Stitching Inter-Domain Paths over IXPs

Vasileios Kotronis[1] Rowan Klöti[1] Matthias Rost[2] , Panagiotis Georgopoulos[1]
Bernhard Ager[1] Stefan Schmid[3] Xenofontas Dimitropoulos[4,1]
[1]ETH Zurich, Switzerland [2]TU Berlin, Germany
[3]Aalborg University, Denmark [4]Foundation of Research and Technology Hellas (FORTH), Greece

ABSTRACT

Modern Internet applications, from HD video-conferencing to health monitoring and remote control of power-plants, pose stringent demands on network latency, bandwidth and availability. Centralized inter-domain routing brokers is an approach to support such applications and provide inter-domain guarantees, enabling new avenues for innovation. These entities centralize routing control for mission-critical traffic across domains, working in parallel to BGP. In this work, we propose using IXPs as natural points for stitching inter-domain paths under the control of inter-domain routing brokers. To evaluate the potential of this approach, we first map the global substrate of inter-IXP pathlets that IXP members could offer, based on measurements for 229 IXPs worldwide. We show that using IXPs as stitching points has two useful properties. Up to 91% of the total IPv4 address space can be served by such inter-domain routing brokers when working in concert with just a handful of large IXPs and their associated ISP members. Second, path diversity on the inter-IXP graph increases by up to *29* times, as compared to current BGP valley-free routing. To exploit the rich path diversity, we introduce algorithms that inter-domain routing brokers can use to embed paths, subject to bandwidth and latency constraints. We show that our algorithms scale to the sizes of the measured graphs and can serve diverse simulated path request mixes. Our work highlights a novel direction for SDN innovation across domains, based on logically centralized control and programmable IXP fabrics.

1. INTRODUCTION

A great success of the Internet is that it has been used in ways that were never anticipated during its early days. Carrying voice data[1] and connecting stock exchange markets are just two examples of such use cases. Nothing suggests that this innovation will not persist in the future. We see though that modern applications have increasingly tighter requirements for bandwidth, latency and/or availability [90]. For example, real-time HD video streaming, telemusic [32], remote control of critical infrastructure, such as power plants [20], or even telesurgery [51] are emerging or envisioned applications with strict network requirements. Presently, ISPs are able to provide certain QoS guarantees [80] only in intra-domain settings based on technologies such as leased circuits and VPN tunnels, e.g., over MPLS-TE. However, despite several research and standardization efforts, providing QoS guarantees at the inter-domain level has seen very limited success so far [16, 19, 85, 87]. Besides, current BGP routing can lead to inefficient paths across domains, triangle inequality violations, and long-lasting outages [9, 56, 66].

During the last decade, an increasing number of proposals coming from diverse angles advocate inter-domain routing brokers [34, 59, 63, 81, 82, 88] as an approach to enable ISPs to cooperate and provide end-to-end (e2e) guarantees. In these schemes, ISPs provide QoS-enabled pathlets [46], which are stitched together by an inter-domain routing mediator, e.g., a bandwidth broker [82]. Related initiatives are currently explored in the industry [3] and in standardization bodies, in particular in the context of the PCE (Path Computation Element) architecture [37, 55, 83].

This work visits logically centralized inter-domain mediators in light of the evolving Internet ecosystem. Namely, the Internet is becoming denser and more flat [30, 47, 62] because public Internet eXchange Points (IXPs) are continuously rising in number and size [5, 25]. In parallel, the paradigm shift towards network virtualization [73] and Software-Defined Networking (SDN) [70] introduces new possibilities in network management and innovation, also in the context of IXPs, e.g., as shown in the Software-Defined eXchange (SDX) approach [49]. While SDX enables new services at individual IXPs, we focus on multi-IXP services.

Contribution 1: Stitching inter-domain paths via IXPs. We propose using IXPs for stitching paths under the control of inter-domain routing brokers. We call such brokers *Control eXchange Points* (CXPs)[2]. The choice of IXPs as switching points exploits their rich connectivity, enabling high path diversity and global client reach with deployment in only a few well-connected IXPs. CXPs enable the utilization of additional path diversity compared to current BGP-based inter-domain paths, which typically follow valley-free routing policies [41, 44, 45]. CXP-stitched paths can freely cross multiple IXPs, yielding new paths that BGP hides.

