knowing the past, we can extrapolate what may happen in the future, as CXPs expand within an IXP-based Internet.

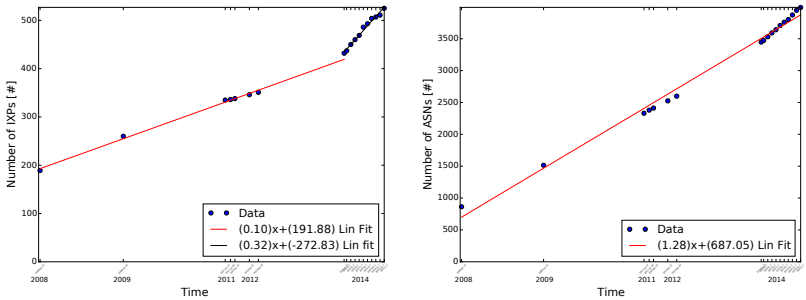
Datasets and Multigraph Construction

For our temporal analysis we use monthly snapshots from crawling the PeeringDB website over the months 3/2014 to 1/2015, effectively covering the monthly evolution of the data during the year 2014. We also process the data extracted from SQL dumps on an almost yearly basis over 2008-2012, for the following dates: 15/2/2008, 14/10/2009, 27/10/2011, 5/12/2011, 7/1/2012, 21/4/2012 and 26/6/2012. We map the CXP multigraph based on the PeeringDB IXP and *(IXP,ASN)* membership information following the same procedure as the one described in Section 6.3.2. In particular, we take into account the largest connected component of the undirected multigraph formed with IXPs as nodes and AS-based pathlets as edges between these IXPs, as long as these ASes connect concurrently to the corresponding IXP pair.

Indicative Results and Insights

First, we note that the x-axes in all the following figures (Figure 6.6, 6.7, 6.8, 6.9) are linearly scaled based on the number of *days* between the snapshots; for example, a linear fit with a +0.10 slope corresponds to an increase of ~ 36 units per year for the associated y-value.

We start with the evolution of the total number of the IXPs and ASNs which participate in the peering ecosystem, over time, in Figure 6.6. We see that the number of IXPs has been linearly increasing at a rate of ∼36 IXPs/year between the start of 2008 and the end of 2013, while we observe an acceleration to a ∼115 IXPs/year rate of increase between the start of 2014 and the end of 2014 (cf. Figure 6.6a). The latter is a result of the recent influx of small IXPs mostly located in South America, Africa and Australia; we will later revisit these IXPs



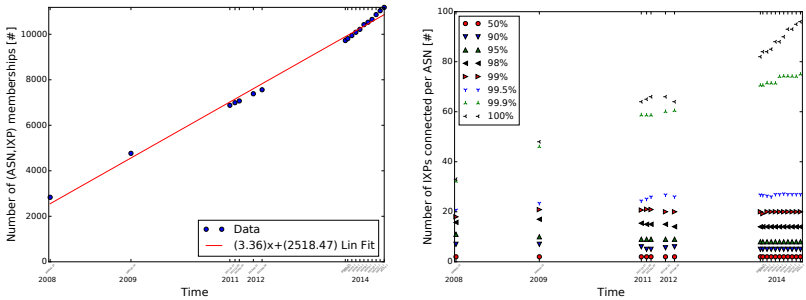
(a) Total number of IXPs in the dataset vs time (with linear fit).

(b) Total number of ASNs in the dataset vs time (with linear fit).

Figure 6.6: Number of IXPs and ASNs in PeeringDB over time.

to determine their impact on the CXP multigraph. On the other hand, the number of ASNs that are reported in PeeringDB seems to follow a steady linear increase at a rate of ∼460 ASNs/year (cf. Figure 6.6b). Some of these ASes, as we show later, may be capable of acting as inter-IXP pathlet providers, thus contributing to the density of the multigraph. In general, we observe that IXPs and their connected AS peers are rising monotonically in sheer numbers over the years; IXPs have increased from less than 200 in the beginning of 2008 to more than 500 in the end of 2014 (cf. Figure 6.6a), while the participating ASes have increased from ∼900 to ∼4000 between the respective time points (cf. Figure 6.6b).

Next, we observe a linear increase in the total number of *(ASN,IXP)* memberships over the years, at a steady rate of ∼1.2k memberships/year (cf. Figure 6.7a). The size of this membership set started at ∼2.6k in 2008 and reached about ∼10.5k at the end of 2014. This is a good indicator that the peering mesh becomes more and more dense, but is not enough to justify the potential of a CXP in its own right. Therefore, we further look into the distribution of the number of IXPs per ASN, since CXPs can take advantage of the fact that a single AS usually peers concurrently at multiple IXP locations. This AS can then serve as a pathlet provider between each IXP pair. As we see in Figure 6.7b, the 50% mass of the ASes are connected to a single IXP; this fact seems to be invariant over the years, while these ASes are obviously



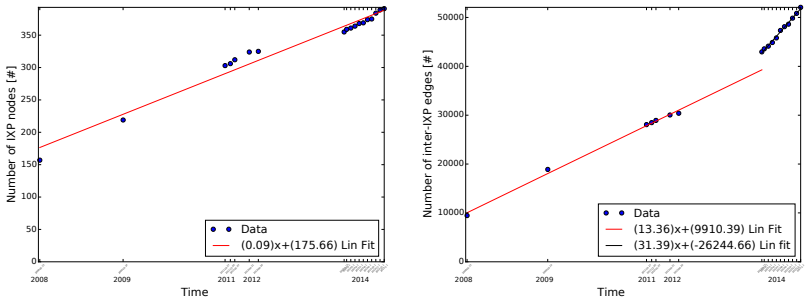
(a) Total number of (ASN,IXP) memberships in the dataset vs time (with linear fit).

(b) Total number of connected IXPs per ASN in the dataset vs time (with indicative percentiles).

Figure 6.7: Number of (IXP,ASN) memberships and percentiles of IXPs per ASN connectivity in PeeringDB over time.

not useful as transit pathlet providers in a CXP context. They can be used though as access providers for connecting client IP endpoints to the rest of the multigraph via their first hop IXP anchor. The behavior of the next 49.5% of the ASNs seems to be also quite stable over the years, with these ASes being connected though to at least two IXPs; they are thus very useful for the formation of a stable IXP multigraph. In fact, the upper 0.5% of this percentile connect to over 20 IXPs. In general, if an AS connects to n distinct IXPs, then it can potentially contribute at least $n(n-1)/2$ bidirectional pathlets of type *(IXP_A,ASN,IXP_B)* to the CXP multigraph. Last but not least, we note the interesting behavior of the upper 0.5% of the ASNs, which connect to a monotonically increasing (large) number of IXPs, ranging from over 30 in 2008, to over 60 in 2014. We observed that these are generally extremely prolific peers which apply extensive peering as their business model, such as Akamai Technologies—a well-known global Content Distribution Network (CDN)—and Hurricane Electric—a global ISP[16]. As a take-away message, we note that the evolution of the *(IXP,ASN)* membership points to a stable multigraph base, with a

[16]In fact, Hurricane Electric claims the following in their website (`http://he.net/about_us.html`): "Within its global network, Hurricane Electric is connected to 90 major exchange points and exchanges traffic directly with more than 3,600 different networks." This statement is also verified via the observed data.



(a) Total number of IXP nodes in the CXP multigraph vs time (with linear fit).

(b) Total number of inter-IXP edges in the CXP multigraph vs time (with linear fit).

Figure 6.8: Number of nodes and edges in the PeeringDB-based CXP multigraph over time.

∼1% of ASes being responsible for its potential expansion. We will see later that this expansion leads to important increases in path diversity.

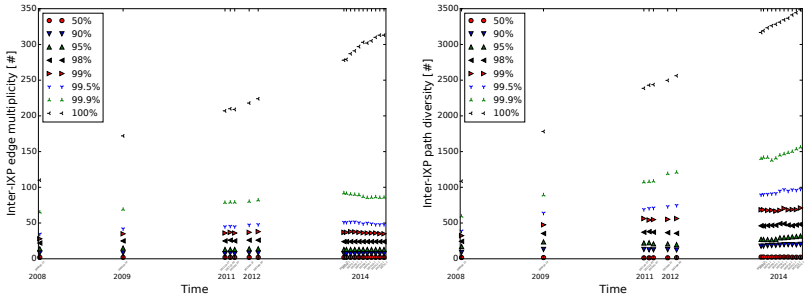
We next form the actual corresponding IXP multigraph instances over time, and examine their sizes in terms of nodes (IXPs) and edges (inter-IXP pathlets passing over AS domains) in Figure 6.8. In Figure 6.8a, we observe that the number of IXP nodes in the multigraph is increasing at a rate of ∼32 IXPs/year. We note here that this behavior is a bit different than the one that we observed for *all* the IXP nodes in Figure 6.6a. This is because the multigraph is based on the largest connected component of the IXP-based full graph; some of the IXP nodes may be left out in case their member ASes cannot connect them to the rest of the multigraph. We see that until the end of 2013, both the multigraph nodes and all IXPs (cf. Figure 6.6a) follow a similar rate of increase, with comparable absolute numbers. After that, we observe that while the rate of increase of the number of the multigraph nodes remains stable, the number of IXPs in the full dataset increases by an extra ∼80 IXPs per year, thus including additionally ∼80 IXPs that do not participate in the multigraph over 2014. After carefully inspecting the data, we deduced that these IXPs are the hubs of highly localized peering ecosystems, connecting ASes that do not peer at any other global IXP. Examples of such IXPs are the ones in some remote

parts of Africa, Australia, East Asia and South America. We speculate that there is a tendency for small ASes and ISPs in these areas—that pay large amounts to their transit ISPs to reach other local ISPs—to peer directly over such IXPs and offload their traffic for cost reduction. Large ISPs that may peer concurrently at multiple IXPs around the world are usually not members of such IXPs, while the IXP peering base is itself relatively small in terms of absolute size.

Moreover, in Figure 6.8b, we observe that the number of inter-IXP edges in the connected multigraph has been increasing at a rate of ∼4.8k edges per year between the years 2008 and the third quarter of 2013, while afterwards the increase reaches a rate of ∼11.3k edges per year. Note that the number of peering edges that PeeringDB sees in 2014 is in agreement with the number that we observed in Euro-IX for a snapshot taken during April 2014 (see the full multigraph properties at Table 6.3). By correlating the observation of the observed rates with the information of Figure 6.7b, we deduce that the responsible ASes for this increase is the upper 1% of all ASes. Each of these ASes is connected to at least 20 IXPs, thus contributing at least 190 edges in the multigraph. The upper 0.1% contributed at least 600 edges per ASN in 2008, and at least 2.5k edges per ASN in 2014. We also see that the "knee" in the third quarter of 2013 correlates with an associated steeper increase on the number of IXPs per ASN for this upper 0.1% of ASNs. This is probably due to their more aggressive peering at geo-diverse public IXPs in the recent years. In total, the number of multigraph edges has evolved from ∼10k edges in 2008 to over 50k edges marking the start of 2015. Further correlation with Figure 6.7b shows that the multigraph has a "slow" changing component increasing at ∼5k edges per year; the lower 50% of all ASes do not contribute at all to this component, while the upper 50% is responsible for sustaining this rate over the years. The upper 1% of the highly connected ASes is much more dynamic, contributing an extra ∼6k edges per year during 2014.

Finally, we examine the most interesting properties of the CXP multigraph, i.e., the edge multiplicity associated with its multi-edges and the corresponding path diversity in Figure 6.9. In Figure 6.9a we examine the number of edges between *directly connected* IXP pairs[17]. We

[17]For the sake of completeness, we note that this class of directly connected pairs corresponds to about 9% of the total number of candidate IXP pairs. That is, the IXP-level peering matrix between all IXPs is 9% full.



(a) Edge multiplicity distribution percentiles between all the directly connected *(IXP-IXP)* pairs of the CXP multigraph.

(b) Edge-wise path diversity distribution percentiles betweeen all the candidate *(IXP-IXP)* pairs of the CXP multigraph.

Figure 6.9: Edge multiplicity and edge-wise path diversity of the PeeringDB-based CXP multigraph over time.

observe that 50% of the directly connected IXP pairs in the multigraph have an edge multiplicity of 1, which is the typical median value. These pairs are connected via a single carrier ISP, while each IXP of such edges can be connected to many other IXPs via different ISPs, albeit with a low redundancy. As we will show later, this behavior is balanced by the indirect path diversity and high redundancy in terms of indirect paths between the IXP nodes. In particular, as opposed to the low redundancy of these pairs, the rest 49.5% of the directly connected IXP pairs have a multiplicity ranging from 2 to 50. We note that the upper 0.5% reaches levels of more than 50 edges per pair, with the top 0.1% striking an increasing multiplicity of over 100 in 2008, to over 300 in 2014. By manual checking, we discovered that these pairs correspond to the largest global IXPs, such as DE-CIX, AMSIX and LINX, connected over large shared ISP peering bases. These IXPs form the most likely targets for an initial deployment. Our findings corroborate the observations of the Euro-IX-based analysis in Section 6.3.3 (cf. Figure 6.2a). A few pairs of IXPs are interconnected by over a hundred distinct ASes, each of which is in a position to offer one or more pathlets between each pair; this behavior seems to become more intense over time.

In Figure 6.9b, we show the distribution percentiles of the path diversity between *all* candidate IXP pairs. The diversity is calculated as the number of edge-disjoint paths between each pair, computed with the minimum cut. We see that our observations regarding the edge multiplicity of Figure 6.9a are amplified by about one order of magnitude. That is, the connectivity-wise rich IXP pairs compose a dense multigraph core, leading to a substantial 10-fold increase in the overall path diversity as opposed to edge multiplicity. In fact, our findings again corroborate the observations of the Euro-IX-based analysis of Section 6.3.3 (cf. Figure 6.2b); the path diversity is much higher than the direct pathlet-based connectivity. Thus even when all direct ISP pathlets between an IXP pair fails, multiple indirect paths crossing other ISPs and IXP anchors may be used to replace the lost connectivity. The related values seem to increase over time.

In summary, we observe that the conclusions of Section 6.3.3 based on Euro-IX hold also for PeeringDB over time. The most important observation is the linear increase of the size of the peering ecosystem[18], with the edge multiplicity in the corresponding multigraph leading to an order of magnitude larger path diversity over any IXP pair (with 1000s of paths available between the upper 0.1% of the pairs). This is intensified as time progresses, especially in the recent years. A heavy tail of well-connected IXPs and aggressive AS peers is responsible for the dynamic expansion of the multigraph over the years. We note finally that a more extensive analysis of the PeeringDB dataset in the context of CXPs is outside the scope of this work. For further results and insights related to the time dynamics of PeeringDB *in general*, e.g., in terms of peering capacity, we refer the reader to the MSc thesis of Demian Jäger [134], that we co-advised with Dr. Bernhard Ager under the supervision of Prof. Dr. Bernhard Plattner.

[18]This increase seems to be due to the more aggressive peering of big players like Akamai and Hurricane Elelctric, and the introduction of many IXPs in remote parts of the globe for the formation of localized peering setups to offload traffic and reduce costs for the small ISPs.

6.4 Pathlet Stitching Algorithms over Dense Multigraphs

As shown in Section 6.3, the IXP-based multigraph—on which CXPs and pathlet brokers in general may operate—is very dense. In this section, we present algorithms to exploit its rich path diversity in order to maximize the number of concurrently embeddable routes subject to QoS guarantees, like maximal latency or minimal bandwidth. The algorithms may serve as the application logic of logically centralized controllers, operating on the global view of the dense IXP multigraph for stitching end-to-end paths with guaranteed properties.

The algorithmic problem that we have to tackle is complex for several reasons. *(i)* Requests from the large client base (cf. Section 6.3.4), i.e., the IP endpoints attached to the edges of the multigraph, dynamically arrive over time in a non-predictable manner, necessitating the use of online algorithms. *(ii)* While finding a single suitable e2e path is quite easy (i.e., can be performed in polynomial time), the IXP-based graph offers rich path choice (cf. Section 6.3.3, Section 6.3.5) and requires suitable selections of which of the (virtual) edges between two IXPs is used. *(iii)* The online selection of e2e paths should reflect multiple conflicting high-level objectives, namely accepting as many requests as possible, avoiding the use of scarce low-latency, high-bandwidth links, and preventing resource fragmentation. The path embedding problem, seen from these three dimensions, is NP-hard (cf. Section 6.6.3).

We formally introduce the e2e routing problem considered in this work as the QoS Multigraph Routing Problem (QMRP) in Section 6.4.1, together with an optimal offline formulation. Subsequently, we present a general algorithmic framework to solve the QMRP in an online manner. In particular, given the computational complexity of the problem, we employ a *sample-select approach*, where in the first stage, a set of *feasible* paths is *sampled* (i.e., generated), and subsequently one of them is *selected* for the actual embedding (cf. Section 6.4.2). Lastly, the framework is extended to support reconfigurations of pre-generated embeddings in order to accommodate further online requests and achieve increased acceptance ratios in Section 6.4.3.

6.4.1 The QoS Multigraph Routing Problem

We model the IXPs and their pathlet interconnections as a directed multigraph $G = (V_G, E_G)$, where V_G is a set of IXPs (nodes/vertices) and E_G are inter-IXP pathlets (links/edges) offered by ISPs. The ISPs annotate their pathlets $e \in E_G$ with their available bandwidth $\mathsf{bw}_e \in \mathbb{R}_{\geq 0}$ and their latency $\mathsf{lat}_e \in \mathbb{R}_{\geq 0}$. This annotation list can be further augmented with additional properties; bandwidth and latency are specific to the QoS problem that we consider here.

On this substrate, we want to embed a set of e2e routing requests, henceforth denoted by $\mathcal{R}$. A request $R \in \mathcal{R}$ asks for the establishment of an e2e connection between IP addresses s_R and t_R with minimal bandwidth bw_R and maximal latency lat_R. Note that these start and end points are not included in the pathlet network G. However, each IP address is, by its access ISP affiliation, implicitly connected to one or multiple IXPs (cf. Figure 6.1). Therefore, each request is actually associated with a set of start IXPs (where the source is attached) and a set of end IXPs (where the destination is attached). While we take these multiple start and end IXPs into account in the implementation of the presented algorithms, we assume simple IXP start and end points for the sake of visual representation and simplicity.

We next study how CXP operators can accept (and embed) as many requests as possible. The latter is a natural objective for any revenue-driven provider aiming at the satisfaction and maximization of its client base. Embedding a request $R \in \mathcal{R}$ here refers to finding a suitable path P_R, such that the latency of P_R is less than lat_R and that the path P_R can carry more than the minimal bandwidth bw_R. Importantly, as inter-IXP pathlets can be used by multiple requests, the maximal available bandwidth (i.e., capacity) of pathlets must never be exceeded.

The offline version of the QoS Multigraph Routing Problem (QMRP), i.e., when $\mathcal{R}$ is given ahead of time, can be formulated as an Integer Program (IP), cf. *Integer Program 1* (OptFlow): the binary variable x_R decides whether request $R \in \mathcal{R}$ is embedded and the binary variable P_R^e indicates whether edge $e \in E_G$ is used by request $R \in \mathcal{R}$. The correctness of the formulation stems from the following observations: *(i)* Constraints OF-1[19] and OF-2 induce a unit flow from s_R towards t_R

[19]The equivalent equation of constraint OF-1, as seen from the perspective of the destination t_R, is omitted for brevity in the main formulation since it is a natural

Integer Program 1: Optimal Flow Formulation (OptFlow)

$$\max \sum_{R \in \mathcal{R}} x_R \tag{OBJ}$$

$$x_R = \sum_{e \in \delta^+(\mathsf{s}_R)} P_R^e - \sum_{e \in \delta^-(\mathsf{s}_R)} P_R^e \qquad \forall R \in \mathcal{R} \tag{OF-1}$$

$$0 = \sum_{e \in \delta^+(v)} P_R^e - \sum_{e \in \delta^-(v)} P_R^e \qquad \forall R \in \mathcal{R},\ v \in V_G \setminus \{\mathsf{s}_R, \mathsf{t}_R\} \tag{OF-2}$$

$$\mathsf{bw}_e \geq \sum_{R \in \mathcal{R}} \mathsf{bw}_R \cdot P_R^e \qquad \forall\ e \in E_G \tag{OF-3}$$

$$\mathsf{lat}_R \geq \sum_{e \in E_G} \mathsf{lat}_e \cdot P_R^e \qquad \forall\ R \in \mathcal{R} \tag{OF-4}$$

$$x_R \in \{0,1\} \qquad \forall\ R \in \mathcal{R} \tag{OF-5}$$

$$P_R^e \in \{0,1\} \qquad \forall\ R \in \mathcal{R}, e \in E_G \tag{OF-6}$$

if request $R \in \mathcal{R}$ is embedded (cf. the work of Ahuja et al. [10]). $\delta^+(v)$ and $\delta^-(v)$ here denote the set of outgoing and incoming edges of $v \in V_G$ respectively. *(ii)* By Constraint OF-4 the path described by variables P_R must obey the maximal latency lat_R. *(iii)* By Constraint OF-3 the available bandwidth (i.e., capacity) bw_e of any pathlet is not exceeded.

We note that while the offline problem is interesting for optimizing existing allocations of requests in the background in order to further increase acceptance ratios (cf. Section 6.4.3), we are in general more interested in the online variant. In the online context, each request R is known to the algorithm only at its arrival time, and the algorithm needs to compute an embedding on demand (for the entire duration of the request) at that point in time.

6.4.2 Online Problem: Sample-Select Strategy

In order to tackle the online variant of the QMRP we propose a sample-select approach. In the first stage a set of feasible paths is sampled from all feasible paths. In the second stage one of the feasible paths is selected for the embedding. We employ this approach as computing

result of the application of constraints OF-1 and OF-2. For completeness, this (symmetric to OF-1) constraint would be the following:
$x_R = \sum_{e \in \delta^-(\mathsf{t}_R)} P_R^e - \sum_{e \in \delta^+(\mathsf{t}_R)} P_R^e, \quad \forall R \in \mathcal{R}$

the *optimal* path under multiple objectives and constraints is generally NP-hard [99], while the algorithm might need to handle workloads of tens or hundreds of requests per second; sampling paths on the dense multigraph intuitively seems a more efficient strategy. In Section 6.5, we will evaluate the performance of this mechanism, substantiating our intuition. In fact, we will show that the strategy is well-tailored for the CXP Routing-as-a-Service use case.

While investigating multiple path sampling strategies, we consider only a single selection strategy which primarily aims at minimizing the utilization of the network (in order to provide room for many requests), while secondarily penalizing the use of high-bandwidth and low-latency links (since links with rich resources are more scarce). We note that determining the best selection strategy is interesting in its own right, but lies outside the scope of this work. The strategy used can be summarized as follows: *(i)* Strictly prefer paths with a smaller hop count. *(ii)* Among paths with the same hop count, choose the one with the minimal cumulative inverse utility, computed edge-wise $\forall e \in P_R$:

$$InvU(e) = \frac{\mathsf{bw}_e}{\min\limits_{e' \in E_G(u,v)} \mathsf{bw}_{e'}} \cdot \frac{\max\limits_{e' \in E_G(u,v)} \mathsf{lat}_{e'}}{\mathsf{lat}_e} / |E_G(u,v)| \ ,$$

where $E_G(u,v)$ denotes the set of edges between nodes u, v[20].

Algorithm 1 summarizes the sample-select mechanism that we employ for the online embedding of e2e path requests. If the embedding is feasible we acquire a path for the request and accept it, else no path can be found and we reject the request.

In the following we focus on the path sampling strategy, and present three different algorithmic variants. The goal of the sampling algorithm is to efficiently compile a set of diverse and efficient paths, giving us the flexibility of choice. In particular, we exploit the fact that computing feasible solutions is *not* NP-hard:

Theorem 1. *A feasible path for a given request R can be computed in polynomial time.*

Proof. The proof is constructive: We first prune all edges $e \in E_G$ whose bandwidth is not sufficient to support the minimal bandwidth

[20] We remind the reader that $E_G(u,v)$ is generally a multi-edge set, from which we need to select a simple edge for the actual traffic forwarding along an e2e path.

Algorithm 1: Outline of Online Sample-Selection Algorithm

Input : Network $G = (V_G, E_G, \mathsf{bw}_e, \mathsf{lat}_e)$,
Request $R = (\mathsf{s}_R, \mathsf{t}_R, \mathsf{bw}_R, \mathsf{lat}_R)$
Output : Path P_R to connect s_R to t_R or `null`

1 **sample** set of *feasible* paths $\mathcal{P}_R$
2 **if** $\mathcal{P}_R \neq \emptyset$ **then**
3 | **select** *best path* $P_R \in \mathcal{P}_R$ and **embed** R accordingly
4 **else**
5 | **return** `null`

requirement bw_R. Projecting the resulting multigraph onto a simple graph by replacing each set of links between any two nodes with the minimal latency link of the set, the simple graph G' is obtained. We can now perform any polynomial shortest-path algorithm to obtain the path $P'_R \in G'$, if such a path exists. If $\sum_{e \in P'_R} \mathsf{lat}_e \leq \mathsf{lat}_R$, a feasible path was constructed; otherwise no such path can exist. Assume that the above mentioned process would not find a feasible path even though such a path $P \in G$ exists. By replacing each link of P with the minimal latency link of the corresponding multi-link set, a feasible path in G' is constructed, proving the theorem. ■

Theorem 1 is an important building block for all our path sampling algorithms, as it allows us to: *(i)* abort the generation of paths early using a single shortest path computation, and *(ii)* devise path sampling algorithms that will *always* return feasible paths, if they exist (see also the work of Korkmaz et al. [147]). In the following, three concrete path sampling algorithms are presented. The *Perturbed Dijkstra* (PD) algorithm is essentially a k-shortest paths variant [76], strictly minimizing

Algorithm	Complexity
Perturbed Dijkstra	$O(k \cdot (\lvert E_G \rvert + \lvert V_G \rvert \log \lvert V_G \rvert))$
Guided Walk	$O(k \cdot \lvert E_G \rvert + \lvert V_G \rvert \log \lvert V_G \rvert)$
Guided Dijkstra	$O(k \cdot (\lvert E_G \rvert + \lvert V_G \rvert \log \lvert V_G \rvert))$

Table 6.5: Algorithmic complexity for sampling k paths from the multigraph $G = (V_G, E_G)$.

latency. The *Guided Dijkstra* (GD) algorithm broadens the search space as edge selection is latency-independent, and the *Guided random Walk* (GW) algorithm aims at finding arbitrary feasible paths. Table 6.5 summarizes the run-time complexity of these three algorithms for generating k feasible paths, based on the cardinality of the node and link sets V_G, E_G, respectively.

Perturbed Dijkstra Scheme

The *Perturbed Dijkstra* (PD) sampling algorithm aims at finding the k latency-wise shortest paths according to some graph perturbation criterion (cf. Algorithm 2). After having pruned links with insufficient bandwidth (see Line 1), the multigraph is projected onto a simple graph as described in the proof of Theorem 1. The Dijkstra algorithm is then used to either find a feasible $\mathtt{s}_R - \mathtt{t}_R$ path of latency less than $\mathtt{lat}_R$ or return `null` if none exists. If a path was found, the edge set for the next iteration is perturbed based on the previously found paths. Note that Algorithm 2 does not return a path if and only if there does not exist a single feasible path (cf. Theorem 1). Different perturbation criteria might be employed; we opted in our evaluation for *edge-disjointness* to guarantee the generation of different edge-disjoint paths for achieving redundancy and resiliency to ISP pathlet failures. Node-disjointness could also be employed, but in our case it is of little practical value: our framework capitalizes on the observation that we need a few, reliable CXP anchors deployed on well-connected IXPs, while pathlets sampled from a large pool may fail or be withdrawn in a dynamic fashion. The run-time of Algorithm 2 is dominated by the stages of edge collapsing and executing Dijkstra k times (cf. Table 6.5). This deterministic algorithm is the baseline for our online evaluation (cf. Section 6.5).

Guided Random Walk

The previous approach has two drawbacks: *(i)* it may fail to produce k-many paths as the perturbation might inhibit the discovery of additional paths, and *(ii)* found paths might be biased towards using scarce low-latency links due to the nature of the dijkstra-based path exploration. Using the insights by Korkmaz et al. [147], we now present a *random walk* scheme to explore path diversity: randomization is limited in the sense that the random walk is "guided" to always find a path. This is

Algorithm 2: Perturbed Dijkstra Path Sampling

Input : Network $G = (V_G, E_G, \mathsf{bw}_e, \mathsf{lat}_e)$,
Request $R = (\mathsf{s}_R, \mathsf{t}_R, \mathsf{bw}_R, \mathsf{lat}_R)$,
$k =$ the maximal number of paths to generate

Output : Set of feasible paths $\mathcal{P}_R$

1 $E_G^1 \leftarrow \{e \in E_G | \mathsf{bw}_e(e) \geq \mathsf{bw}_R \wedge \mathsf{lat}_R(e) \leq \mathsf{lat}_R\}$
2 **set** $\mathcal{P}_R \triangleq \emptyset$
3 **for** $i \in \{1, \ldots, k\}$ **do**
4 $\quad E_G^{C,i} \leftarrow \texttt{collapseToMinimalLatency}(E_G^i, \mathsf{lat}_e)$
5 $\quad P_R^i \leftarrow \texttt{Dijkstra}(V_G, E_G^{C,i}, \mathsf{lat}_e, \mathsf{lat}_R, s_r, t_r)$
6 $\quad$ **if** $P_R^i \neq \texttt{null}$ **then**
7 $\quad\quad \mathcal{P}_R \leftarrow \mathcal{P}_R \cup \{P_R^i\}$
8 $\quad\quad E_G^{i+1} \leftarrow \texttt{perturbMultigraph}\ (V_G,\ E_G^i,\ \mathcal{P}_R)$
9 $\quad$ **else**
10 $\quad\quad$ **break**
11 **return** $\mathcal{P}_R$

Algorithm 3: Guided Random Walk Path Sampling

Input : Network (V_G, E_G^1) as in Algorithm 2,
Request $R = (\mathsf{s}_R, \mathsf{t}_R, \mathsf{bw}_R, \mathsf{lat}_R)$,
Minimal distances $d_t : V_G \to \mathbb{R}_{\geq 0}$ towards t_R

Output : Feasible path P

1 **set** $P = \langle\rangle$, $lat_P = 0$ **and** $u = \mathsf{s}_R$
2 **while** $u \neq \mathsf{t}_R$ **do**
3 $\quad$ **set** $N = \emptyset$
4 $\quad$ **for** $v \in \delta_G^+(u)$ **with** $d_t(v) < d_t(u)$ **do**
5 $\quad\quad E_v \leftarrow \{e = (u, v) \in E_G^1 | lat_P + \mathsf{lat}_e + d_t(v) \leq \mathsf{lat}_R\}$
6 $\quad\quad$ **if** $E_V \neq \emptyset$ **then**
7 $\quad\quad\quad N \leftarrow N \cup \{v\}$
8 $\quad$ **choose** $v \in N$ **uniformly at random**
9 $\quad$ **choose** $e \in E_v$ **uniformly at random**
10 $\quad$ **extend** P by e **and set** $lat_P \leftarrow lat_P + \mathsf{lat}_e$
11 **return** P

achieved by guiding the path construction process via initially computed minimal distances $d_t : V_G \to \mathbb{R}_{\geq 0}$ towards the destination node t_R. Note that these distances can be computed by performing a single Dijkstra computation using the collapsed edge set $E_G^{C,1}$ of Algorithm 2.

Algorithm 3, i.e., the *Guided random Walk* (GW), outlines the procedure to generate/sample a *single* path. This procedure can be easily extended to a full path sampling algorithm by initially checking whether a feasible path exists, i.e., by testing whether $d_t(\mathsf{s}_R) \leq \mathsf{lat}_R$ holds. If this is the case, multiple paths can be sampled by invoking the procedure multiple times and collecting its results. The key to always sampling a feasible path is to enforce the possibility to extend path P: in Line 5 of Algorithm 3 only outgoing edges towards a neighbor v are considered, such that the current latency of the path lat_P plus the edge latency lat_e and the minimal distance $d_t(v)$ is less than lat_R. Note that we restrict the exploration process in Line 4 to nodes that are *closer* to the destination than the current node. We include this additional constraint to guarantee *loop-freedom*, which ensures a maximal path length of $|V_G|$. The overall run-time is dominated by the single Dijkstra iteration to compute shortest distances towards t_R and the graph scanning process during each iteration. Algorithm 3 may be repeated k times in order to sample k feasible paths (if they exist), while different node and edge probability distributions may be employed in Lines 9 and 10 of the algorithm for the randomized selection process[21].

Guided Randomized Dijkstra

The last considered path sampling algorithm (cf. Algorithm 4), i.e., the *Guided Dijkstra* (GD), tries to strike a balance between using arbitrary edges and finding latency-wise short paths, while still being guaranteed to sample a feasible path in each iteration. The main difference with the classic Dijkstra algorithm is the adapted neighbor exploration in Lines 8-13. Based on the same insight as for Algorithm 3, Line 9 only selects edges that can still lead to a feasible path. The set of edges between u and v are projected onto a single edge $e \in E_v$ and the classic Dijkstra relaxation is applied by adapting the queue Q, the minimal (found) distances d_Q and the parent pointers p. Together with the initial Dijkstra computation for obtaining the minimal distances towards t_R, the overall run-time equals running Dijkstra $k+1$ times, where k is the maximal number of feasible paths to generate. Again, different edge

[21] In our evaluation in Section 6.5, the uniform distribution yielded the best results in terms of acceptance ratios and path quality.

Algorithm 4: Guided Randomized Dijkstra Path Sampling

Input : Network (V_G, E_G^1) as in Algorithm 2,
Request $R = (\mathsf{s}_R, \mathsf{t}_R, \mathsf{bw}_R, \mathsf{lat}_R)$,
Minimal distances $d_t : V_G \to \mathbb{R}_{\geq 0}$ towards t_R

Output : Feasible path P

1 **set** $Q \triangleq \{\mathsf{s}_R\}$
2 **set** $d_Q : V_G \to \mathbb{R}$ **with** $d_Q(\mathsf{s}_R) \triangleq 0$ **and** $d_Q(v) \triangleq \infty$ **else**
3 **set** $p : V_G \to V_G \cup \{\perp\}$ **with** $p(v) \triangleq \perp$ for all $v \in V_G$
4 **while** $|Q| > 0$ **do**
5 **choose** $u \in Q$ **with** $d_Q(u)$ minimal
6 **if** $u = \mathsf{t}_R$ **then**
7 **return** `reconstruct` $(\mathsf{s}_R, \mathsf{t}_R, p)$
8 **for** $v \in \delta_G^+(u)$ **do**
9 $E_v \leftarrow \{e = (u,v) \in E_G^1 | d_u + \mathsf{lat}_e + d_t(v) \leq \mathsf{lat}_R\}$
10 **if** $E_v \neq \emptyset$ **then**
11 **choose** $e \in E_v$ **uniformly at random**
12 **if** $d_Q(v) > d_Q(u) + lat_E(e)$ **then**
13 $d_Q(v) \leftarrow d_Q(u) + lat_E(e)$ **and** $p(v) \leftarrow u$
14 **remove** u from Q

probability distributions may be employed in Line 11 of the algorithm for the randomized selection process[22].

[22] In our evaluation in Section 6.5, the uniform distribution yielded the best results in terms of acceptance ratios and path quality.

Algorithm 5: Offline Reconfiguration Scheme

Input : Initially rejected request R^-,
Accepted requests $\mathcal{R}^+$ with path P_R^+ for $R^+ \in \mathcal{R}^+$

1 **sample** set of *feasible* paths $\mathcal{P}_{R^-}$ for R^- in the *empty* graph
2 **if** $\mathcal{P}_R \neq \emptyset$ **then**
3 **compute** conflicting requests
$\mathcal{P}_{\mathsf{confl}} = \{R^+ \in \mathcal{R}^+ | \exists P_{R^-} \in \mathcal{P}_{R^-}, P_{R^+} \cap P_{R^-} \neq \emptyset\}$
4 **try to (re-)embed** $\mathcal{P}_{\mathsf{confl}} \cup \{R^-\}$ with an offline algorithm

Integer Program 2: Heuristic Path Formulation (HeurPaths)

$$\max \sum_{R \in \mathcal{R}} x_R \qquad \text{(OBJ)}$$

$$x_R = \sum_{P_R \in \mathcal{P}_R} y_{P_R} \qquad \forall R \in \mathcal{R} \qquad \text{(HP-1)}$$

$$\mathsf{bw}_e \geq \sum_{R \in \mathcal{R}, e \in P_R} \mathsf{bw}_R \cdot y_R \qquad \forall\, e \in E_G \qquad \text{(HP-2)}$$

$$x_R \in \{0,1\} \qquad \forall\, R \in \mathcal{R} \qquad \text{(HP-3)}$$

$$y_{P_R} \in \{0,1\} \qquad \forall\, R \in \mathcal{R}, P_R \in \mathcal{P}_R \qquad \text{(HP-4)}$$

6.4.3 Adding Offline Reconfiguration Support

The sample-select scheme as presented in Algorithm 1 can be used to find good embeddings of e2e path requests arriving one-by-one over time. In particular, the algorithms try to embed each arriving request if this is possible, otherwise they reject the request. However, greedily embedding one request after the other may not be optimal over time, and sometimes, it may be worthwhile to reconfigure existing paths in order to defragment the current allocation and make space for additional requests. Thus in the following, we propose a hybrid online-offline scheme which performs exactly that: requests which are arriving online over time are embedded using one of the sample-select approaches described above. However, in addition, we run an offline optimization procedure in the background: this reconfigures *sets of paths* in order to improve acceptance ratios and resource utilization further. Such reconfigurations may be the only possibility to accept a request on a heavily loaded network substrate.

To enable such reconfigurations, we extend the sample-selection scheme depicted as Algorithm 1 with the fallback scheme depicted as Algorithm 5. Given a just rejected request R^-, feasible paths are first sampled in the empty network, i.e., without any embedded requests. If feasible paths exist and the request R^- could in general be embedded, all requests that conflict with any of the found paths are selected in Line 3. In Line 4 the algorithm tries to reconfigure conflicting requests in order to embed R^-. Note that the reconfiguration task corresponds to solving the offline QMRP, where *all* given requests *must be embedded*. While generally the Integer Program 1 could be used by requiring $x_R = 1$ for all $R \in \mathcal{R}^+ \cup \{R^-\}$, its run-time is prohibitive. We have therefore developed Integer Program 2, which does not compute paths on its own, but is given the previous set of *feasible* paths $\mathcal{P}_R$ of each request $R \in \mathcal{R}$ as input. Again, by forcing $x_R = 1$ for all $R \in \mathcal{R}^+ \cup \{R^-\}$, we can compute whether there exists a reconfiguration of embedded paths that allows accepting R^-, thus increasing the overall acceptance ratio.

We note that the proposed formulation is an adaptation of the classic multi-dimensional knapsack problem [91] which, despite its NP-hardness, can be solved quite efficiently in practice using branch-and-bound solvers [119] when only dozens of paths are used for each request. In the evaluation (cf. Section 6.5), we use the *HeurPaths* program as follows: first we produce sets of paths for all requests (5 to 20 per request) using the previous path sampling algorithms on the initial empty graph, and then we employ *HeurPaths* to allocate the requests in an offline manner using the path set input. In this way we simulate a full offline reconfiguration of the paths that are sampled and selected initially by the online algorithmic variants.

6.5 Evaluation of Pathlet Stitching over IXP Multigraphs

In this section, we evaluate the performance of the pathlet stitching algorithms that we developed in Section 6.4, applied on the IXP multigraphs of Section 6.3. We begin with our experimental setup in Section 6.5.1, including our evaluation metrics and simulation parameters. We then present the associated results and insights in Section 6.5.2, including the most important take-away messages and observations which were yielded during the evaluation process.

6.5.1 Experimental Setup

What we Evaluate

We evaluate the performance of the online, offline and hybrid algorithms presented in Section 6.4 in terms of Acceptance Ratio (AR), substrate utilization, i.e., the ratio of occupied bandwidth to the total available capacity[23], and computation time per request. High acceptance ratios, low utilization (so that there is room for more incoming requests) and short embedding times are desirable. We use a custom simulator[24] and the inter-IXP multigraphs described in Section 6.3, on top of which CXPs can stitch pathlets; we then observe the performance of this stitching process in different algorithmic contexts. All our algorithmic variants ran on the scaled-down versions of the IXP multigraph based on the Euro-IX snapshot from April 19th, 2014 [79].

Parametric Search Space

The parametric search space of our simulations is composed of the cross-product of the following parameter dimensions: *(i)* pathlet latencies and *(ii)* pathlet bandwidths (capacities), *(iii)* requested latencies and *(iv)* requested bandwidths, *(v)* graph sizes (based on the Scale-Down

[23]This metric takes into account only inter-IXP pathlets since the utilization on the access pathlets is coordinated directly between the client endpoint and the access ISP; the ISP reserves as much bandwidth as requested between the client and the available CXP anchors (access utilization=100%).

[24]The associated Python-based code (simulator, parsers, etc.) is available online as open-source software [152] under the Apache 2.0 licence.

Parameter	Space (online)	Space (offline/hybrid)
Compared algorithms	Perturbed Dijkstra (PD) Guided Walk (GW) Guided Dijkstra (GD)	HeurPaths-PD HeurPaths-GW HeurPaths-GD OptFlow
Scaling-down factor (SDF)	32, 16, 8, 4, 2, 1	32, 16
Request latency	unif(100,150), (150,200), (200,250), (250,300) ms	
Paths per request	5, 10, 20	
Number of requests per run	10,000	

Table 6.6: Parameter space for the simulation of the online and offline/hybrid algorithms.

Factor SDF, cf. Table 6.3), and *(iv)* maximal number of paths generated per request (k-paths). Furthermore, we need to take into account the number of requests per simulation run for gaining sufficient statistical confidence in the yielded results, and the temporal characteristics of the requests for the online variant of the problem (e.g., arrival times, durations). This search space has to be explored for each evaluated algorithmic variant. Due to its large volume, we make some simplifications and constrain our search space as follows in the next subsections. The objective is that the simulations may run within reasonable time frames on multi-core environments (∼several weeks) and can be effectively replicated by other researchers using the available software [152]. Table 6.6 summarizes the used parameters.

Latency

Pathlets connecting IXPs pass over ISP domains and are terminated at the edges of these domains, i.e., their IXP-facing PoPs. To model pathlet latency in a geographically diverse ISP, we utilize the Hurricane Electric (HE) looking glass server [126] and perform measurements between pairs of routers situated at major PoPs around the world. The variance of the measured latencies appears not to depend on the geographical distance d. We therefore model the RTT as a linear function (parameterized by a and b) of the distance d combined with a random variable X to reflect the uncertainty in the model: $rtt(d) = a \cdot d + b + X$. Through linear regression we find: $a = 0.016[\frac{\text{ms}}{\text{km}}]$ and $b = 26[\text{ms}]$. Figure 6.10 shows the measured RTT latency, the linear regression $rtt(d)$ (without the

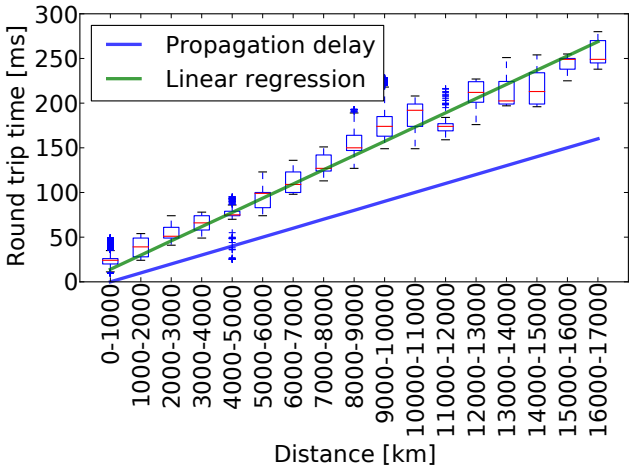


Figure 6.10: Round Trip Times (RTT) between HE PoPs versus inter-PoP distances. We separate the measurements into distance categories and then remove values outside the 10th to 90th percentiles for each category in order to reduce the number of outliers. We show the box plots, the propagation delay baseline and the linear fit that we used.

random variable) and the propagation delay[25]. For X, we observe that within each category, the values are approximately normally distributed. Therefore, by least squares fitting, we model X as a normal distribution $N(\mu, \sigma)$ with $\mu = 0$ and $\sigma = 14$[ms]. We approximate the one-way latency as: $\mathsf{lat}_e = 1/2 \cdot rtt(d)$ assuming symmetric paths.

We note that while a more general intra-ISP one-way pathlet latency model is outside the scope of this work, in practice *guaranteed* pathlet latencies might be much lower than the ones we observed via the HE looking glass server. The reason for this is that the measurement was performed while taking into account the presence of cross-traffic and consequent buffer accumulation on the queues of the HE router interfaces; a guaranteed pathlet though would be probably formed using dedicated QoS queues for the associated traffic and maybe shorter, more efficient

[25] Assuming propagation of light within an optical fiber with reflection coefficient $n = 1.5$, as in the usual case.

paths inside the ISP. Therefore these latencies may be treated as upper bounds for more optimistic QoS-enabled pathlet latencies. We use this latency model for both access and transit pathlets (cf. Figure 6.1), since both types are intra-ISP pathlets that traverse the ISP's domain; the first between the edge endpoint and the CXP anchor, while the latter between the CXP anchors, i.e., the IXPs. Request latencies are selected uniformly at random from four ranges (cf. Table 6.6) to evaluate diverse sets of requirements; ranging from real-time VoIP applications to applications that are less sensitive to latency.

Bandwidth

For simplicity, we consider unitary requests, where each embedded request occupies the full bandwidth of the edge(s) it uses. These can be for example TCP connections that occupy their full allocated bandwidth, while being long-lived (see assumptions on request durations later). This simplification removes offered and requested bandwidth modeling from the search space and allows us to assess our algorithms based only on the topological characteristics of the inter-IXP substrate, rather than the artifacts of modeled bandwidth distributions. In particular, we "fill" the multigraph with allocated bandwidth in order to discover its inherent potential for hosting arbitrarily placed requests. On the one hand, this practice removes the realistic dependency of the results on the interaction between cross-traffic flows on the same edges, but on the other hand it highlights the topological properties (such as path diversity and rich routing choice) of the multigraph which are of primary interest to our analysis. Non-unitary request settings and corresponding simulations and effects are the subject of future work. As a starting point, the effects of cross-traffic and low-level queue-based QoS differentiation are also investigated in other works (cf. Section 6.6.1). For simplicity, non-access IXP-IXP pathlets are also aligned with the unitary request bandwidth setting. In reality, their bandwidth/capacity can be determined by ISP competition and auctioning practices, which are out of the scope of this work but have been partly approached by other researchers in similar envisioned markets [78, 241].

Request Endpoints

We choose candidate IP addresses uniformly at random from the IPv4 address space adjacent to the AS members of the IXPs under examination. The CAIDA RouteViews AS-to-prefix data [51] was used for mapping these ASes to IPv4 prefixes; this information was then used for the randomized generation of candidate IPv4 endpoints of the incoming CXP requests. After we choose a source and destination address for a request, we retrieve their respective coordinates using the MaxMind GeoIP2 database [182] for the geolocation of IP addresses.

The IP coordinates, together with the IXP locations, are used for geographical distance calculations between endpoints and IXPs; these distances can then be used as input to the simulated pathlet latency model that we described beforehand. We assume that IP-IXP pathlets are not constrained by bandwidth, since the access ISP can offer exactly the bandwidth requested in direct collaboration with its client, even without CXP-based mediation. Thus, the "edge" is not the bottleneck in our use case; we mainly investigate the capabilities of a potential CXP-mediated "core" based on inter-IXP connections (i.e., transit pathlets).

Online Requests

The requests arrive in order and are handled one-by-one in an online fashion[26]. Each embedded request persists during the lifetime of the simulation ("infinite" duration), so that the peak load in the online case corresponds to the offline case, allowing for fair comparison at the corresponding graph scales.

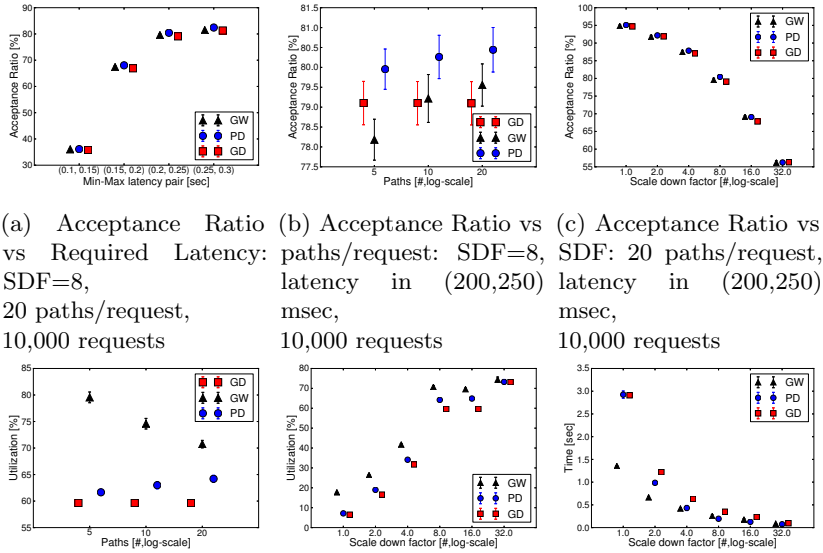
[26] We use Poisson distributions to model the arrival times of the incoming requests. The Poisson process is characterized by a tunable load λ (requests per time unit). This is a well-known model for simulating arrival times for incoming requests in a plethora of scenarios [39]. The start time of each request is determined for simplicity by the run-time of the embedding algorithm. That is, each request requires to be deployed as soon as possible, with the bottleneck being the calculation of the appropriate embedding. Support for fixed start times in the future is also provided in the simulation framework. The duration of the requests follows the log-normal distribution with tunable parameters μ and σ, as in the work of Brown et al. [39] regarding calls within classic telephony networks. To reduce the search space as already mentioned, we simulate "infinite" durations while the start times dynamically depend on the algorithmic performance that we want to evaluate.

6.5.2 Results, Observations and Insights

Figure 6.11 and Figure 6.12 present key observations regarding algorithmic performance, which we further explain and analyze below. Note that the ranges on the y-axes do not have a 0-baseline but are adapted per figure and that all results are based on 10 runs per simulation. We show average values with error-bars of 1 standard deviation. The baseline algorithm for the online case is the *Perturbed Dijkstra*, while *OptFlow* is the offline/hybrid baseline variant.

Which online path sampling algorithm allows for the maximal acceptance ratio, at the lowest utilization? The winner in terms of acceptance ratio is the Perturbed Dijkstra approach with a lead of 1-2% (cf. Figure 6.11a, Figure 6.11b, Figure 6.11c), as opposed to Guided Dijkstra and Guided Walk. In terms of utilization, Guided Dijkstra wins by about 2-5% followed closely by Perturbed Dijkstra, while the Guided Walk is worse within a best-case gap of about 10% from its Dijkstra-based counterparts (cf. Figure 6.11d), across scales (cf. Figure 6.11e). The reason for the prevalence of Perturbed Dijkstra regarding acceptance ratios lies in its k-shortest path discovery; the edge-disjointness perturbation criterion, accompanied by the path selection function (cf. Section 6.4.2), counteracts its tendency to consume precious (latency-wise) paths and leads to good embeddings. Both Dijkstra approaches embed low-latency, low-hop paths that consume small amounts of bandwidth on the substrate network. Especially the Guided Dijkstra performs shortest path routing on random samples of the network, further lowering utilization. The Guided Walk on the other hand, due to the fully randomized path sampling process, embeds feasible but higher-hop paths, with an important penalty on utilization and a small disadvantage in acceptance ratios. Its behavior in these two areas gets better as the number of calculated paths increases (cf. Figure 6.11b, Figure 6.11d), since its progressive, random path sampling process benefits from exploring richer path sets (cf. Section 6.4.2).

How do hybrid variants behave regarding acceptance ratios? HeurPaths with Guided Walk performs the best in terms of acceptance ratios and is very close to the offline optimal values. On the other hand, HeurPaths with Perturbed or Guided Dijkstra leads to lower acceptance ratios as seen in Figure 6.12a, with differences up to 10% for relaxed latency requirements. This is explained due to the optimal latency seeking stages of these algorithms that do not couple



(a) Acceptance Ratio vs Required Latency: SDF=8, 20 paths/request, 10,000 requests

(b) Acceptance Ratio vs paths/request: SDF=8, latency in (200,250) msec, 10,000 requests

(c) Acceptance Ratio vs SDF: 20 paths/request, latency in (200,250) msec, 10,000 requests

(d) Utilization vs paths/request: SDF=8, latency in (200,250) msec, 10,000 requests

(e) Utilization vs SDF: 20 paths/request, latency in (200,250) msec, 10,000 requests

(f) Time per request vs SDF: 20 paths/request, latency in (200,250) msec, 10,000 requests

Figure 6.11: Indicative results for the moderate-scale online simulations. The ranges on the y-axes do not have a 0-baseline but are adapted per figure and all results are based on 10 runs per simulation. We show average values with error-bars of 1 standard deviation (where not visible, $s.d. << avg$).

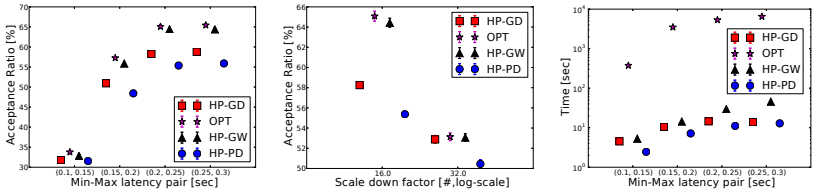
well with the heuristic hybrid allocation. Thus they fail to exploit the richness of the substrate, being biased towards the same low-latency edges. This leads HeurPaths to saturation and limits maneuverability in path allocation. The advantage of Guided Walk is preserved across scales (cf. Figure 6.12b) and latencies (cf. Figure 6.12a).

How do offline, hybrid and online algorithms compare with each other in terms of acceptance ratios and utilization? Our experiments on the 32-SDF and 16-SDF graphs show that the online algorithms perform as good as the optimal offline and hybrid in terms of

acceptance ratios, but have 20-30% lower utilization. The main reason for this is the path selection criterion for the online simulation (cf. Section 6.4.2), which prefers low-hop paths: the online variants hit the optimal value through low utilization, while the offline variants optimize based on sophisticated but computationally expensive allocation of requests, ignoring utilization. Note that with SDFs of 32 and 16 due to the small number of IXPs and the nature of the request model, many of the requests can be served directly using their access ISPs and a single IXP, without occupying bandwidth on the inter-IXP graph. Larger graph sizes were not feasible for OptFlow due to run-time scaling issues as we explain in the following.

How do graph sizes affect run-times? Increasing the graph size (i.e., lowering the SDF) leads to longer run-times as expected, with the online Perturbed and Guided Dijkstras scaling worse than the Guided Walk (cf. Figure 6.11f). This is because the Guided Walk simply finds *feasible* paths quickly, without taking latency *optimality* into consideration and has lower computational complexity (cf. Table 6.5). On the other hand, the optimal offline algorithm operates roughly at 1 to 3 orders of magnitude slower than the hybrid variants at scales of 32-SDF or 16-SDF (cf. Figure 6.12c), and does not scale at all for larger graphs. For the heuristic hybrid algorithm (HeurPaths), the bottleneck is the preemptive path sampling for all requests, while the path embedding stage has negligible time overhead. The use of HeurPaths in collaboration with the Guided Walk yields near-optimal acceptance ratios (cf. Figure 6.12a, Figure 6.12b) at efficient run-times; the latter is evident in Figure 6.12c, which presents the run-time of the Mixed Integer Programming computations versus the requested latencies. HeurPaths needs 10-100s to embed 10,000 paths, but the path computations can be heavily parallelized, or be augmented by existing online paths. For example, the Guided Dijkstra and Walk can be parallelized after their first Dijkstra iteration, reducing run-times on multiple cores. Such optimizations are further described as future engineering steps in Section 7.4.3.

What is the effect of looser latency guarantees? The acceptance ratio (cf. Figure 6.11a, Figure 6.12a) and utilization generally increase monotonically as the latency requirements become looser, i.e., less strict. This behavior comes to a halt when the substrate is heavily utilized. The utilization ceiling is first hit by the Guided Walk, then by



(a) Acceptance Ratio vs Required Latency: SDF=16, 20 paths/request, 10,000 requests

(b) Acceptance Ratio vs SDF: 20 paths/request, latency in (200,250) msec, 10,000 requests

(c) Mixed Integer Progam Time vs Latency: SDF=16, 20 paths/request, 10,000 requests

Figure 6.12: Indicative results for the small-scale offline/hybrid simulations. The ranges on the y-axes do not have a 0-baseline but are adapted per figure and all results are based on 10 runs per simulation. We show average values with error-bars of 1 standard deviation (where not visible, $s.d. << avg$).

the Perturbed Dijkstra and last by the Guided Dijkstra. In the hybrid case, increased latencies and therefore increased search spaces widen the gap between HeurPaths with Guided Walk and the Dijkstra-based variants (cf. Figure 6.12a).

What have we learned from the online-offline cooperation? We have observed that using direct online-offline cooperation as described in Algorithm 5 increases acceptance ratios marginally (∼1%) in overloaded (>70%) substrates. An interesting observation here relates to the request load distribution. The optimal and heuristic offline algorithms have increased utilizations (20-30% more than the online variants), and do not improve too much in terms of acceptance ratios when coupled with online request management. These variants solve the problem purely from the perspective of maximizing the acceptance ratio for the *entire current* set of requests at their disposal, but have no incentive to optimize for utilization at the same time. Thus they prefer to embed as many requests as possible, even at the cost of saturating the substrate. On the other hand, the pure online variants cannot see all the requests concurrently; therefore, they are doing their best to allocate each and every incoming request, or reject it when needed, without sacrificing utilization and jeopardizing future acceptance. A

lesson learned is that, depending on the CXP operator's goals, the heuristic hybrid variant can be reformed to optimize also for utilization and not only acceptance ratios, in order to efficiently defragment the substrate when required.

Summary: which algorithm should we prefer? In our experiments, we observed different behaviors in terms of acceptance ratios in the online and hybrid case. In the online case, Dijkstra-based approaches prevail, while in the hybrid case fully randomized sampling performs better. More precisely, in the online scenario Perturbed Dijkstra is a better choice at small graph scales because of its high acceptance ratios and low utilization; at these scales the run-time of all algorithms is short. We would opt for Guided Walks only at large scales, when fast request allocation is desirable, especially if the incoming load of requests is high (e.g., due to higher penetration of CXPs). In this case, rich path sets (e.g., 20 per request) are important, since they allow the Guided Walk to achieve good acceptance ratios at reasonable utilization levels, close to its Dijkstra-based counterparts. Finally, HeurPaths is a much better candidate for scaling up the hybrid version of the problem as opposed to optimal offline OptFlow, because it achieves similar acceptance ratios—in particular when combined with the Guided Walk—at much shorter run-times.

6.6 Comparison with Related Work

In this section, we first place the CXP approach within the general scope of Internet-wide QoS in Section 6.6.1. We continue with relevant developments in the context of the IXP-based peering ecosystem of the Internet in Section 6.6.2. Finally, we report on related work in the field of QoS routing and path/flow embedding in Section 6.6.3. In particular, we compare such efforts with our algorithmic approach, applied on the special kind of multigraphs that we investigate.

6.6.1 Internet QoS in the Scope of Our Work

Quality-of-Service (QoS) is an evergreen topic that has been discussed for decades [21, 254, 264], together with the challenges associated with its implementation [28, 253, 261]. Such challenges have hindered its Internet-scale adoption in parallel with classic best-effort IP routing

and peering agreements [168]. Our work is not intended to address the whole stack of end-to-end Internet QoS, from low-level implementation (queueing mechanisms, QoS-oriented MPLS-TE) to high-level policies (SLAs, traffic isolation). Instead, we propose CXPs as an IXP-centric approach that can be used to support the deployment of inter-domain QoS in the context of centralized pathlet brokers and resource controllers (cf. Section 6.2.1, Section 6.2.2). This research complements works related to the development of such entities [33, 129, 146, 167, 239].

CXPs could, for example, capitalize on prior work for the implementation [36, 65, 70, 125] and monitoring [194] of QoS-enabled pathlets. More details on this can be found in the discussion of Section 6.2.3 and the future steps of Section 7.4.3. The CXP scheme assumes per-ISP QoS essentially "for-free" and focuses on what can be done assuming that ISPs provide guaranteed pathlets with edges anchored to IXPs, irrespective of how they are implemented. This establishes a clean separation of concerns between intra-domain pathlet provisioning and inter-domain stitching that is consistent with the status quo and the 2-level routing hierarchy in the Internet. While till now the inter-domain level was managed solely via BGP, we propose that IXP-based overlay networks could be used in parallel for the provision of QoS-enabled guaranteed services that BGP is not technically capable of supporting.

6.6.2 Internet eXchange Points: New Developments

The research community has only recently begun to understand the importance of IXPs [57] in terms of: *(i)* the flattening of the Internet topology [71, 110], *(ii)* the prevalence of IXP-based peering links in the Internet ecosystem [7, 20], and *(iii)* IXP-related performance improvements, such as the reduction of average Internet delays and path lengths [8]. This understanding, coupled with the potential rise of SDN within IXPs, enabled by Software Defined Internet eXchanges (SDX) [118], could prove to be an avenue for multi-site CXP deployments and the gradual formation of an IXP-based multigraph with the properties presented in Section 6.3. Further innovative practices—such as remote peering [54]—using the control plane of the IXPs, i.e., their route servers [214], could further support this observation.

Moreover, we note the following regarding our proposed policy relaxation scenarios that we presented in Section 6.3.5. Hu et al. [123] investigate how a version of *on-demand* peering policy relaxation can take place at IXPs in order to recover from route failures. The approach that we follow in Section 6.3.5, based on our prior work on policy-compliant path diversity [144] (cf. Chapter 5): *(i)* is more general, and *(ii)* proposes to actively use the additional path diversity induced from different variants of valley-free policies, based on the sequential composition of *inter-IXP* pathlets as proposed in the CXP model (cf. Figure 6.1). In general, we believe that our work encourages research on IXPs and their role in the Internet, by proposing a new model of how the Internet topology may be viewed, measured and used. Further technical details like fast recovery from route failures—using the rich path choice of the multigraph—is part of future work (cf. Section 7.4.3).

6.6.3 QoS Routing and Embeddings

Finding suitable paths[27] between a pair of endpoints is a classic problem in computer science; the problem has been studied intensively in the context of online call control [180], virtual-circuit routing [19, 23] and also specifically QoS provisioning [99]. In the area of QoS routing, exact, approximate and heuristic algorithms have been considered for finding paths subject to (possibly) multiple constraints and objectives. Based on the dense nature of the CXP multigraph and the online fashion in which requests arrive, we have adapted two well-known heuristic algorithms frequently used in the context of QoS: k-shortest paths [76, 99] and the look-ahead scheme employed by Korkmaz et al. [147]. In contrast to stochastic QoS routing algorithms as presented by Orda [197], we assume QoS guarantees over the ISP pathlets for our basic model. These guarantees may be monitored and verified by the CXP, but we have no incentive for stochastic approaches over virtual paths guaranteed by the ISPs themselves. Moreover, optimal solutions to the QoS routing problem are generally NP-hard to achieve, due to having to consider multiple objectives (minimizing costs, avoiding scarce low-latency links, etc.) or multiple constraints (latency, bandwidth, jitter, etc.) [99].

The heuristic offline variant of our problem (embed as many e2e paths as possible), is a variant of unsplittable flow problems [72]

[27] with specific properties—such as QoS attributes—if required.

and is related to the VPN [116] and virtual testbed mapping [63] problems. For a comprehensive survey, we refer the reader to the work of Fischer et al. [87]. A relevant virtualization architecture for realizing such embeddings is presented by Schaffrath et al. [217]. The hybrid online-offline approach that enables the reconfiguration of existing e2e embeddings for increasing acceptance ratios, was shown to be beneficial also in the domain of virtual network embeddings by Fan and Ammar [80]. Frikha and Lahoud have recently proposed to precompute QoS paths in order to improve performance [93]. In contrast, the paths that have already been computed in our work are reused at a later stage (possibly in different routing contexts), thereby not introducing any additional computational overhead.

Moreover, Ascigil et al. [18] debunk the conventional wisdom that logically centralized computations do not scale in terms of domain-level end-to-end Internet routes; this is an encouraging observation that helps to further support CXP-like approaches. The differences of that work with our research are the following: *(i)* the authors do not consider how their computation scheme can be ported to the current Internet, *(ii)* they use a more simplified version of the AS-level topology based solely on inter-AS relationships, while we invest in a new approach for treating multi-domain topologies based on peering datasets[28], *(iii)* their evaluation is orthogonal to ours focusing more on the (simulated) flow setup delay rather than request acceptance ratios, and *(iv)* they focus more on the system side than the routing and flow embedding algorithms. These works can be thus viewed as complementary to each other.

We further note that the comparison of the algorithms presented in Section 6.4 and evaluated in Section 6.5, with simple-graph approaches of previous work—not tailored to multigraph sampling—was not applicable due to the following reasons. First, simple-graph approaches would have to use orders of magnitude larger substrates for simulation, e.g., combining 2 "half-edges" and one AS node to simulate each pathlet. In such graphs $|V_G| = O(|E_G|)$, while in our case $|V_G| \ll |E_G|$. Therefore, we observed that the simple graph extension induces a considerable bias in the comparison with these approaches. Also, such graphs over-abstract ASes as simple nodes while in our case the multigraph abstraction is more intuitive and maps correctly IXPs (i.e., the peering points of

[28] We also note that off-the-shelf Dijkstra can work quickly on simple AS-level graphs, while we need to deal with the multigraph context.

multiple ASes) as nodes, while AS-traversing pathlets are mapped as edges, as expected. Furthermore, all nodes in the multigraph are of the same type, while the same property applies for the edges; this is a simple homogeneous model on which the algorithms that we developed can easily operate. We also note that the baselines against which we compare in Section 6.5 are optimal strategies for the offline/hybrid case, and basic dijkstra-based approaches for the online case.

6.7 Discussion Points on CXPs and Associated Services

In this section, we place emphasis on the new services that an incrementally formed IXP multigraph—mediated by CXPs—could support, such as telesurgery (cf. Section 6.7.1). We also examine in which ways this support could be provided. We refrain from further discussing system aspects of CXPs, since these are the subject of future work. Instead, we focus on the value of the IXP multigraph as the network substrate on which CXP systems may stitch end-to-end paths, to incrementally provision new inter-domain services (cf. Section 6.7.2).

6.7.1 Telesurgery on CXP-mediated Multigraphs

To get a better understanding of the usage potential that CXPs bring, we investigate the telesurgery [120, 181, 255] example. Telesurgery is undoubtedly a use case with stringent requirements on both availability and latency. Availability is essential for ensuring uninterrupted operations and the patient's safety. Latency is important for making remote surgery feasible with real-time feedback [120]. In addition, the bandwidth requirements can be generally high, e.g., for transmission of video streams [181], but are of secondary importance.

One way to achieve high availability is for the CXP to allocate multiple disjoint paths on the multigraph and send redundant packet copies on each path; one copy is then selected by the receiver and is delivered to the application. A more efficient approach could be using Forward Error Correction (FEC) coding schemes such as Reed-Solomon. For example, a CXP could allocate 12 disjoint paths with 1/10 of

the required capacity each; then use a FEC scheme with 12 channels including 2 times redundancy at 20% bandwidth overhead.

A CXP can check online for path failures; if a path is degraded, it immediately allocates a replacement using the pool of available pathlets, leaving the rest of the operational paths intact. Obviously, less reliable pathlets within ISPs mandate more redundancy to achieve high availability. In contrast, traditional practices for establishing guaranteed paths for sensitive services are limited within a single domain, using either long-term tunnels and leased lines, which are static and usually prohibitively expensive [43], or short-term MPLS VPNs, which incur high administrative and financial overhead [61, 261] as compared to their temporal utility. Quick fail-over in case of emergencies is challenging [229, 261]. As a consequence, higher redundancy is needed *a priory* to achieve acceptable availability.

In a CXP context, the ISP's network resources are virtualized. This example demonstrates how *on-demand* resource provisioning may be used to bring prices down, by bringing up the utilization of the resource and amortizing its costs, analogously to how CPU and storage are better utilized in the context of cloud computing [185]. Client flows can be dynamically assigned to (multiple) pathlets depending on the resources that are available within the "CXP cloud".

Under certain circumstances, CXPs may be able to find lower latency paths than traditional routing. If a path is subject to a triangle inequality [175] violation (the majority of paths in fact are [16]), and there is a well-placed CXP anchor available to route over, the CXP can potentially provide a path with lower latency. This implies the need for a broad CXP deployment footprint. While starting with only a few selected IXPs can serve as an initial step for deploying CXP anchors, it may not be sufficient for optimizing latency [9]. In this case, expansion to private peering points and many geographically dispersed anchors might help with flexible waypoint routing over better-latency paths.

Bandwidth requirements could also be addressed by steering traffic towards IXP-based anchors. The spare capacity of the already installed IXP-ISP peering ports, amounting to Tbps [14], could be potentially used for serving multi-Gbps streams, if required. The CXP approach generally encourages the addition of high capacity peering links to the currently deployed switching infrastructure and enables more efficient bandwidth utilization and dynamic allocation, as described beforehand.

Finally, the application logic of a CXP that stitches paths e2e in order to support sensitive services, could be based on our developed algorithmic framework (cf. Section 6.4). In this case, the input of the algorithms would be the collected state of the pathlet network and the service request requirements, while the output would be the proper path embeddings that satisfy the requests in terms of latency, bandwidth and availability (in terms of multiplicity of edge-disjoint paths).

6.7.2 New Services and Incremental Deployment

There is an abundance of emerging services [266] which could benefit from bandwidth or latency guarantees on an inter-domain network substrate with global footprint. These may range from remote education and real-time feedback from public transport networks, to public safety and emergency channels. Since cloud providers are usually collocated with IXPs [57], coordination with cloud applications and real-time big data analytics and communication could be achieved using the current IXP peering base. This translates to on-site, scalable computation and storage capacity also for the Routing-as-a-Service offering of CXPs.

The deployment of new services could align to the following workflow, assuming a staged use of IXPs as CXP anchors. CXPs gradually approach last-mile access ISPs that serve the intended client base (i.e., IPv4/IPv6 endpoints). These ISPs connect, in turn, to a multitude of IXPs that bridge the clients with the rest of the multigraph. Guaranteed transit paths can be then achieved over regional ISPs within a continental setting, while transcontinental path segments can be covered by ISPs with larger coverage [228], similarly to the status quo. At first, pilot services may start within areas with strong IXP presence and large peering bases (cf. IXPs in Europe and US [79]). Smaller IXPs in different continents could be targeted afterwards via the shared ISP peering base, in order to connect remote peering ecosystems to the current multigraph and augment geographic coverage, where required.

6.8 Summary of the CXP Concept and its Potential

We considered a novel abstraction of the Internet topology; the IXP multigraph. This Internet-wide overlay network can aid the deployment and operation of logically centralized QoS pathlet brokers; the CXPs. CXPs may be based on SDN principles and mechanisms, with the goal of provisioning guaranteed end-to-end services across domains. This is another use case related to the cross-domain routing centralization approach that we presented in Chapter 3, constituting a potential RaaS offering spanning multiple ISPs and peering connections.

Based on our study using extensive peering datasets, we evaluated the potential of an IXP-based substrate—where CXPs may operate—in the following ways. *(i)* In terms of IP address coverage, we showed that even a small deployment (∼5 IXP anchors) could directly cover a high fraction (∼40%) of the Internet address space. *(ii)* In terms of AS-level path diversity, we showed the potential of generalized routing policies applied on a dense IXP multigraph. We observed an increase of at least one order of magnitude in path diversity, as compared to classic inter-domain routing practices, regarding the multiplicity of edge-disjoint paths. More disjoint paths indicate potentially higher availability for demanding applications. We further observed that our multigraph-related conclusions, deduced from a snapshot of Euro-IX, hold also for PeeringDB over a time frame of seven years. One important finding is the linear increase of the size of the peering ecosystem over time, with the edge multiplicity in the corresponding multigraph leading to an order of magnitude larger edge-wise path diversity over any IXP pair. This is intensified especially in the recent years. In particular, a heavy tail of well-connected IXPs and aggressive AS peers is responsible for the dynamic expansion of the multigraph over the years. *(iii)* We exhibited the importance of having suitable path sampling algorithms that take advantage of the new routing flexibilities and the richness of the multigraph, running over a simulated CXP RaaS platform.

We further evaluated the performance and applicability of diverse algorithmic variants—online, offline and hybrid—for different traffic requirements and graph scales; we have shown that centralized routing variants work efficiently on the global IXP multigraph view. Lastly, we placed our analysis within the scope of emerging and resource-

intensive applications, such as telesurgery, investigating the usefulness of a potential deployment based on IXP and ISP support. This is in line with the Internet's status quo and its 2-level routing management hierarchy, with the inter-domain layer being managed in parallel in part by centralized CXPs and in part by distributed BGP interactions.

Finally, open questions for future research are the engineering aspects of building CXP systems and the investigation of the temporal dynamics of the IXP multigraph and the associated ecosystem from a market's perspective (since we explored only some technical aspects of these dynamics, e.g., related to the evolution of path diversity). A potential course of action is emulating a CXP prototype on a small-scale IXP-based deployment, e.g., using multi-site SDXes [117] as traffic switching anchors and controllers; a pilot service to be considered would be IXP-oriented waypoint routing with certain latency or bandwidth properties on the paths. Generally, we see this work as an extension of the key value proposition of SDN across multiple domains, complementary to the SDX and PCE approaches. Furthermore, we believe that it is a strong use case supporting the routing outsourcing approach that we presented in Chapter 3. More details on future steps in the CXP context can be found in Section 7.4.3.

Chapter 7

Conclusions and Future Work

> Everything that has a beginning has an ending. Make your peace with that and all will be well.
>
> —Jack Kornfield, *Buddha's Little Instruction Book*

In this chapter, we first summarize what we have done and what we have learned from this research work (cf. Section 7.1). We continue with a recap of our research contributions, including the publications that were produced in association with this thesis (cf. Section 7.2, Appendix A). We then perform a critical assessment of our research directions, stating and verifying some important observations around the field of our work (cf. Section 7.3). We conclude with an overview of future work associated with this thesis and the outlook of the performed research (cf. Section 7.4).

7.1 Summary and Conclusions

In this thesis, we began with the motivation behind evolvable inter-domain routing and the promise of SDN and routing control plane centralization in this context. In particular, our aim was to deal with challenges related to BGP such as slow convergence, or lack of end-

to-end service guarantees, within a setting governed by inter-domain policies. The basic premise was the difficulty in changing BGP; this challenge mandated the need for incremental deployment, with SDN principles and mechanisms being the vehicles to implement the required changes in an Internet-like setting.

Therefore, we proposed a new architectural approach where a contractor can control (parts of) the inter-domain routing logic of multiple ASes based on their policy requirements and network state, using SDN and Routing-as-a-Service outsourcing mechanisms. As more and more ASes choose the same contractor, AS clusters are gradually formed. These contractors can talk to legacy ASes via BGP, while providing inter-domain optimizations and new services within the clusters in parallel; these optimizations, such as more stable routing and improved convergence, may also benefit non-client ASes. Such contractors may also work independently from BGP by performing overlay stitching of partial paths crossing ISP domains, in order to form end-to-end routes with specific properties. We investigated both the benefits and challenges associated with the basic approach, along with use cases that spawn from its (potential) incremental adoption.

As a first use case, we evaluated the interplay between SDN-based routing centralization and classic BGP routing in terms of convergence[1]. To support that, we developed a hybrid BGP-SDN emulation framework and a multi-AS SDN controller running on top of it. We made the emulation framework and the controller logic publicly available as open-source software [96]. Our fail-over experiments on synthetic hybrid AS-level graphs of diverse scales and types indicate that inter-domain routing centralization can improve convergence times even at small SDN penetration levels. Churn rates are comparable or slightly worse than pure BGP at small scales, with benefits shown at larger scales. We observed that the partial logical centralization of routing control accelerates convergence because of two main factors. *(i)* The state propagation process is accelerated due to the central point where parts of the state are gathered and are then directly communicated outside. This acceleration benefits both client and non-client ASes, but may increase the associated churn in some cases. *(ii)* The controller has a global overview of its client cluster and the inter-domain network that is seen by its clients; this view can be efficiently used for well-

[1]The corresponding quantitative results can be found in Chapter 4.

informed decisions related to path exploration of AS-level routes, based on the cumulative routing feedback from the clients and the external BGP speakers. The ability to study the BGP-SDN interplay may be important also for other works capitalizing on gradual penetration of SDN centralization within legacy distributed networks. Our work serves as a proof of concept along such a direction.

We further considered the fact that the NOS platform of an inter-domain RaaS contractor provides an ideal vantage point on (parts of) the inter-domain graph. Therefore, we can use it for a number of logically centralized calculations pertaining for example to shortest path routing, or the discovery of the available path diversity for multipath inter-domain routing and other network services. In practice though, such calculations are far from simple; the graph that the contractor sees is actually governed by the policies of the individual participants that it serves, e.g., based on valley-free policies and relationships between peers, customers and providers. The effect of routing policy on the properties of such (inter-)network graphs is not well understood. We thus developed an algorithm for transforming the AS-level graph (and in general any arbitrary network graph) in order to calculate policy-compliant min-cuts and estimate the available path diversity and bisection bandwidth within a policy regime, assuming policies expressible as regular expressions and NFAs. Classic off-the-shelf algorithms, such as Ford-Fulkerson for min-cuts or Dijkstra for shortest path routing, can run unmodified over the transformed graph and yield policy-compliant paths and flows. Our approach can be applied in a variety of scenarios, some of which are briefly showcased in this thesis. These include the investigation of Internet topology and alternative policy models in the Internet; we have made the software used to simulate such experiments publicly available as part of the CXP open-source software suite [152].

As a second use case, we considered a novel abstraction of the Internet topology; the IXP multigraph. This Internet-wide substrate can potentially aid the deployment and operation of logically centralized QoS path brokers. Such brokers may be based on SDN principles and mechanisms, with the goal of provisioning guaranteed end-to-end services across domains. We called this family of brokers/controllers deployed over IXPs, Control eXchange Points (CXPs). Based on our study using extensive peering datasets, we evaluated the potential of an IXP-based substrate—where CXPs may operate—in the following ways. *(i)* In

terms of IP address coverage, we showed that even a small deployment (∼5 IXP anchors) could directly cover a high fraction (∼40%) of the Internet address space. *(ii)* In terms of AS-level path diversity, we showed the potential of generalized routing policies applied on a dense IXP multigraph. We observed an increase of at least one order of magnitude in path diversity, as compared to classic inter-domain routing practices, regarding the multiplicity of edge-disjoint paths. More disjoint paths indicate potentially higher availability for demanding applications. We further observed that our multigraph-related conclusions deduced from a snapshot of the Euro-IX dataset, also hold for PeeringDB over a time frame of seven years. One important observation here is the linear increase of the size of the peering ecosystem, with the edge multiplicity in the corresponding multigraph leading to an order of magnitude larger path diversity over any IXP pair, edge-wise. A heavy tail of well-connected IXPs and aggressive AS peers is responsible for the dynamic expansion of the IXP-based multigraph over the years. *(iii)* We exhibited the importance of having suitable path sampling algorithms that take advantage of the new routing flexibilities and the richness of the multigraph, while running on top of a CXP routing platform. Using simulation, we further evaluated the performance and applicability of diverse algorithmic variants—online, offline and hybrid—for different traffic requirements and graph scales; we have shown that centralized routing variants work efficiently on the global IXP multigraph view. Lastly, we placed our analysis within the scope of emerging and resource-intensive applications, such as telesurgery, investigating the usefulness of a potential deployment based on IXP and ISP support which is in line with the Internet's status quo. We made the experimental CXP simulation framework publicly available as open-source software [152].

Generally, we see our work as another step towards extending the value proposition of SDN on the inter-domain level, based on the radical idea of logically centralizing parts and functions of the inter-AS routing control plane. We note that this is one of the most challenging arenas for SDN to penetrate, due to the difficulty of making changes on how core routing works; politics and established practices can put a brake on novel technical approaches. Nevertheless, our current findings encourage further research along the direction of inter-domain SDN, specifically in the context of routing convergence, IXP-based SDN deployment for the provision of new inter-domain services, and centralized computations of multi-domain network metrics based on the global view offered to

a RaaS contractor. Therefore, as a key take-away message, we believe that this work supports our initial hypothesis: *Inter-domain routing centralization, performed in a staged and consistent manner, can help to evolve and improve Internet routing on the AS level.*

7.2 Review of Research Contributions

The contributions of this thesis, as already stated in the summary in detail (cf. Section 7.1), are briefly the following:

- Proposal of a new architectural approach for Internet routing, based on inter-domain logical centralization of parts and functions of the control plane and its management logic (cf. Chapter 3). The core of the approach is a multi-domain RaaS contractor and controller.

- Investigation of a use case associated with improving the convergence behavior of inter-domain routing and BGP, as the proposed approach is gradually applied within a multi-domain setting (cf. Chapter 4). The software for the related fail-over routing experiments on an emulation framework is publicly available [96].

- Investigation of the central computation of policy-compliant min-cuts (and thus path diversity and bisection bandwidth values) on network graphs governed by policies. This includes the case of a RaaS contractor with a global overview of parts of the AS-level Internet, subject to valley-free or even more liberal policies (cf. Chapter 5). The software for conducting such experiments using simulation is publicly available [152].

- Investigation of a use case associated with the provisioning of guaranteed inter-domain end-to-end Internet services. This can be realized with the cooperation of IXPs, acting as deployed programmable traffic switching points with rich connectivity and reach, and ISPs, as providers of partial guaranteed paths over their domains. The stitching of these paths happens under the supervision of CXPs, the Control eXchange Points (cf. Chapter 6). The software of the related experiments, on a simulation framework, is publicly available [152].

In general, we studied a new approach applied to an old, but timely, research topic from multiple perspectives; the objective was to deal with real-world challenges associated with the current state of inter-domain routing. We do not claim that all challenges have been addressed or that all problems have been solved; in fact, our work opens up a number of new research and engineering directions that the community can continue to work on for the coming years (cf. Section 7.4). We also note that where it was applicable, we followed an open-source philosophy for our software implementation. Our intention is that researchers can independently verify our findings and build upon our emulation and simulation frameworks to support their own novel use cases and experiments in the context of SDN and inter-domain routing.

The full list of publications and software-related contributions that were produced during the process of writing this thesis can be found at Appendix A.

7.3 Critical Assessment and Observations

While thinking about the overall project, i.e., the design, implementation and evaluation of the RaaS approach, together with the different use cases, the following observations have to be made before suggesting to other researchers to work in this area. The goal of this section is a critical analysis of some aspects of this thesis to aid future related research, rather than a full report on all pros and cons of this work. In particular we verify—through the prism of our work—some observations that have been made over time by other researchers working in the area of SDN and network management, with the hope that they will also be of use to other prospective researchers.

Know thy Abstractions. The power and promise of SDN relies on defining the proper abstractions; the *southbound* and *northbound* interfaces, as well as the state distribution abstractions are examples of such concepts. In order to demonstrate the potential of our architectural approach in Chapter 4, we abstracted away the domains that are clients of a RaaS contractor as well as the legacy BGP ones as nodes, or "big switches/routers". We have pointed out the strengths and weaknesses of this simplification in Chapter 4, noting that such abstractions are consistent with the information-hiding nature of BGP and the privacy-related needs of candidate RaaS client domains. For example,

these domains might wish to simply "test" the performance of a RaaS contractor, by sharing the least amount of required information for the proper operation of inter-domain routing interactions, while maintaining full intra-domain control independently from the contractor. Such abstractions are tunable and extendable, depending on the information that the RaaS client would like to share, and do not fundamentally limit the use of the SIREN framework in particular or the application of the inter-domain RaaS approach in general.

Furthermore, we capitalized on the abstraction of ISP domains as sets of pathlets connecting IXP-facing points of presence in the CXP context. This abstraction is consistent with the status quo: we advocated a clean separation of concerns between the ISP's governance within its domain, and the CXP's control on the inter-domain level. This abstraction is also tunable in terms of the pathlet annotations, or the offering of visible waypoints *within* the ISP's domain, making the CXP stitching process more fine-granular. In general, the CXP-based pathlet abstraction relaxes the requirement for the outsourcing of the full routing control plane of a domain, and may be more appealing to ISPs for political reasons that we analyze afterwards. Finally, both for the CXPs and the generic RaaS approach we assume a proper *southbound* interface for control and monitoring such as OpenFlow, while services interact with the respective routing controller via a *northbound* interface, such as the ones offered by controllers such as POX [5] or ONOS [33]. The state distribution challenge may be either abstracted away using the event-based processing of POX, or actively dealt with to the programmer's knowledge, by frameworks such as ONIX [146].

Know thy Data. Acquiring data from public sources of information, such as peering information providers for our CXP use case (Euro-IX, PeeringDB, PCH), requires a lot of engineering effort for parsing and processing the data, but is otherwise a straightforward task. It is quite challenging though to define the strengths and weaknesses of the extracted datasets, e.g., in terms of accuracy or completeness. This is the reason why we also invested some of our research efforts on this front, e.g., using BGP information to verify the *(IXP, ASN)* memberships. In general, we warrant caution when working on combined or "super" peering datasets, since the new compiled dataset inherits inaccuracies and artifacts from each original source. These problems may be amplified if inaccurate data is used, e.g., for mapping common

entities between the datasets. This is one of the reasons that we studied PeeringDB in isolation from Euro-IX and PCH. A first effort in the dataset mapping context was performed by Klöti et al. [143] with our collaboration. Interestingly, the situation of incomplete peering datasets might be analogous to the incompleteness of the Internet AS-level topology. In fact, over a decade of research efforts have shown that the AS topology, which can be measured from the available datasets, is largely incomplete, even if multiple datasets are combined. Besides, further research is needed to understand the accuracy of individual attributes in the individual datasets, e.g., prefix or contact information, and how to integrate inconsistent attributes.

Moreover, in the context of the AS-level topology, we are aware that the synthetic graphs we used in the SIREN convergence evaluation case may be an over-simplification of the reality. Nevertheless, the results we have are useful for examining different graph patterns occurring frequently in the (hypothetic) full dataset. We also note that this full dataset is not actually available. Using the AS relationships dataset from CAIDA for topology-based research is usually not recommended since it was originally compiled for other purposes, i.e., research on AS-level relationships and policies. Nevertheless, researchers has extensively used these data in lack of the full knowledge of the accurate topology. Thus, we only made careful use of this information for calculating AS-level min-cuts, while we explicitly pointed out its limitations in the context of our analysis. As a take-away message, future researchers should always carefully investigate the properties of the datasets at their disposal and point out their limitations in the context of their respective use case.

Politics Matter. The inter-domain RaaS approach that we propose has to take not only technical, but also political and financial factors into account (cf. Chapter 3). In fact, besides the engineering challenges of the implementation of a SDN-based routing outsourcing system, researchers working on similar ideas have to face problems related to the nature and structure of the Internet market itself. For example, there may be potential conflicts or difficulties associated with bridging the new outsourcing practices with the current, well-established business logic of an ISP. Moreover, in terms of conflicting policies or SLA violations between two client ISPs, an important matter is that of the behavior of the RaaS contractor. Should it function as a “court” and solve the problem as a “judge”, or serve as a passive observer, emulating the

classic BGP setup where these conflicts unfold as tussles [67] and may result even in depeerings and service disruptions? What happens when the ISPs are not willing to divulge these policies to the contractor? These matters are related to factors beyond engineering and technical research, but are interesting from a political point of view.

Furthermore, we note that making changes on how inter-domain routing works is extremely hard, as previous research has shown (cf. Chapter 2). Even when equipped with novel technical approaches, politics and established practices can slow or hinder the adoption of a new paradigm. In our case, we understand that the changes that we propose may not have an actual impact on the Internet market for years. Our research encourages though incremental deployment, with a relatively high ratio of (potential) impact as opposed to the number of deployment points (e.g., IXP-based CXP anchors). Also, it takes advantage of the ever-rising interest of the community in SDN and its potential applications within the current Internet, particularly across domains. We plan to further build upon this approach from a commercial point of view in order to get a better grasp of the status of the market and its willingness to partly adopt such a proposal.

Keep up with the Hype, but See Beyond. Our work on this thesis was performed within a turmoil of interest and scepticism towards SDN and its promise as an "evolutionary" or "revolutionary" network control and management paradigm. In fact, we observed *(i)* large numbers of SDN-related startup companies rise and either get bought by larger companies or fall into oblivion, *(ii)* classic networking products, such as switches and routers, being marketed under different labels to fit the SDN era, and *(iii)* research on old, even cold, topics being revitalized from the perspective of the new SDN concepts. Discussions on whether SDN scales or not, or how it performs in terms of security, are still raging on. We note that this hype about SDN is quite active and will remain so during the upcoming years. We thus warrant caution for new researchers that aim at working on related fields. Ideally, every researcher should keep up in terms of knowledge and expertise with these rapid changes, distill what really matters amidst such a "chaos", and retain only the basic principles that he/she needs for his/her work, making order out of the influx of information and marketing proposals. In fact, research should see beyond the hype and recognize interesting directions on which the community may work during the coming years.

We believe that this is one of the main outcomes of this work: i.e., distilling the basic principles of a new inter-domain RaaS approach, investigating related use cases and proposing research directions that are associated with the applications of SDN in the very challenging arena of inter-domain routing and ISP operations.

Reality and Emulation/Simulation Trade-offs. Ideally, new routing schemes such as the one that we propose in Chapter 3 should be tested on real (inter-)network equipment, at Internet-wide scale. We note though that achieving this testing scale, especially in the context of AS-level topologies, is an extremely challenging task. Furthermore, deploying a pilot implementation on the real Internet, e.g., interfacing via BGP with actual ISPs, comes with its own set of limitations and associated problems. In fact, it may exceed the temporal horizon of a PhD thesis requiring more focus on efficient engineering and addressing matters of policy and operations, rather than on performing basic research.

Therefore, in this work we used primarily emulation (cf. Chapter 4) and simulation (cf. Chapter 5, Chapter 6) to test our assumptions and hypotheses pertaining to the proposed approach of Internet RaaS. While these methods allow to address challenges of scale better than real-world testing, where the limitations are usually not only technical but also financial (e.g., CAPEX), they of course have their own trade-offs. While a detailed review of the general pros and cons of the used methods is out of the scope of this work, we refer the reader to works such as the one from Roy et al. [216], on the challenges of the emulation of large-scale SDNs. However, we note that the main strength of experiments based on well-defined emulation or simulation environments is their reproducibility by other researchers, assuming the proper documentation. This is why, where applicable, we followed an open-source approach regarding the experimentation code, making our software available to the community. We believe that it is more important to be open, even when some of the simplifications and assumptions might account for a less realistic experiment, than be based on closed proprietary datasets and software that—albeit being more realistic—are not useful for openly verifying and expanding previous research.

7.4 Future Work and Outlook

We see the following research and engineering directions as interesting future work. We remind the reader that before attempting to follow up along these lines, he/she should first go through the critical assessment points that were described in Section 7.3. We believe that the following directions could be the basis for several theses in the field of inter-domain RaaS, using SDN principles and mechanisms. In fact, many of these topics are currently investigated under the umbrella of the associated Netvolution project [189]. Before elaborating on the tasks involved per section, we describe the main insights that are expected to be learned within the respective research area.

7.4.1 RaaS Controller and the SIREN Framework

The main insight that the implementation of the following future work tasks is expected to yield is the practical upscaling of the RaaS architectural approach at Internet scales, both from a technical (i.e., based on engineering) and financial (i.e., based on real outsourcing policies and services) point of view.

Controller Design and Implementation Trade-offs. Future work includes the quantification of the scalability, resiliency and centralization trade-offs for the controller of the multi-AS cluster [167]. This process may be based on the lessons learned from the ONOS [33] and ONIX [146] NOS projects, as well as related projects in the context of SDN controllers as distributed systems [203, 239]. For example, the proper placement of the controller in a multi-domain setting, in order to deal with latency and distribution trade-offs, is an interesting avenue to explore [122], as well as possible fail-over setups and the resulting state consistency of the control and data planes upon fail-over events [211]. A first direction in this context is the 2-layer contractor-delegate hierarchy that we proposed in Section 3.3.4 of Chapter 3. Moreover, scaling the state management logic to multiple hundreds of thousands up to millions of prefixes can be achieved in two (complementary) ways: *(i)* using cloud-based deployments for dealing with the state storage requirements of the graph implementations on the controller's side, and *(ii)* using efficient prefix/flow space aggregation techniques by reusing graphs for prefixes or flows that have the same forwarding behavior on the AS

level. The latter approach may reduce the need for excessive storage resources as well as the computation times after inter-domain events.

Multi-Controller, Multi-Cluster Experiments. Future work includes modifying the technical implementation of the SIREN framework in order to support multiple SDN controllers and corresponding clients. At a first stage, coordination between these controllers could be achieved with classic BGP using the native per-cluster BGP speakers that the framework provides. At a next stage, new protocols for the coordination between multi-domain routing controllers could be employed instead of BGP; this would though require custom software for implementing this protocol on top of SIREN. Such experiments could investigate the effect of distributed clusters on convergence, and try to verify claims related to gradual centralization of different parts of the Internet and its impact on the behavior of routing convergence. In this case, convergence would be the result of the interplay between the multiple clusters with each other, between the clusters and external (BGP-governed) ASes, and between the external ASes among themselves. This experimental setup would yield even more realistic results.

Strategic Selection of Client ISPs. In the experiments of Chapter 4, ISP domains were selected uniformly at random from a set of available ISPs in different formations (graph scales and types). Future work could investigate whether smarter selection of ISPs would benefit convergence behavior with a very small percentage of SDN RaaS penetration within a multi-domain network. In particular, one could systematically explore strategies as the ones we employed for gradually attracting IXPs and their members in Chapter 6. In the convergence case for example, ISPs which have higher node degrees and node centrality index in the multi-domain network graph may be more appealing targets for the RaaS penetration, since the controller can take advantage of their location to drastically augment its bird's eye view and stabilize the network faster after a convergence trigger event. In summary, future work in this context could examine which characteristics make ISPs more appealing targets for RaaS deployment and the maximal ratio of impact to penetration; ideally a contractor should use ISPs that benefit the convergence behavior the most. Note though that such ISPs may be unwilling to adopt an outsourcing scheme unless substantial profit is imminent, since the related expertise is already "in-house" for big players that already have excellent inter-domain connectivity.

Policy Support. Support for the policy-compliant path calculation is another aspect of further technical extensions on the controller's side, combined with the policy interactions between inter-domain services running on top of the controller. The latter may be implemented based on policy composition and application modularization techniques, e.g., as implemented by the Pyretic framework [210]. Consistent policy composition and state update implementation by distributed SDN control planes could also be based on natural transactional interfaces, as proposed by Canini et al. [52]. Efficient algorithms for computing policy-compliant shortest paths and path diversity for arbitrary topologies, treating network policies as regular expressions and associated NFAs, have been already demonstrated in Chapter 5. We have investigated the support for the calculation of valley-free shortest paths, based on c2p, p2c and p2p policies, together with more relaxed versions of these policies. Future work could investigate more complex policies that are not well-known in the research community due to lack of public datasets. In this context, researchers could examine whether newly arising inter-ISP policy conflicts can be resolved in an efficient manner, or how they affect network properties such as path diversity.

New Abstractions and Services. A multi-domain routing control platform can become a vehicle for the deployment of novel services, which are hard to implement in today's environment. This requires the identification of the proper abstractions that will be offered to the services running on the RaaS platform. In this context, we can take advantage of layered control channel architectures using network programming languages [90] and compilers [186], identifying the proper *northbound* interface between the control platform and the multi-domain services. The virtualization/slicing abstraction [11] is another piece of the puzzle. One potential service that would be interesting to run on a cross-domain level using our platform, is collaborative defense against new DDoS attacks, such as the *Crossfire* link-flooding attack [136]. Such a service could for example take advantage of SDN-based traffic engineering for *joint* detection and mitigation [106]. Moreover, another aspect is the technical management of the advertisements of the controllers providing RaaS to the potential client domains, assuming a setting with multiple on-demand contractors that dynamically manage service-related slices of prospective clients' networks. The northbound interface should facilitate the requirement for dynamic selection of contractors. Apart from that, we point the reader to the recent work

by the IETF in the direction of eBGP controllers, based on the link-state routing abstraction, for the efficient deployment of inter-AS traffic engineering as a use case [165]. We note that the IETF work is oriented towards multi-AS setups consisting of multiple BGP ASNs *under the same administrative control* (e.g., the same ISP with multiple AS siblings), while our work is a superset of this case. Nevertheless, finding a common modus operandi would be desirable for the expansion of our inter-domain RaaS approach, which assumes multiple independent ASes served by the same contractor. A multi-sibling eBGP/SDN controller could be a first step in this direction, with a single ISP acting as a "self-contractor"; that is, it provides RaaS to its own sibling domains.

Taking Convergence out of the Critical Path. We would further like to explore maximally redundant techniques for fast re-routing on the IP layer, such as the ones that are currently under discussion in IETF [130]. The objective there is to minimally disrupt traffic upon re-routing, thus making the convergence process itself less of an issue. Moreover, we plan to investigate complementary IETF efforts on making Internet routing more scalable, such as LISP [81]. LISP can be used to reliably forward traffic to prefixes, even while the network is converging. It is also backwards compatible with BGP, reducing the amount of the needed signaling. We note that such mechanisms could be safely deployed within the sphere of influence of a SDN-based RaaS controller, benefiting client ASes while shielding the rest of the Internet from any associated issues during the hybrid deployment phase. Moreover, the relationship between convergence and packet loss should be further investigated within a hybrid BGP-SDN environment. Further research could also focus on the interaction between new convergence mechanisms (e.g., CRWI timers on the SDN cluster controller) with legacy BGP mechanisms such as the MRAI timer.

SIREN Framework Extensions. In the context of the SIREN framework, future technical work includes scaling up its capabilities in terms of the sizes of the networks it can support, in conjunction with ongoing advances in the Mininet emulation software base. Furthermore, in this work we primarily focused on eBGP and OpenFlow, but the framework could be extended to also support classic intra-domain routing protocols such as iBGP and IGP (OSPF, IS-IS). This would also extend the scope of the available abstractions, going beyond the "AS as a big switch" abstraction to perform more detailed and realistic

experiments. Furthermore, we would be interested in extending the emulation framework with real network equipment and actual inter-domain routing interactions, for example in collaboration with the MiniNext initiative [220] and the PEERING testbed [219].

Financial Incentives. Besides the incentives described in Chapter 3, supporting RaaS-oriented outsourcing, future work could investigate the following. *(i)* Research concrete financial gains in terms of absolute numbers (e.g., dollars per outsourcing step), based on actual data from ISPs. We note that this is a hard task due to the sensitive nature of such data. *(ii)* Investigate the economies of scale that an inter-domain RaaS approach may create in terms of gradual expansion, critical mass of clients per contractor, and the Nash equilibrium point, as well as if and when such a point can be reached together with its stability characteristics. *(iii)* Model the game between the involved entities, such as competing RaaS contractors, using game theory primitives taking both the flow of money and service value carefully into account.

7.4.2 Policy-compliant Min-Cuts and Graph Transform

The main insight that the implementation of the following future work tasks is expected to yield is the understanding of the effect of real operational ISP policies on the performance of the graph transformation algorithm and the generality of its application in different use cases.

NFA Decomposition. As a follow-up on the maximal biclique finding problem, described in Section 5.4.5, future research could investigate general solutions and heuristics dealing with the NP-hard nature of the problem. Moreover, another course of action could be devising a dual methodology for decomposing the transition sets of the NFA into cartesian products of subsets of the full set of states. As we explained, this problem is closely related to the biclique finding challenge, but maybe faster and/or more accurate heuristics can be found for tackling it using set-theoretic algorithms. Besides constructing the appropriate decomposition, one could also investigate how to limit its effect on the looseness of the calculated min-cut bounds. For example, one idea would be to mark the traversed paths that are subject to inflation while running the corresponding max-flow/min-cut algorithm. Afterwards, this information might be used to normalize

the yielded results closer to accuracy. Of course, this process requires modifications to the min-cut algorithm, but this is the trade-off for achieving more accurate results in the face of complex, non-aggregatable NFA transitions.

Exact-form NFAs. Further research could also be performed in the context of the following questions. Which policies correspond to exact-form NFAs (cf. (5.21) of Section 5.4.4), with fully aggregatable state transitions? Can we generalize these policies to a class of NFAs with certain properties? Are there policies for which only an approximation is feasible? Can this class also be generalized in a similar fashion? Such questions have been partly answered in this work, with proper examples described in Chapter 6 related to relaxations of valley-free policies; some lead to exact min-cut values while others do not. Future work could investigate the ability to generalize these policies and check how frequent each class is used in, e.g., an ISP's day-to-day policy-based operations. Moreover, are regular expressions, and consequently DFAs and NFAs enough to model the majority of used policies, from a network administrator's perspective? Which class of policies is useful but cannot be expressed via regular expressions? This would require deep insight into realistic ISP policies which are usually sensitive secrets, but would help to estimate the generality of such approaches. In case the NFAs are sufficient, then one could further investigate the complexity of state transitions that are used in practice, if such datasets are available.

Generalize Implementation. Another aspect of future work is the generalization of the technical algorithmic implementation (i.e., the Python framework), so that it can be applied in different use cases, beyond valley-free versions in the context of inter-domain routing and AS-level path diversity calculations. For example, an interesting avenue in this regard is the encapsulation of a graph transform version within a MPTCP stack, or its integration with an open-source SDN controller framework. In the latter case, the min-cut calculations might be used to aid the controller's decisions regarding multipath flow routing applications, e.g., used in an intra-/inter-domain RaaS implementation.

7.4.3 CXPs on IXP Multigraphs

The basic project—that could potentially form the basis of another PhD thesis—is the design and implementation of a fully fledged CXP system.

We note the following relevant engineering and research tasks. These points augment the CXP description of Section 6.2.3. The main insight that the implementation of these future work tasks is expected to yield is to find out whether the deployment of CXP systems can reach Internet scales in practice, both from a pure technical and a political/financial point of view.

CXP Workflow. One aspect of future work is the evaluation of the CXP system's workflow, time-wise. Example steps include: *(i)* the reception of the client's request, *(ii)* the retrieval of the state of the pathlet network, *(iii)* running the embedding algorithm, *(iv)* accepting/denying the request, and, if acceptance is possible, *(v)* pushing the new associated state to the CXP anchors, including the state related to the backup fail-over paths.

Client-to-CXP. A CXP should handle the communication between the client and the controller, after defining the proper interface(s) to be used. The interface-related calls could include the announcement, update, and withdrawal of the request from the client's side, as well as the corresponding response from the CXP controller.

ISP-to-CXP. A CXP should handle the communication between the ISP, i.e., the pathlet provider, and the controller, after defining the proper interface(s) to be used. The related calls in this context could include the advertisement, update, and withdrawal of pathlets from the ISP's side, as well as the corresponding response from the CXP.

IXP-to-CXP. A CXP should handle the communication between the IXP-based data plane anchor and the controller, after defining the proper interface(s) to be used. Since the anchor is the manifestation of the CXP's control on the data plane, we could have separate interfaces for the control, management and monitoring of the associated devices, with the choice of synchronous or asynchronous communication.

Algorithms as SDN Applications. A CXP operator could run the embedding algorithms that we presented in Chapter 6 over a production SDN controller as SDN applications. In this context, a CXP operator could use the online variants for quick, on-the-fly embeddings and the offline variants for periodic defragmentation of the pathlet network, as well as for evaluating the performance of the online variants in terms of utilization and acceptance ratios.

Monitoring. A CXP should handle the supervision of: *(i)* the quality of the flows on the anchor's level (including, e.g., loss rates

or congestion at the device's ports), *(ii)* the quality of the pathlets provided by the ISPs, and *(iii)* the end-to-end quality of the paths and the embedded flows, across domains. This process could be based on properly retrieving the state of the overlay network graph, including annotations on the anchors and pathlets that the CXP learns about.

Fail-over. A CXP should deal with failures and perform fast recovery, especially in the case of sensitive applications such as telesurgery (cf. Section 6.7.1). The system should be resilient against potential failures of the CXP controllers, anchors or pathlets. Regarding the pathlets, a failure could also be the degradation of the associated QoS levels rather than a complete loss of connectivity. We note that when pushing the new state to the anchors in order to recover from a failure, multiple aspects should be kept in mind. For example, the down-time and QoS degradation should be minimized, together with possible disruptions on other traffic passing via the failed pathlet or anchor. Furthermore, state updates should be consistent end-to-end in order to avoid blackholes or other potential routing pathologies that could further hurt the performance of the network service(s). "Hot-standby" CXP controllers should also be kept operational for resiliency. Path fail-over should be pushed to the data plane as much as possible in order to minimize reaction times and dependencies on the controller.

Elasticity. Building an elastic CXP controller is another aspect of future work. More load from the client base should lead to the spawning of more controllers, while dealing with scalability, handovers, state distribution and consistency challenges. In a similar context as the RaaS controller (cf. Section 7.4.1), a CXP operator should investigate the trade-offs associated with the selection of the location for the controller. One could start with data center-based deployments, which are already collocated with IXPs and thus with the targeted anchors. In this context, the following question could be interesting for future researchers and engineers: *How can CXP-like systems scale for billions of flows, i.e., achieve an Internet-scale operational level?*

TE Tasks. In terms of Traffic Engineering, a CXP should deal with proper traffic shaping on the edge anchors, where the client's access ISP connects the endpoint to the rest of the multigraph. This shaping should take into account the behavior of the related transport protocols, such as TCP, while dealing with traffic bursts that could be potentially accommodated with excess pathlets. Moreover, the proper forwarding of

traffic flows and end-to-end tunneling on the level of the anchors could be managed via classic techniques, such as MPLS label-switching. The practical isolation of cross-traffic on the data plane anchors should also be investigated, while our simulator [152] could be used for evaluating some coarse-grained effects of the co-existence of such traffic on the performance of the proposed path stitching algorithms.

Pilot Services. Another potential field of future work is the development of a small-scale testbed for testing the CXP concept, combined with an initial service offering. One course of action is to start with renting virtual machines in data centers, using them as CXP data plane anchors and switching elements. This makes sense location-wise, in case the cloud host is collocated with an IXP; the latter setup is usual in practice [57]. A pilot service to be considered is the (negative) waypoint routing service, where the pathlets are constructed between certain waypoints without assuming any participation from the ISPs; IP-in-IP tunneling may be utilized for such a purpose. In this case, the "guarantee" that the CXP itself could provide is the proper geolocation of the traffic packets, avoiding certain nodes (such as IXPs) over which the client does not want the traffic to pass. Bandwidth and latency guarantees from the ISPs' side could be gradually added to the offering, as the CXP starts to collaborate with more ISPs and IXPs.

Addressing Privacy Concerns. Future CXP implementations should investigate how to deal with the privacy-related challenges that were described in Section 6.2.3. In particular, the protection against privacy violations could be a cooperative effort between CXPs (e.g., trying to hide information from external end-users regarding path selection), and ISPs (e.g., advertising only pathlets at a fraction of the actual capacity and with a larger latency than the real one, while not violating any guarantees). This subject has both technical and political aspects that need to be taken into account when implementing and deploying a CXP system in the wild; the core of the "trusted third party" scheme can be though maintained, with the researchers focusing on the obfuscation of external inference practices rather than compromised or malicious CXP controllers (the latter case should be extremely infrequent in practice).

CXP Multigraph Analysis. Besides the analysis performed in this work, based on peering datasets, researchers could also investigate the dynamics of the multigraph over time in terms of capacity and

latencies. For example, one potential course of action is a measurement framework for estimating latency and available bandwidth between IXPs. The results could be further correlated with end-to-end measurements, in order to evaluate whether stitching and waypoint routing over IXPs could technically improve the quality of current inter-domain connections.

CXP Markets. Last but not least, a research direction that would be interesting as an inter-disciplinary topic is the investigation of the financial markets that can be based on CXPs, together with their dynamics. A starting point is the analysis done in the context of bandwidth transit markets [241] and route "bazaars" [78]. Besides bandwidth and latency, pathlets could, e.g., be annotated with cost, with the CXP trying to minimize the cost for the incoming client flows while at the same time accepting as many of them as possible.

Appendix A

List of Publications and Software Contributions

During the work on this dissertation, the following publications were written and released, supporting this thesis (in chronological order):

- **Main author**

  Kotronis, V., Dimitropoulos, X., and Ager, B. Outsourcing the Routing Control Logic: Better Internet Routing Based on SDN Principles. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks* (2012), ACM, pp. 55–60

  Kotronis, V., Dimitropoulos, X., and Ager, B. Outsourcing Routing using SDN: The Case for a Multi-Domain Routing Operating System. Poster Session of the Research Track of the 2nd Open Networking Summit (ONS), 2013

  Kotronis, V., Dimitropoulos, X., Klöti, R., Ager, B., Georgopoulos, P., and Schmid, S. Control Exchange Points: Providing QoS-enabled End-to-End Services via SDN-based Inter-Domain Routing Orchestration. In *Proceedings of the Research Track of the 3rd Open Networking Summit (ONS)* (2014)

  Kotronis, V., Gämperli, A., and Dimitropoulos, X. Routing Centralization Across Domains via SDN: A Model and

Emulation Framework for BGP Evolution. *Computer Networks* (2015)

Kotronis, V., Klöti, R., Rost, M., Georgopoulos, P., Ager, B., Schmid, S., and Dimitropoulos, X. Investigating the Potential of the Inter-IXP Multigraph for the Provisioning of Guaranteed End-to-End Services. Tech. Rep. 360, ETH Zurich, Laboratory TIK, Feb 2015

Kotronis, V., Klöti, R., Rost, M., Georgopoulos, P., Ager, B., Schmid, S., and Dimitropoulos, X. Investigating the Potential of the Inter-IXP Multigraph for the Provisioning of Guaranteed End-to-End Services. In *Poster Session of the ACM SIGMETRICS conference* (2015)

Kotronis, V., Klöti, R., Rost, M., Georgopoulos, P., Ager, B., Schmid, S., and Dimitropoulos, X. Stitching Inter-Domain Paths over Internet Exchange Points for Rich Routing Possibilities. In *under submission* (2015)

- **Co-author (with ∼40% contribution)**

  Gämperli, A., Kotronis, V., and Dimitropoulos, X. Evaluating the Effect of Centralization on Routing Convergence on a Hybrid BGP-SDN Emulation Framework. Demo Session of the ACM SIGCOMM conference, 2014

  Klöti, R., Kotronis, V., Ager, B., and Dimitropoulos, X. Policy-Compliant Path Diversity and Bisection Bandwidth. In *Proceedings of the IEEE INFOCOM conference* (2015)

- **Co-author (with ∼25% contribution)**

  Klöti, R., Ager, B., Kotronis, V., Nomikos, G., and Dimitropoulos, X. A Comparative Look into Public IXP Datasets. *under submission* (2015)

Additionally, we created, in collaboration with Adrian Gämperli, Rowan Klöti, Bernhard Ager, Matthias Rost and Stefan Schmid the following software suites:

- CXP experimentation (simulation framework) [152].
- SIREN framework and controller (emulation framework) [96].

Further research work that we co-authored in association to SDN (not directly supporting this thesis but going beyond its scope) is briefly presented in Chapter B. We give here the associated references of our secondary research contributions.

- **Secondary publications, main author**

  Kotronis, V., Schatzmann, D., and Ager, B. On Bringing Private Traffic into Public SDN Testbeds. Poster Session of the Second ACM SIGCOMM HotSDN Workshop, 2013

- **Secondary publications, co-author**

  Klöti, R., Kotronis, V., and Smith, P. Openflow: a Security Analysis. In *Proceedings of the NPSec Workshop on Secure Network Protocols, part of the IEEE ICNP conference* (2013), pp. 1–6

  Suñé, M., Bergesio, L., Woesner, H., Rothe, T., Köpsel, A., Colle, D., Puype, B., Simeonidou, D., Nejabati, R., Channegowda, M., et al. Design and Implementation of the OFELIA FP7 Facility: the European OpenFlow Testbed. *Computer Networks 61* (2014), 132–150

  Broadbent, M., Georgopoulos, P., Kotronis, V., Plattner, B., and Race, N. OpenCache: Leveraging SDN to Demonstrate a Customisable and Configurable Cache. Demo session of IEEE INFOCOM, 2014

  Gkounis, D., Kotronis, V., Liaskos, C., and Dimitropoulos, X. Efficient DDoS Link-Flooding Attack Mitigation using SDN. In *under submission* (2015)

We also contributed to software related to bringing private user traffic into public SDN testbeds, at https://bitbucket.org/vkotronis/of_privacy_proxy/wiki/Home, and to the software primarily authored by Rowan Klöti on comparing public IXP datasets [142]. In the context of our work on the OFELIA SDN project, we contributed to the software used for the testbed management at the associated GitHub repository at https://github.com/fp7-ofelia/ocf.

Appendix B

Further Research

This appendix contains the abstracts of further research work in which I participated either as a primary or as a secondary author, and which is relevant to the thesis as a complementary body of research.

OpenFlow Security. Together with Rowan Klöti and Paul Smith, we co-wrote a paper on a first security analysis of the OpenFlow protocol [145] using well-known methodologies such as attack tree modeling approaches. The analysis and evaluation approach are not meant to be exhaustive, but are intended to be adaptable and extensible to new versions and deployment contexts of OpenFlow.

The OFELIA Testbed. Together with our European partners, we co-wrote a paper on the design and implementation of the OFELIA FP7 facility [233], which served for several years as the core European OpenFlow testbed. Such SDN testbeds can be used as a public service by multiple experimenters around the world in order to deploy new ideas and innovative solutions to networking problems. The use of real network equipment at large scales surpasses the limits of simulation or emulation when testing new research ideas. In particular, the OFELIA project was initiated to design a pan-European testbed that connects individual testbeds installed by project partners, based on a common layer-2 (L2) infrastructure. Furthermore, it intended to support multi-layer and multi-technology experiments related to OpenFlow. The objective was to expose OpenFlow as the core service of the testbed, therefore allowing the users to precisely and dynamically control the

L2 experimental network itself, while at the same time try to limit the capabilities of OpenFlow as less as possible. The basic infrastructure, which is used to conduct experimentation in OFELIA and is offered as a service to the experimenter, is the shared L2 Ethernet OpenFlow network along with virtualized end-hosts. With this infrastructure a complete network experiment setup can be deployed. This setup includes end-hosts to establish a communication over the OpenFlow experimental network and to deploy the experiment controller(s) to program the forwarding plane of the network. The application end-hosts are made available as virtualized computing resources in an Infrastructure as a Service (IaaS)-like fashion. Real physical computing (servers), as well as other types of resources can also be provided by each individual island in cooperation with the experimenter to fulfill the needs of the experiment. To enable the experimenter to register to the facility, configure their experiments, request, setup and release resources, the OFELIA facility provides an experiment orchestration software or control framework (the OFELIA Control Framework [195]), that has been developed within the project. As ETH Zurich partners we helped in particular with the development and extension of the software related to the management and orchestration of OpenFlow resources within the testbed. Also, as a founding partner, we have built and maintained one of the testbed's islands between 2011-2015.

Bringing User Traffic to SDN Testbeds. Together with Dominik Schatzmann and Bernhard Ager, we co-wrote a paper on a novel approach on how a testbed operator can give privacy and availability guarantees to users who are willing to share part of their traffic with experimenters [156], thus making it less risky for users to opt-in to experiments. The motivation behind this work is that a common shortcoming among publicly available SDN testbeds, such as OFELIA, is the fact that they lack real user-driven Internet traffic for experimentation. While having real user traffic inside such testbeds is an indisputable advantage, the users' right for privacy and wish for availability of the network often make it impossible to simply make a testbed part of the communication path. This work has been augmented by the follow-up MSc thesis of Adrian Friedli that we co-advised [92].

Defeating DDoS Link-flooding Attacks via SDN. Together with Dimitrios Gkounis, Christos Liaskos and Xenofontas Dimitropoulos, we participate in an ongoing work on a novel approach [106] exploiting

online traffic engineering to detect and mitigate a new class of Distributed Denial of Service (DDoS) link-flooding attacks. Such attacks are very hard to defend against and could disconnect entire countries from the Internet. This work proposes using online Traffic Engineering (TE) in a novel way as an approach to detect and mitigate such costly attacks. Our approach can exploit existing TE algorithms. In addition, we introduce a new TE approach that, besides load balancing the attack traffic and detecting the malicious sources, also aims at reducing the number of routing changes needed for online TE. We build a full prototype of our proposal, applying SDN principles, and show that it is possible to detect attackers fast, with a moderate number of TE routing changes. The approach is akin to Software Defined Networking (SDN) principles, which render the management of network traffic more flexible through centralized flow-level control. Our prototype was built on an emulated SDN environment using OpenFlow to interface with the network devices, accompanied by a custom simulator. We further discuss useful insights gained from our experiments as well as a number of open research questions. Such a defense method might also be offered "as a service" on an inter-domain level over a multi-AS routing control platform, in order to deal with large-scale WAN attacks involving large numbers of bots and affected ASes. This work will continue beyond the end of the current thesis, as there is a lot of potential and interest on such ideas especially in light of recent attacks with detrimental effects on certain parts of the Internet [205]. This work is currently under submission.

SDN-aided Content Caching. Together with partners from the Lancaster University, we helped with the experimentation on SDN-aided caching mechanisms. The approach leverages SDN concepts and mechanisms in order to transparently store and deliver content from a local cache to the client, thus lightening the load on the WAN and relieving the necessity for urgent network capacity upgrades. An open interface to the cache presents owners with new possibilities for cache control and maintenance. This demonstration [38] showcased a prototype implementation in action on a large-scale OpenFlow testbed deployed across Europe, the OFELIA testbed. Such SDN-enabled smart caching mechanisms may also be of value on the multi-AS level, under the supervision of a multi-AS routing control platform.

Bibliography

[1] ExaBGP: The BGP Swiss Army Knife of Networking. https://github.com/Exa-Networks/exabgp.

[2] GEANT Bandwidth-on-Demand Provisioning Tool. http://geant3.archive.geant.net/service/autobahn/about_autoBAHN/Pages/About_autoBAHN.aspx.

[3] IX-F: Internet eXchange Federation. http://www.ix-f.net/.

[4] Open vSwitch: Production Quality, Multilayer Open Virtual Switch. http://openvswitch.org/.

[5] POX GitHub Code Repository. https://github.com/noxrepo/pox.

[6] Quagga Routing Software Suite. http://www.nongnu.org/quagga/.

[7] Ager, B., Chatzis, N., Feldmann, A., Sarrar, N., Uhlig, S., and Willinger, W. Anatomy of a Large European IXP. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication* (2012), ACM, pp. 163–174.

[8] Ahmad, M. Z., and Guha, R. Studying the Effect of Internet eXchange Points on Internet Link Delays. In *Proceedings of the 2010 Spring Simulation Multiconference* (2010), Society for Computer Simulation International, p. 103.

[9] Ahmad, M. Z., and Guha, R. A Tale of Nine Internet eXchange Points: Studying Path Latencies Through Major Regional IXPs. In *LCN* (2012), Citeseer, pp. 618–625.

[10] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., 1993.

[11] Al-Shabibi, A., De Leenheer, M., Gerola, M., Koshibe, A., Parulkar, G., Salvadori, E., and Snow, B. OpenVirteX: Make Your Virtual SDNs Programmable. In *Proceedings of the third workshop on Hot topics in software defined networking* (2014), ACM, pp. 25–30.

[12] Alaettinoglu, C., Bates, T., Gerich, E., Karrenberg, D., Meyer, D., Terpstra, M., and Villamizar, C. Routing Policy Specification Language (RPSL). RFC 2280 (Proposed Standard), Jan. 1998. Obsoleted by RFC 2622.

[13] Alim, M. A., and Griffin, T. G. Hybrid Link-state, Path-vector Routing. In *Proceedings of the Sixth Asian Internet Engineering Conference* (2010), ACM, pp. 32–39.

[14] AMS-IX. AMSterdam Internet eXchange. https://ams-ix.net/.

[15] AMS-IX. AMSterdam Internet eXchange Mobile Peering. https://ams-ix.net/services-pricing/mobile-peering/grx.

[16] Anderson, T. Networking as a Service, HOTI-21 keynote. http://www.youtube.com/watch?v=bEtq_4arFz0, 2013.

[17] Arlinghaus, S. L., Arlinghaus, W. C., and Harary, F. *Graph Theory and Geography: An Interactive View*. Wiley New York, 2001.

[18] Ascigil, O., Calvert, K. L., and Griffioen, J. N. On the Scalability of Interdomain Path Computations. In *Networking Conference, 2014 IFIP* (2014), IEEE, pp. 1–9.

[19] Aspnes, J., Azar, Y., Fiat, A., Plotkin, S., and Waarts, O. On-line Load Balancing with Applications to Machine Scheduling and Virtual Circuit Routing. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing* (1993), ACM, pp. 623–631.

[20] Augustin, B., Krishnamurthy, B., and Willinger, W. IXPs: Mapped? In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference* (2009), ACM, pp. 336–349.

[21] Aurrecoechea, C., Campbell, A. T., and Hauw, L. A Survey of QoS Architectures. *Multimedia systems 6*, 3 (1998), 138–151.

[22] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and Xiao, X. Overview and Principles of Internet Traffic Engineering. RFC 3272 (Informational), May 2002. Updated by RFC 5462.

[23] Awerbuch, B., Azar, Y., Plotkin, S. A., and Waarts, O. Competitive Routing of Virtual Circuits with Unknown Duration. In *SODA* (1994), pp. 321–327.

[24] Baldine, I., Xin, Y., Mandal, A., Ruth, P., Heerman, C., and Chase, J. Exogeni: a Multi-Domain Infrastructure-as-a-Service Testbed. In *Testbeds and Research Infrastructure. Development of Networks and Communities*. Springer, 2012, pp. 97–113.

[25] Ballani, H., Francis, P., and Zhang, X. A Study of Prefix Hijacking and Interception in the Internet. In *ACM SIGCOMM Computer Communication Review* (2007), vol. 37, ACM, pp. 265–276.

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

[27] Bauer, D. N., Dechouniotis, D., Dimitropoulos, C.-X., and Kind, A. Valley-free Shortest Path Method, Mar. 15 2011. US Patent 7,907,596.

[28] Bell, G. Failure to Thrive: QoS and the Culture of Operational Networking. In *Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?* (2003), ACM, pp. 115–120.

[29] Bennesby, R., Mota, E., Fonseca, P., and Passito, A. Innovating on Interdomain Routing with an Inter-SDN Component. In *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on* (2014), IEEE, pp. 131–138.

[30] Benson, T., Akella, A., and Maltz, D. A. Unraveling the Complexity of Network Management. In *NSDI* (2009), pp. 335–348.

[31] Benson, T., Akella, A., and Shaikh, A. Demystifying Configuration Challenges and Trade-offs in Network-based ISP Services. *ACM SIGCOMM Computer Communication Review 41*, 4 (2011), 302–313.

[32] Benson, T., Akella, A., Shaikh, A., and Sahu, S. CloudNaaS: a Cloud Networking Platform for Enterprise Applications. In *Proceedings of the 2nd ACM Symposium on Cloud Computing* (2011), ACM, p. 8.

[33] Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., et al. ONOS: Towards an Open, Distributed SDN OS. In *Proceedings of the third workshop on Hot topics in software defined networking* (2014), ACM, pp. 1–6.

[34] BGP4.AS. BGP Looking Glasses for IPv4/IPv6, Traceroute and BGP Route Servers. http://www.bgp4.as/looking-glasses.

[35] Binkele-Raible, D., Fernau, H., Gaspers, S., and Liedloff, M. Exact Exponential-Time Algorithms for Finding Bicliques. *Information Processing Letters 111*, 2 (2010), 64–67.

[36] Bistarelli, S., Montanari, U., Rossi, F., and Santini, F. Unicast and Multicast QoS Routing with Soft-Constraint Logic Programming. *ACM Transactions on Computational Logic (TOCL) 12*, 1 (2010), 5.

[37] Bremler-Barr, A., Chen, N., Kangasharju, J., Mokryn, O., and Shavitt, Y. Bringing Order to BGP: Decreasing Time and Message Complexity. *Computer Networks 53*, 12 (2009), 2241–2256.

[38] Broadbent, M., Georgopoulos, P., Kotronis, V., Plattner, B., and Race, N. OpenCache: Leveraging SDN to Demonstrate a Customisable and Configurable Cache. Demo session of IEEE INFOCOM, 2014.

[39] Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., and Zhao, L. Statistical Analysis of a Telephone Call Center: a Queueing-Science Perspective. *Journal of the American statistical association 100*, 469 (2005), 36–50.

[40] Brown, M. Pakistan Hijacks YouTube. http://research.dyn.com/2008/02/pakistan-hijacks-youtube-1/. Dyn Research Blog.

[41] Brown, M. A., Popescu, A., and Zmijewski, E. Peering Wars: Lessons Learned from the Cogent-Telia De-peering. Presentation, Renesys Corp, New York, 2008.

[42] Brüggemann-Klein, A. Regular Expressions into Finite Automata. *Theoretical Computer Science 120*, 2 (1993), 197–213.

[43] BSNL. Leased Line Services with Flexible Access Bandwidth. http://bsnl.in/opencms/bsnl/BSNL/services/enterprises/mlln.html.

[44] Buldyrev, S. V., Parshani, R., Paul, G., Stanley, H. E., and Havlin, S. Catastrophic Cascade of Failures in Interdependent Networks. *Nature 464*, 7291 (2010), 1025–1028.

[45] Caesar, M., Caldwell, D., Feamster, N., Rexford, J., Shaikh, A., and van der Merwe, J. Design and Implementation of a Routing Control Platform. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2* (2005), USENIX Association, pp. 15–28.

[46] Caesar, M., Casado, M., Koponen, T., Rexford, J., and Shenker, S. Dynamic Route Recomputation Considered Harmful. *ACM SIGCOMM Computer Communication Review 40*, 2 (2010), 66–71.

[47] Caesar, M., and Rexford, J. BGP Routing Policies in ISP Networks. *Network, IEEE 19*, 6 (2005), 5–11.

[48] CAIDA. AS Relationships Dataset (acquired on 2013-11-01). http://www.caida.org/data/as-relationships/.

[49] CAIDA. CAIDA Data Overview. http://www.caida.org/data/.

[50] CAIDA. IPv4 AS Links Dataset. http://data.caida.org/datasets/topology/ark/ipv4/as-links/.

[51] CAIDA. Routeviews Prefix to AS mappings Dataset for IPv4 and IPv6 (acquired on 2014-01-11). http://www.caida.org/data/routing/routeviews-prefix2as.xml.

[52] Canini, M., Kuznetsov, P., Levin, D., Schmid, S., et al. A Distributed and Robust SDN Control Plane for Transactional Network Updates. In *Proceedings of the IEEE INFOCOM conference* (2015).

[53] Carl, G., and Kesidis, G. Large-scale Testing of the Internet's Border Gateway Protocol (BGP) via Topological Scale-down. *ACM Transactions on Modeling and Computer Simulation (TOMACS) 18*, 3 (2008), 11.

[54] Castro, I., Cardona, J. C., Gorinsky, S., and Francois, P. Remote Peering: More Peering without Internet Flattening. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies* (2014), ACM, pp. 185–198.

[55] Cetinkaya, E. K., Alenazi, M. J., Peck, A. M., Rohrer, J. P., and Sterbenz, J. P. Multilevel Resilience Analysis of Transportation and Communication Networks. *Telecommunications Systems* (2013).

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

[57] Chatzis, N., Smaragdakis, G., Feldmann, A., and Willinger, W. There is More to IXPs Than Meets the Eye. *ACM SIGCOMM Computer Communication Review 43*, 5 (2013), 19–28.

[58] Chaudhury, R., and Terfloth, C. The Lure of Network Outsourcing. *White paper, Oliver Wyman* (2007).

[59] Chen, E., and Yuan, J. Autonomous-System-Wide Unique BGP Identifier for BGP-4. RFC 6286 (Proposed Standard), June 2011.

[60] Cheng, Y., Li, J., and Sterbenz, J. P. Path Geo-Diversification: Design and Analysis. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2013 5th International Congress on* (2013), IEEE, pp. 46–53.

[61] China Telecom. MPLS VPN Tariff. http://www.chinatelecomglobal.com/www/other/MPLSVPN.

[62] Chow, S. T., Wiemer, D., and Robert, J.-M. Distributed Defence Against DDoS Attacks, July 5 2007. US Patent App. 11/822,341.

[63] Chowdhury, M., Samuel, F., and Boutaba, R. Polyvine: Policy-based Virtual Network Embedding Across Multiple Domains. In *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures* (2010), ACM, pp. 49–56.

[64] CISCO. BGP Best Path Selection Algorithm. http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html.

[65] Civanlar, S., Parlakisik, M., Tekalp, A. M., Gorkemli, B., Kaytaz, B., and Onem, E. A QoS-Enabled Openflow Environment for Scalable Video Streaming. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE* (2010), IEEE, pp. 351–356.

[66] Clark, D. The Design Philosophy of the DARPA Internet Protocols. *ACM SIGCOMM Computer Communication Review 18*, 4 (1988), 106–114.

[67] Clark, D. D., Wroclawski, J., Sollins, K. R., and Braden, R. Tussle in Cyberspace: Defining Tomorrow's Internet. In *ACM SIGCOMM Computer Communication Review* (2002), vol. 32, ACM, pp. 347–356.

[68] Costa, P., Migliavacca, M., Pietzuch, P., and Wolf, L. A. NaaS: Network-as-a-Service in the Cloud. In *Proceedings of Hot-ICE* (2012).

[69] Cowie, J. Brazil Leak: If a Tree falls in the Rainforest... http://research.dyn.com/2008/11/brazil-leak-if-a-tree-falls-in. Dyn Research Blog.

[70] Cruvinel, L., and Vazão, T. Improving Performance for Multimedia Traffic with Distributed Dynamic QoS Adaptation. *Computer Communications 34*, 10 (2011), 1222–1234.

[71] Dhamdhere, A., and Dovrolis, C. The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh. In *Proceedings of the 6th International COnference* (2010), ACM, p. 21.

[72] Dinitz, Y., Garg, N., and Goemans, M. X. On the Single-Source Unsplittable Flow Problem. *Combinatorica 19*, 1 (1999), 17–41.

[73] Drioli, C., Allocchio, C., and Buso, N. Networked Performances and Natural Interaction via LOLA: Low Latency High Quality A/V Streaming System. In *Information Technologies for Performing Arts, Media Access, and Entertainment*. Springer, 2013, pp. 240–250.

[74] Elmokashfi, A., Kvalbein, A., and Dovrolis, C. On the Scalability of BGP: The Role of Topology Growth. *Selected Areas in Communications, IEEE Journal on 28*, 8 (2010), 1250–1261.

[75] Elmokashfi, A., Kvalbein, A., and Dovrolis, C. BGP Churn Evolution: a Perspective from the Core. *Networking, IEEE/ACM Transactions on 20*, 2 (2012), 571–584.

[76] Eppstein, D. Finding the k Shortest Paths. *SIAM Journal on computing 28*, 2 (1998), 652–673.

[77] Erlebach, T., Hall, A., Panconesi, A., and Vukadinović, D. Cuts and Disjoint Paths in the Valley-Free Path Model of Internet BGP Routing. In *Combinatorial and Algorithmic Aspects of Networking* (2005), Springer, pp. 49–62.

[78] Esquivel, H., Muthukrishnan, C., Niu, F., Chawla, S., and Akella, A. RouteBazaar: An Economic Framework for Flexible Routing. Tech. rep., Technical Report 1654, UW-Madison, 2009.

[79] Euro-IX. European Internet Exchange Association. https://www.euro-ix.net/. Dataset acquired on 2014-04-09.

[80] Fan, J., and Ammar, M. H. Dynamic Topology Configuration in Service Overlay Networks: a Study of Reconfiguration Policies. In *INFOCOM* (2006).

[81] Farinacci, D., Fuller, V., Meyer, D., and Lewis, D. The Locator/ID Separation Protocol (LISP). RFC 6830 (Experimental), Jan. 2013.

[82] Farrel, A., Vasseur, J.-P., and Ash, J. A Path Computation Element (PCE)-Based Architecture. RFC 4655 (Informational), Aug. 2006.

[83] Faucheur, F. L. Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering. RFC 4124 (Proposed Standard), June 2005.

[84] Feamster, N., and Balakrishnan, H. Detecting BGP Configuration Faults with Static Analysis. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2* (2005), USENIX Association, pp. 43–56.

[85] Feamster, N., Balakrishnan, H., Rexford, J., Shaikh, A., and Van Der Merwe, J. The Case for Separating Routing from Routers. In *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture* (2004), ACM, pp. 5–12.

[86] Feamster, N., Johari, R., and Balakrishnan, H. Implications of Autonomy for the Expressiveness of Policy Routing. *ACM SIGCOMM Computer Communication Review 35*, 4 (2005), 25–36.

[87] Fischer, A., Botero, J. F., Till Beck, M., De Meer, H., and Hesselbach, X. Virtual Network Embedding: a Survey. *Communications Surveys & Tutorials, IEEE 15*, 4 (2013), 1888–1906.

[88] Ford, A., Raiciu, C., Handley, M., and Bonaventure, O. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824 (Experimental), Jan. 2013.

[89] Ford, L., and Fulkerson, D. R. *Flows in Networks*, vol. 1962. Princeton University Press, 1962.

[90] Foster, N., Harrison, R., Freedman, M. J., Monsanto, C., Rexford, J., Story, A., and Walker, D. Frenetic: a Network Programming Language. In *ACM SIGPLAN Notices* (2011), vol. 46, ACM, pp. 279–291.

[91] Fréville, A., and Hanafi, S. The Multidimensional 0-1 Knapsack ProblemŮBounds and Computational Aspects. *Annals of Operations Research 139*, 1 (2005), 195–227.

[92] Friedli, A. Policy-based Injection of Private Traffic into a Public SDN Testbed. Master's thesis, ETH Zurich, 2013.

[93] Frikha, A., and Lahoud, S. Performance Evaluation of Pre-Computation Algorithms for Inter-Domain QoS Routing. In *Telecommunications (ICT), 2011 18th International Conference on* (2011), IEEE, pp. 327–332.

[94] Fu, J., Sjödin, P., and Karlsson, G. Intra-Domain Routing Convergence with Centralized Control. *Computer Networks 53*, 18 (2009), 2985–2996.

[95] Gämperli, A. Evaluating the Effect of SDN Centralization on Internet Routing Convergence. Master's thesis, ETH Zurich, 2014.

[96] Gämperli, A., and Kotronis, V. SIREN Public Code Repository. https://bitbucket.org/gaadrian/siren/downloads. Bitbucket Public Code Repository.

[97] Gämperli, A., Kotronis, V., and Dimitropoulos, X. Evaluating the Effect of Centralization on Routing Convergence on a Hybrid BGP-SDN Emulation Framework. Demo Session of the ACM SIGCOMM conference, 2014.

[98] Gao, L., and Rexford, J. Stable Internet Routing without Global Coordination. *IEEE/ACM Transactions on Networking (TON) 9*, 6 (2001), 681–692.

[99] Garroppo, R. G., Giordano, S., and Tavanti, L. A Survey on Multi-constrained Optimal Path Computation: Exact and Approximate Algorithms. *Computer Networks 54*, 17 (2010), 3081–3107.

[100] Geleji, G., Perros, H. G., Xin, Y., and Beyene, T. A Performance Analysis of Inter-Domain QoS Routing Schemes Based on Path Computation Elements. In *High Capacity Optical Networks and Enabling Technologies, 2008. HONET 2008. International Symposium on* (2008), IEEE, pp. 146–152.

[101] Gibb, G., Zeng, H., and McKeown, N. Outsourcing Network Functionality. In *Proceedings of the first workshop on Hot topics in software defined networks* (2012), ACM, pp. 73–78.

[102] Gill, P., Schapira, M., and Goldberg, S. A Survey of Interdomain Routing Policies. *ACM SIGCOMM Computer Communication Review 44*, 1 (2013), 28–34.

[103] Gill, R. K. *Effect of MRAI Timers on BGP Convergence Times*. PhD thesis, Applied Sciences: School of Engineering Science, 2013.

[104] Giotsas, V., Luckie, M., Huffaker, B., et al. Inferring Complex AS Relationships. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (2014), ACM, pp. 23–30.

[105] Gkounis, D. Cross-Domain DoS Link-Flooding Attack Detection and Mitigation Using SDN Principles. Master's thesis, ETH Zurich, 2014.

[106] Gkounis, D., Kotronis, V., Liaskos, C., and Dimitropoulos, X. Efficient DDoS Link-Flooding Attack Mitigation using SDN. In *under submission* (2015).

[107] Godfrey, P., Caesar, M., Haken, I., Singer, Y., Shenker, S., and Stoica, I. Stabilizing Route Selection in BGP. *Networking, IEEE/ACM Transactions on 23*, 1 (2015), 282–299.

[108] Godfrey, P., Ganichev, I., Shenker, S., and Stoica, I. Pathlet Routing. *ACM SIGCOMM Computer Communication Review 39*, 4 (2009), 111–122.

[109] Greene, R. B., and Smith, P. Cisco ISP Software and Router Management. http://www.ciscopress.com/articles/article.asp?p=27137&seqNum=3.

[110] Gregori, E., Improta, A., Lenzini, L., and Orsini, C. The Impact of IXPs on the AS-level Topology Structure of the Internet. *Computer Communications 34*, 1 (2011), 68–82.

[111] Griffin, T. G., and Premore, B. J. An Experimental Analysis of BGP Convergence Time. In *Network Protocols, 2001. Ninth International Conference on* (2001), IEEE, pp. 53–61.

[112] Griffin, T. G., Shepherd, F. B., and Wilfong, G. The Stable Paths Problem and Interdomain Routing. *IEEE/ACM Transactions on Networking (ToN) 10*, 2 (2002), 232–243.

[113] Griffin, T. G., and Wilfong, G. An Analysis of BGP Convergence Properties. *ACM SIGCOMM Computer Communication Review 29*, 4 (1999), 277–288.

[114] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and Shenker, S. NOX: Towards an Operating System for Networks. *ACM SIGCOMM Computer Communication Review 38*, 3 (2008), 105–110.

[115] Guo, H., Su, W., Zhang, H., and Kuo, S.-Y. On the Convergence Condition and Convergence Time of BGP. *Computer Communications 34*, 2 (2011), 192–199.

[116] Gupta, A., Kleinberg, J., Kumar, A., Rastogi, R., and Yener, B. Provisioning a Virtual Private Network: a Network Design Problem for Multicommodity Flow. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing* (2001), ACM, pp. 389–398.

[117] Gupta, A., Vanbever, L., Shahbaz, M., Donovan, S. P., Schlinker, B., Feamster, N., Rexford, J., Shenker, S., Clark, R., and Katz-Bassett, E. SDX: a Software Defined Internet eXchange. In *Proceedings of the 2014 ACM conference on SIGCOMM* (2014), ACM, pp. 551–562.

[118] Gupta, D., Segal, A., Panda, A., Segev, G., Schapira, M., Feigenbaum, J., Rexford, J., and Shenker, S. A New Approach to Interdomain Routing Based on Secure Multi-party Computation. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks* (2012), ACM, pp. 37–42.

[119] Gurobi Optimization. Gurobi Optimizer Reference Manual. http://www.gurobi.com, 2014.

[120] Haidegger, T., and Benyó, Z. *Extreme Telesurgery*. INTECH Open Access Publisher, 2010.

[121] Han, H., Shakkottai, S., Hollot, C., Srikant, R., and Towsley, D. Multi-path TCP: a Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet. *IEEE/ACM Transactions on Networking (TON) 14*, 6 (2006), 1260–1271.

[122] Heller, B., Sherwood, R., and McKeown, N. The Controller Placement Problem. In *Proceedings of the first workshop on Hot topics in software defined networks* (2012), ACM, pp. 7–12.

[123] Hu, C., Chen, K., Chen, Y., Liu, B., and Vasilakos, A. V. A Measurement Study on Potential Inter-Domain Routing Diversity. *Network and Service Management, IEEE Transactions on 9*, 3 (2012), 268–278.

[124] Hu, F., Hao, Q., and Bao, K. A Survey on Software Defined Networking (SDN) and Openflow: from Concept to Implementation.

[125] Hunt, R. A Review of Quality of Service Mechanisms in IP-based NetworksŮIntegrated and Differentiated services, Multi-layer Switching, MPLS and Traffic Engineering. *Computer Communications 25*, 1 (2002), 100–108.

[126] Hurricane Electric Internet Services. Network Looking Glass. http://lg.he.net/. Dataset acquired on 2014-04-10.

[127] Huston, G. BGP in 2013 - The Churn Report . http://www.potaroo.net/ispcol/2014-02/upds.html.

[128] Huston, G., Rossi, M., and Armitage, G. A Technique for Reducing BGP Update Announcements Through Path Exploration Damping. *Selected Areas in Communications, IEEE Journal on 28*, 8 (2010), 1271–1286.

[129] ICT-PACE. PACE: Next Steps in PAth Computation Element (PCE) Architectures. http://www.ict-pace.net/.

[130] IETF Routing Area Working Group (RTGWG). Charter for Working Group. https://datatracker.ietf.org/wg/rtgwg/charter/.

[131] International Telecommunication Union. Introduction to CCITT Signalling System No. 7. Recommendation Q.700, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Mar. 1993.

[132] InvisibleHand Networks, Inc. Main Website. http://www.invisiblehand.net/.

[133] Iqbal, H. *A Logically Centralized Approach for Control and Management of Large Computer Networks*. PhD thesis, University of Pittsburgh, 2013.

[134] Jäger, D. Analyzing the Time Dynamics in IXP Datasets. Master's thesis, Semester thesis. ETH Zurich, 2015.

[135] Jarraya, Y., Madi, T., and Debbabi, M. A Survey and a Layered Taxonomy of Software-Defined Networking.

[136] Kang, M. S., Lee, S. B., and Gligor, V. D. The Crossfire Attack. In *Security and Privacy (SP), 2013 IEEE Symposium on* (2013), IEEE, pp. 127–141.

[137] Kayaaslan, E. On Enumerating All Maximal Bicliques of Bipartite Graphs. In *CTW* (2010), pp. 105–108.

[138] Kazemian, P., Chan, M., Zeng, H., Varghese, G., McKeown, N., and Whyte, S. Real Time Network Policy Checking Using Header Space Analysis. In *NSDI* (2013), pp. 99–111.

[139] Kazemian, P., Varghese, G., and McKeown, N. Header Space Analysis: Static Checking for Networks. In *NSDI* (2012), pp. 113–126.

[140] Kent, S., Lynn, C., and Seo, K. Secure Border Gateway Protocol (S-BGP). *Selected Areas in Communications, IEEE Journal on 18*, 4 (2000), 582–592.

[141] Khare, V., Ju, Q., and Zhang, B. Concurrent Prefix Hijacks: Occurrence and Impacts. In *Proceedings of the 2012 ACM conference on Internet measurement conference* (2012), ACM, pp. 29–36.

[142] Klöti, R., Ager, B., and Kotronis, V. Software related to a "Comparative Look into Public IXP Datasets: Partially Consistent". https://bitbucket.org/RKloti/a-comparative-look-into-public-ixp-datasets-partially.git. Bitbucket Public Code Repository.

[143] Klöti, R., Ager, B., Kotronis, V., Nomikos, G., and Dimitropoulos, X. A Comparative Look into Public IXP Datasets. *under submission* (2015).

[144] Klöti, R., Kotronis, V., Ager, B., and Dimitropoulos, X. Policy-Compliant Path Diversity and Bisection Bandwidth. In *Proceedings of the IEEE INFOCOM conference* (2015).

[145] Klöti, R., Kotronis, V., and Smith, P. Openflow: a Security Analysis. In *Proceedings of the NPSec Workshop on Secure Network Protocols, part of the IEEE ICNP conference* (2013), pp. 1–6.

[146] Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T., et al. Onix: a Distributed Control Platform for Large-Scale Production Networks. In *OSDI* (2010), vol. 10, pp. 1–6.

[147] Korkmaz, T., and Krunz, M. Multi-constrained Optimal Path Selection. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (2001), vol. 2, IEEE, pp. 834–843.

[148] Kotronis, V., Dimitropoulos, X., and Ager, B. Outsourcing the Routing Control Logic: Better Internet Routing Based on SDN Principles. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks* (2012), ACM, pp. 55–60.

[149] Kotronis, V., Dimitropoulos, X., and Ager, B. Outsourcing Routing using SDN: The Case for a Multi-Domain Routing Operating System. Poster Session of the Research Track of the 2nd Open Networking Summit (ONS), 2013.

[150] Kotronis, V., Dimitropoulos, X., Klöti, R., Ager, B., Georgopoulos, P., and Schmid, S. Control Exchange Points: Providing QoS-enabled End-to-End Services via SDN-based Inter-Domain Routing Orchestration. In *Proceedings of the Research Track of the 3rd Open Networking Summit (ONS)* (2014).

[151] Kotronis, V., Gämperli, A., and Dimitropoulos, X. Routing Centralization Across Domains via SDN: A Model and Emulation Framework for BGP Evolution. *Computer Networks* (2015).

[152] Kotronis, V., Klöti, R., Rost, M., Ager, B., and Schmid, S. Software for CXP-based Experimentation. https://bitbucket.org/vkotronis/cxp_experimentation. Bitbucket Public Code Repository.

[153] Kotronis, V., Klöti, R., Rost, M., Georgopoulos, P., Ager, B., Schmid, S., and Dimitropoulos, X. Investigating the Potential of the Inter-IXP Multigraph for the Provisioning of Guaranteed End-to-End Services. Tech. Rep. 360, ETH Zurich, Laboratory TIK, Feb 2015.

[154] Kotronis, V., Klöti, R., Rost, M., Georgopoulos, P., Ager, B., Schmid, S., and Dimitropoulos, X. Investigating the Potential of the Inter-IXP Multigraph for the Provisioning of Guaranteed End-to-End Services. In *Poster Session of the ACM SIGMETRICS conference* (2015).

[155] Kotronis, V., Klöti, R., Rost, M., Georgopoulos, P., Ager, B., Schmid, S., and Dimitropoulos, X. Stitching Inter-Domain Paths over Internet Exchange Points for Rich Routing Possibilities. In *under submission* (2015).

[156] Kotronis, V., Schatzmann, D., and Ager, B. On Bringing Private Traffic into Public SDN Testbeds. Poster Session of the Second ACM SIGCOMM HotSDN Workshop, 2013.

[157] Kreutz, D., Ramos, F. M., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. Software-Defined Networking: a Comprehensive Survey. *proceedings of the IEEE 103*, 1 (2015), 14–76.

[158] Kühne, M. AS Path Lengths Over Time. https://labs.ripe.net/Members/mirjam/update-on-as-path-lengths-over-time, 2012. RIPE NCC Article.

[159] Kushman, N., Kandula, S., and Katabi, D. Can You Hear Me Now?!: It Must Be BGP. *ACM SIGCOMM Computer Communication Review 37*, 2 (2007), 75–84.

[160] Labovitz, C., Ahuja, A., Bose, A., and Jahanian, F. Delayed Internet Routing Convergence. *ACM SIGCOMM Computer Communication Review 30*, 4 (2000), 175–187.

[161] Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., and Jahanian, F. Internet Inter-Domain Traffic. *ACM SIGCOMM Computer Communication Review 41*, 4 (2011), 75–86.

[162] Lakshminarayanan, K., Stoica, I., and Shenker, S. Routing as a Service. Tech. rep. ucb-cs-04-1327, UC Berkeley, 2004.

[163] Lambert, A., Buob, M.-O., and Uhlig, S. Improving Internet-wide Routing Protocols Convergence with MRPC Timers. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies* (2009), ACM, pp. 325–336.

[164] Lantz, B., Heller, B., and McKeown, N. A Network in a Laptop: Rapid Prototyping for Software-Defined Networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks* (2010), ACM, p. 19.

[165] Lapukhov, P., and Nkposong, E. Centralized Routing Control in BGP Networks Using Link-State Abstraction. Network Working Group Internet-Draft (Informational), Sept. 2013. Expired.

[166] Lee, J.-N., Huynh, M. Q., Kwok, R. C.-W., and Pi, S.-M. IT Outsourcing Evolution—: Past, Present, and Future. *Communications of the ACM 46*, 5 (2003), 84–89.

[167] Levin, D., Wundsam, A., Heller, B., Handigol, N., and Feldmann, A. Logically Centralized?: State Distribution Trade-offs in Software Defined Networks. In *Proceedings of the first workshop on Hot topics in software defined networks* (2012), ACM, pp. 1–6.

[168] Levis, P., and Boucadair, M. Considerations of Provider-to-Provider Agreements for Internet-Scale Quality of Service (QoS). RFC 5160 (Informational), Mar. 2008.

[169] Li, J., Guidero, M., Wu, Z., Purpus, E., and Ehrenkranz, T. BGP Routing Dynamics Revisited. *ACM SIGCOMM Computer Communication Review 37*, 2 (2007), 5–16.

[170] Lodhi, A., Dhamdhere, A., Laoutaris, N., and Dovrolis, C. Complexities in Internet Peering: Understanding the "Black" in the "Black Art". In *Proceedings of the IEEE INFOCOM conference* (2015).

[171] Lodhi, A., Larson, N., Dhamdhere, A., Dovrolis, C., et al. Using PeeringDB to Understand the Peering Ecosystem. *ACM SIGCOMM Computer Communication Review 44*, 2 (2014), 20–27.

[172] Lougheed, K., and Rekhter, Y. Border Gateway Protocol (BGP). RFC 1105 (Experimental), June 1989. Obsoleted by RFC 1163.

[173] Lougheed, K., and Rekhter, Y. Border Gateway Protocol (BGP). RFC 1163 (Historic), June 1990. Obsoleted by RFC 1267.

[174] Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., et al. AS Relationships, Customer Cones, and Validation. In *Proceedings of the 2013 conference on Internet measurement conference* (2013), ACM, pp. 243–256.

[175] Lumezanu, C., Baden, R., Spring, N., and Bhattacharjee, B. Triangle Inequality and Routing Policy Violations in the Internet. In *Passive and Active Network Measurement*. Springer, 2009, pp. 45–54.

[176] Mahajan, R., Wetherall, D., and Anderson, T. Understanding BGP Misconfiguration. In *ACM SIGCOMM Computer Communication Review* (2002), vol. 32, ACM, pp. 3–16.

[177] Mahajan, R., Wetherall, D., and Anderson, T. Towards Coordinated Interdomain Traffic Engineering. In *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)* (2004).

[178] Mahajan, R., Wetherall, D., and Anderson, T. Negotiation-based Routing Between Neighboring ISPs. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2* (2005), USENIX Association, pp. 29–42.

[179] Mao, Z. M., Govindan, R., Varghese, G., and Katz, R. H. Route Flap Damping Exacerbates Internet Routing Convergence. In *ACM SIGCOMM Computer Communication Review* (2002), vol. 32, ACM, pp. 221–233.

[180] Marbach, P., Mihatsch, O., and Tsitsiklis, J. N. Call Admission Control and Routing in Integrated Services Networks Using Neuro-Dynamic Programming. *Selected Areas in Communications, IEEE Journal on 18*, 2 (2000), 197–208.

[181] Marescaux, J., and Rubino, F. Transcontinental Robot-Assisted Remote Telesurgery, Feasibility and Potential Applications. In *Teleophthalmology*. Springer, 2006, pp. 261–265.

[182] MAXMIND, Inc. GeoLite2 Free Downloadable Databases. http://dev.maxmind.com/geoip/geoip2/geolite2/. Dataset acquired on 2014-04-10.

[183] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review 38*, 2 (2008), 69–74.

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

[185] Mell, P., and Grance, T. The NIST Definition of Cloud Computing. *National Institute of Standards and Technology 53*, 6 (2009), 50.

[186] Monsanto, C., Foster, N., Harrison, R., and Walker, D. A Compiler and Run-time System for Network Programming Languages. *ACM SIGPLAN Notices 47*, 1 (2012), 217–230.

[187] Nascimento, M. R., Rothenberg, C. E., Salvador, M. R., Corrêa, C. N., de Lucena, S. C., and Magalhães, M. F. Virtual Routers as a Service: the Routeflow Approach Leveraging Software-Defined Networks. In *Proceedings of the 6th International Conference on Future Internet Technologies* (2011), ACM, pp. 34–37.

[188] Netflix Help Center. Internet Connection Speed Recommendations. https://help.netflix.com/en/node/306.

[189] Netvolution Project. Evolving Internet Routing: A Paradigm Shift to Foster Innovation. http://www.netvolution.eu/.

[190] NetworkX Developers. NetworkX Ford-Fulkerson. http://networkx.lanl.gov/reference/generated/networkx.algorithms.flow.ford_fulkerson.html.

[191] NetworkX Developers. NetworkX Graph Generators. http://networkx.lanl.gov/reference/generators.html.

[192] Norton, W. B. Internet Transit Prices - Historical and Projected. http://drpeering.net/white-papers/Internet-Transit-Pricing-Historical-And-Projected.php. Dr-Peering Blog.

[193] Nunes, B., Mendonca, M., Nguyen, X., Obraczka, K., and Turletti, T. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks.

[194] Oberortner, E., Sobernig, S., Zdun, U., and Dustdar, S. Monitoring Performance-Related QoS Properties in Service-Oriented Systems: a Pattern-Based Architectural Decision Model. In *Proceedings of the 16th European Conference on Pattern Languages of Programs* (2012), ACM, p. 13.

[195] OFELIA Consortium. OFELIA Control Framework (OCF) GitHub Repository. https://github.com/fp7-ofelia/ocf.

[196] Oliveira, R., Zhang, B., Pei, D., and Zhang, L. Quantifying Path Exploration in the Internet. *Networking, IEEE/ACM Transactions on 17*, 2 (2009), 445–458.

[197] Orda, A. Routing with End-to-End QoS Guarantees in Broadband Networks. *IEEE/ACM Transactions on Networking (TON) 7*, 3 (1999), 365–374.

[198] PCH. Packet Clearing House. https://www.pch.net/resources/data.php.

[199] PeeringDB. PeeringDB. https://www.peeringdb.com/.

[200] Peeters, R. The Maximum Edge Biclique Problem is NP-complete. *Discrete Applied Mathematics 131*, 3 (2003), 651–654.

[201] Pei, D., Azuma, M., Massey, D., and Zhang, L. BGP-RCN: Improving BGP Convergence Through Root Cause Notification. *Computer Networks 48*, 2 (2005), 175–194.

[202] Pei, D., Zhao, X., Massey, D., and Zhang, L. A Study of BGP Path Vector Route Looping Behavior. In *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on* (2004), IEEE, pp. 720–729.

[203] Phemius, K., Bouet, M., and Leguay, J. Disco: Distributed Multi-Domain SDN Controllers. In *Network Operations and Management Symposium (NOMS), 2014 IEEE* (2014), IEEE, pp. 1–4.

[204] Prasad, R., Dovrolis, C., Murray, M., and Claffy, K. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *Network, IEEE 17*, 6 (2003), 27–35.

[205] Prince, M. The DDoS That Knocked Spamhaus Offline (And How We Mitigated It). (Mar. 20, 2013, 06:26 PM), CloudFlare Blog, 2013.

[206] Psounis, K. Active Networks: Applications, Security, Safety, and Architectures. *Communications Surveys & Tutorials, IEEE 2*, 1 (1999), 2–16.

[207] Qazi, Z. A., Tu, C.-C., Chiang, L., Miao, R., Sekar, V., and Yu, M. SIMPLE-fying Middlebox Policy Enforcement Using SDN. In *ACM SIGCOMM Computer Communication Review* (2013), vol. 43, ACM, pp. 27–38.

[208] Quoitin, B., Pelsser, C., Swinnen, L., Bonaventure, O., and Uhlig, S. Interdomain Traffic Engineering with BGP. *Communications Magazine, IEEE 41*, 5 (2003), 122–128.

[209] Raghavan, B., Casado, M., Koponen, T., Ratnasamy, S., Ghodsi, A., and Shenker, S. Software-defined Internet Architecture: Decoupling Architecture from Infrastructure. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks* (2012), ACM, pp. 43–48.

[210] Reich, J., Monsanto, C., Foster, N., Rexford, J., and Walker, D. Modular SDN Programming with Pyretic. *Communications Magazine, IEEE 38*, 5 (2013), 128–134.

[211] Reitblatt, M., Foster, N., Rexford, J., and Walker, D. Consistent Updates for Software-Defined Networks: Change You Can Believe In! In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* (2011), ACM, p. 7.

[212] Rekhter, Y., Li, T., and Hares, S. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), Jan. 2006. Updated by RFCs 6286, 6608, 6793.

[213] Rexford, J. Rethinking Internet Routing. In *STOC* (2008), pp. 55–56.

[214] Richter, P., Smaragdakis, G., Feldmann, A., Chatzis, N., Boettger, J., and Willinger, W. Peering at Peerings: On the Role of IXP Route Servers. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (2014), ACM, pp. 31–44.

[215] Rothenberg, C. E., Nascimento, M. R., Salvador, M. R., Corrêa, C. N. A., Cunha de Lucena, S., and Raszuk, R. Revisiting Routing Control Platforms with the Eyes and Muscles of Software-Defined Networking. In *Proceedings of the first workshop on Hot topics in software defined networks* (2012), ACM, pp. 13–18.

[216] Roy, A., Yocum, K., and Snoeren, A. C. Challenges in the Emulation of Large Scale Software Defined Networks. In *Proceedings of the 4th Asia-Pacific Workshop on Systems* (2013), ACM, p. 10.

[217] Schaffrath, G., Werle, C., Papadimitriou, P., Feldmann, A., Bless, R., Greenhalgh, A., Wundsam, A., Kind, M., Maennel, O., and Mathy, L. Network Virtualization Architecture: Proposal and Initial Prototype. In *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures* (2009), ACM, pp. 63–72.

[218] Schapira, M., Zhu, Y., and Rexford, J. Putting BGP on the Right Path: a Case for Next-hop Routing. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks* (2010), ACM, p. 3.

[219] Schlinker, B., Zarifis, K., Cunha, I., Feamster, N., and Katz-Bassett, E. PEERING: An AS for Us. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks* (2014), ACM, p. 18.

[220] Schlinker, B., Zarifis, K., Cunha, I., Feamster, N., Katz-Bassett, E., and Yu, M. Try Before you Buy: SDN Emulation with (Real) Interdomain Routing. *Proc. of ONS* (2014).

[221] Scott-Hayward, S., O'Callaghan, G., and Sezer, S. SDN Security: a Survey. In *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for* (2013), IEEE, pp. 1–7.

[222] Shenker, S. The Future of Networking, and the Past of Protocols. https://www.youtube.com/watch?v=YHeyuD89n1Y.

[223] Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., and Sekar, V. Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service. *ACM SIGCOMM Computer Communication Review 42*, 4 (2012), 13–24.

[224] Sherwood, R., Gibb, G., Yap, K.-K., Appenzeller, G., Casado, M., McKeown, N., and Parulkar, G. Flowvisor: a Network Virtualization Layer. *OpenFlow Switch Consortium, Tech. Rep* (2009).

[225] Snijders, J. PeeringDB Accuracy: Is Blind Faith Reasonable? NANOG 58, 2013.

[226] Sobrinho, J. L., and Quelhas, T. A Theory for the Connectivity Discovered by Routing Protocols. *Networking, IEEE/ACM Transactions on 20*, 3 (2012), 677–689.

[227] Soulé, R., Basu, S., Kleinberg, R., Sirer, E. G., and Foster, N. Managing the Network with Merlin. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks* (2013), ACM, p. 24.

[228] Sprint. SLA Performance for Global MPLS. https://www.sprint.net/sla_performance.php?network=gmpls.

[229] Sprintson, A., Yannuzzi, M., Orda, A., and Masip-Bruin, X. Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* (2007), IEEE, pp. 1820–1828.

[230] Steenstrup, M. An Architecture for Inter-Domain Policy Routing. RFC 1478 (Historic), June 1993.

[231] Studer, A., and Perrig, A. The Coremelt Attack. In *Computer Security–ESORICS 2009*. Springer, 2009, pp. 37–52.

[232] Subramanian, L., Caesar, M., Ee, C. T., Handley, M., Mao, M., Shenker, S., and Stoica, I. HLP: a Next Generation Inter-Domain Routing Protocol. *Algorithms 100* (2005), 2–6.

[233] Suñé, M., Bergesio, L., Woesner, H., Rothe, T., Köpsel, A., Colle, D., Puype, B., Simeonidou, D., Nejabati, R., Channegowda, M., et al. Design and Implementation of the OFELIA FP7 Facility: the European OpenFlow Testbed. *Computer Networks 61* (2014), 132–150.

[234] Tariq, M. B., Motiwala, M., Feamster, N., and Ammar, M. Detecting Network Neutrality Violations with Causal Inference. In *Proceedings of the 2009 ACM CoNEXT Conference* (2009), pp. 289–300.

[235] Teixeira, R., Griffin, T. G., Resende, M. G., and Rexford, J. TIE breaking: Tunable Interdomain Egress Selection. In *Proceedings of the 2005 ACM conference on Emerging network experiment and technology* (2005), ACM, pp. 93–104.

[236] Teixeira, R., Marzullo, K., Savage, S., and Voelker, G. M. Characterizing and Measuring Path Diversity of Internet Topologies. In *ACM SIGMETRICS Performance Evaluation Review* (2003), vol. 31, ACM, pp. 304–305.

[237] Teixeira, R., Shaikh, A., Griffin, T., and Rexford, J. Dynamics of Hot-Potato Routing in IP Networks. *ACM SIGMETRICS Performance Evaluation Review 32*, 1 (2004), 307–319.

[238] Thai, P., and De Oliveira, J. C. Decoupling Policy from Routing with Software Defined Interdomain Management: Interdomain Routing for SDN-based Networks. In *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on* (2013), IEEE, pp. 1–6.

[239] Tootoonchian, A., and Ganjali, Y. HyperFlow: a Distributed Control Plane for OpenFlow. In *Proceedings of the 2010 internet network management conference on Research on enterprise networking* (2010), USENIX Association, pp. 3–3.

[240] Underwood, T. Wrestling With the Zombie: Sprint Depeers Cogent, Internet Partitioned. (Oct. 31, 2008, 10:07 AM), Renesys Blog, 2008.

[241] Valancius, V., Feamster, N., Johari, R., and Vazirani, V. Mint: a Market for Internet Transit. In *Proceedings of the 2008 ACM CoNEXT Conference* (2008), ACM, p. 70.

[242] Van der Merwe, J., Cepleanu, A., D'Souza, K., Freeman, B., Greenberg, A., Knight, D., McMillan, R., Moloney, D., Mulligan, J., Nguyen, H., et al. Dynamic Connectivity Management with an Intelligent Route Service Control Point. In *Proceedings of the 2006 SIGCOMM workshop on Internet network management* (2006), ACM, pp. 29–34.

[243] Vasseur, J., and Roux, J. L. Path Computation Element (PCE) Communication Protocol (PCEP). RFC 5440 (Proposed Standard), Mar. 2009.

[244] Villamizar, C., Chandra, R., and Govindan, R. BGP Route Flap Damping. RFC 2439 (Proposed Standard), Nov. 1998.

[245] Vissicchio, S., Vanbever, L., and Rexford, J. Sweet Little Lies: Fake Topologies for Flexible Routing. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks* (2014), p. 3.

[246] Viswanathan, R., Sabnani, K. K., Holt, R. J., and Netravali, A. N. Expected convergence properties of bgp. In *Network Protocols, 2005. ICNP 2005. 13th IEEE International Conference on* (2005), IEEE, pp. 13–pp.

[247] Wang, J., Zhou, M., and Li, Y. Survey on the End-to-End Internet Delay Measurements. In *High Speed Networks and Multimedia Communications*. Springer, 2004, pp. 155–166.

[248] Wang, P.-C. Scalable Packet Classification for Datacenter Networks. *Selected Areas in Communications, IEEE Journal on 32*, 1 (2014), 124–137.

[249] Wang, Y., Avramopoulos, I., and Rexford, J. Design for Configurability: Rethinking Interdomain Routing Policies from the Ground Up. *Selected Areas in Communications, IEEE Journal on 27*, 3 (2009), 336–348.

[250] Willinger, W., Alderson, D., and Doyle, J. C. Mathematics and the Internet: a Source of Enormous Confusion and Great Potential. *Notices of the American Mathematical Society 56*, 5 (2009).

[251] Willinger, W., and Roughan, M. Internet Topology Research Redux. *ACM SIGCOMM eBook: Recent Advances in Networking* (2013).

[252] Wu, J., Zhang, Y., Mao, Z. M., and Shin, K. G. Internet Routing Resilience to Failures: Analysis and Implications. In *Proceedings of the 2007 ACM CoNEXT conference* (2007), ACM, p. 25.

[253] Xiao, X. *Technical, Commercial and Regulatory Challenges of QoS: An Internet Service Model Perspective*. Morgan Kaufmann, 2008.

[254] Xiao, X., and Ni, L. M. Internet QoS: a Big Picture. *Network, IEEE 13*, 2 (1999), 8–18.

[255] Xiaohui, X., Ruxu, D., Lining, S., and Zhijiang, D. Internet Based Telesurgery with a Bone-Setting System. In *Integration Technology, 2007. ICIT'07. IEEE International Conference on* (2007), IEEE, pp. 218–222.

[256] Xu, D., Xiong, Y., Qiao, C., and Li, G. Failure Protection in Layered Networks with Shared Risk Link Groups. *Network, IEEE 18*, 3 (2004), 36–41.

[257] Xu, W., and Rexford, J. MIRO: Multi-path Interdomain Routing. In *Proceedings of ACM SIGCOMM* (2006), ACM, pp. 171–182.

[258] Yang, L., Dantu, R., Anderson, T., and Gopal, R. Forwarding and Control Element Separation (ForCES) Framework. RFC 3746 (Informational), Apr. 2004.

[259] Yang, X., Clark, D., and Berger, A. W. NIRA: a New Inter-Domain Routing Architecture. *Networking, IEEE/ACM Transactions on 15*, 4 (2007), 775–788.

[260] Yannuzzi, M., Masip-Bruin, X., and Bonaventure, O. Open Issues in Interdomain Routing: a Survey. *Network, IEEE 19*, 6 (2005), 49–56.

[261] Yannuzzi, M., Masip-Bruin, X., Sánchez, S., Domingo-Pascual, J., Oreda, A., and Sprintson, A. On the Challenges of Establishing Disjoint QoS IP/MPLS Paths Across Multiple Domains. *Communications Magazine, IEEE 44*, 12 (2006), 60–66.

[262] Zhang, X., Hsiao, H.-C., Hasker, G., Chan, H., Perrig, A., and Andersen, D. G. SCION: Scalability, Control, and Isolation on Next-Generation Networks. In *Security and Privacy (SP), 2011 IEEE Symposium on* (2011), IEEE, pp. 212–227.

[263] Zhang, Z.-L., Duan, Z., Gao, L., and Hou, Y. T. Decoupling QoS Control from Core Routers: a Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services. In *ACM SIGCOMM Computer Communication Review* (2000), vol. 30, ACM, pp. 71–83.

[264] Zhou, X., Wei, J., and Xu, C.-Z. Quality-of-Service Differentiation on the Internet: a Taxonomy. *Journal of Network and Computer Applications 30*, 1 (2007), 354–383.

[265] Zhu, D., Gritter, M., and Cheriton, D. R. Feedback Based Routing. *ACM SIGCOMM Computer Communication Review 33*, 1 (2003), 71–76.

[266] Zhuang, Y., Cappos, J., Rappaport, T. S., and McGeer, R. Future Internet Bandwidth Trends: An Investigation on Current and Future Disruptive Technologies. Tech. rep. tr-cse-2013-04, Polytechnic Institute of NYU, 2013.

Curriculum Vitae

Vasileios Kotronis was born on March 10th, 1989 and is a citizen of Greece. He is a Diplomate Electrical and Computer Engineer (Dipl.-Ing., ECE). His 5-year diploma, equivalent to an M.Sc. degree, was awarded from the National Technical University of Athens (NTUA) in 2011. He graduated with honors (GPA:9.62/10.0), third in his class (out of 300 students) and first in his undergraduate direction of specialization (Communications). During his undergraduate studies at NTUA, he received four prizes and two honorary scholarships for his academic performance. He did his thesis in the context of virtual network embedding and near-optimal heuristic resource allocation in collaboration with the NETwork Management & Optimal DEsign (NETMODE) Laboratory of the school of ECE in NTUA. After he completed his thesis, he started his PhD in the Communication Systems Group (CSG), led by Prof. Dr. Bernhard Plattner. He was a member of CSG between December 2011 and July 2015, working as a research/teaching assistant while doing his PhD-related research in the areas of Internet Routing, Software Defined Networking and IT Outsourcing. He successfully defended his thesis in August 2015.

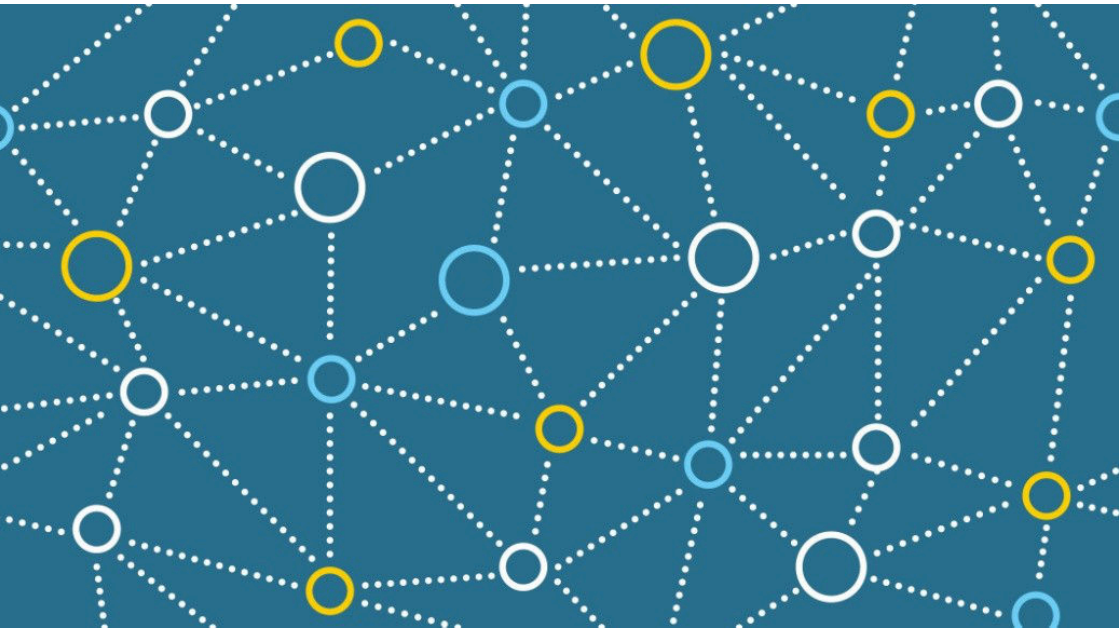
Kotronis was deeply involved in the project OFELIA, which was a collaborative project within the European Commission's FP7 ICT Work Programme (contract number: 258365) and stands for "OpenFlow in Europe: Linking Infrastructure and Applications". He was funded by the project between his arrival at CSG (December 2011) and the official end of the project (September 2013). He mainly worked on the OFELIA testbed infrastructure management, maintenance and administration (ETH Zurich testbed), extensions on the control framework software, experimental testing and verification, testbed federation, monitoring,

management and control tools. He also worked on the GpENI project, which stands for "Great Plains Environment for Network Innovation" between June 2012 and July 2015. He contributed to the testbed infrastructure maintenance and administration (ETH Zurich testbed), dealing with outages and connectivity issues.

Besides his work on OFELIA and GpENI, and his PhD-related research work on Internet Routing and Software Defined Networks, Kotronis was supervising 8 students (3 as a primary advisor and 5 as a co-advisor) that were doing their semester and master theses in CSG. He was a teaching assistant for the courses *Advanced Topics in Communication Networks: Software-Defined Networking* (primary assistant), *Computer Engineering 2* (co-assistant) and *Network Security* (co-assistant). He was also a reviewer for the IEEE ICC'13 TRICANS workshop and regularly for the IEEE Communication Letters.

Kotronis's technical skills and interests include: Internet Routing, Software Defined Networking (SDN), BGP, Network Management, Network Engineering, OpenFlow, SDN testbeds, Internet Peering, Network Security, Network Virtualization, Python, C, Java and GNU/Linux. He is proficient in English and has a fair knowledge of German, while he is a native Greek speaker.

The reader is referred to Kotronis's Linkedin profile for a complete view of his updated extended CV and related information. This can be found at the following URL:
https://ch.linkedin.com/pub/vasileios-kotronis/5b/69b/2b2.



Diss. ETH No. 22897
TIK-Schriftenreihe Nr. 158