Contribution 2: Mapping the IXP Multigraph. To evaluate the potential of CXPs, we map the global Internet substrate for pathlet stitching over IXPs. In particular, we outline a novel abstraction of the Internet topology, in which vertices are IXPs and edges are virtual links connecting two IXPs over an ISP. We call this abstraction the *IXP multigraph* because two IXPs can be generally connected with multiple edges over different ISPs. This abstraction hides the internal details of an ISP (including the technologies that can be leveraged to provide intra-domain QoS [88]), and serves a clean

[1] Increasingly, traditional telcos like Deutsche Telekom are planning to switch to IP telephony exclusively [2].

[2] CXPs can generally use any switching point between ISPs.

separation of concerns between intra- and inter-domain QoS routing that is consistent with the status quo. We analyze the member ISPs of 229 IXPs using data from Euro-IX [35] and show that CXPs can service, e.g., 40 % of the globally announced IPv4 addresses through only the 5 largest IXPs. This increases to 91 % if we also consider the 1-hop customers of the IXP members. Second, we show that by relaxing valley-free constraints, CXPs can greatly increase path diversity by up to 29 times compared to BGP valley-free routing.

Contribution 3: Algorithms. We present algorithms to efficiently exploit the high path diversity observed in the IXP multigraph. In particular, our algorithms aim at maximizing the number of concurrently embedded 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 variants should be leveraged to serve diverse requests.

CXPs provide a possible avenue for SDN innovation at the inter-domain level. In this context, we investigate both the algorithms that can serve as the controller logic of logically centralized inter-domain routing brokers, operating on IXP multigraphs, and the interesting properties of this particular data plane. The latter is studied both in space (incremental deployment at IXPs) and time, as the peering ecosystem evolves over the years. Moreover, we discuss further challenges for future work under the prism of a possible use case.

The rest of the paper is structured as follows. Section 2 provides the background on inter-domain service brokers and the motivation behind our IXP-based approach. Section 3 maps the global inter-IXP multigraph, based on Euro-IX and PeeringDB data, and characterizes its high path diversity and client reach for inter-domain QoS. Section 4 presents algorithms for embedding paths in IXP multigraphs and Section 5 evaluates these algorithms based on a custom simulator. Section 6 discusses our work under the prism of telesurgery as a use case, while Section 7 presents related literature.

2. SERVICE BROKERS, IXPS AND CXPS

This section first gives an overview of previous research on centralized path brokers for inter-domain guaranteed services. Second, we discuss why IXPs are suitable locations for deploying the data plane elements of path brokers. Lastly, we describe in detail the properties of our IXP-based path brokers, which we call Control Exchange Points (CXP).

2.1 Network Service Brokers

Previous research has focused on bandwidth brokers for mediating the concatenation of multiple guaranteed bandwidth pathlets (e.g., MINT [82]), or for scaling up the support for guaranteed bandwidth services within an ISP network (e.g., the work of Zhang et al. [88]). Similar initiatives have created bandwidth markets and commercial brokers, such as Geant's multi-domain Bandwidth-on-Demand service [3]. Other proposals introduce "route bazaars" between ISPs and end-users [34], where pricing mechanisms and interactions directly affect path establishment. Routing-as-a-Service controllers [63] have been proposed as potential broker implementations. Others have proposed entirely outsourcing routing control to inter-domain SDN controllers [59]. 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 operate on this global view. Centralized routing controller platforms based on the Path Computation Element (PCE) architecture [37, 83] have been evaluated in the context of QoS routing schemes for high capacity optical networks [43]. The initial multi-domain intention of PCE 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 [81], and are backed up by IETF standardization efforts [37, 83].

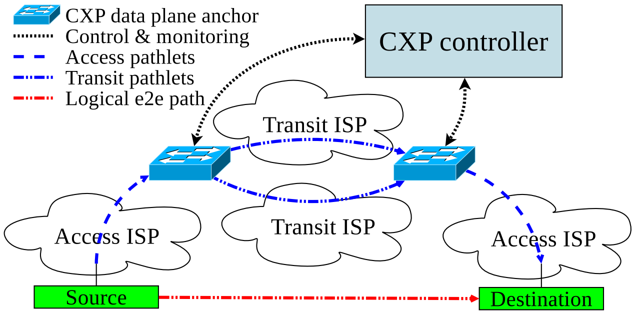


Figure 1: The CXP stitches QoS-enabled e2e paths.

2.2 Deploying Service Brokers on IXPs

Brokers and controllers for guaranteed e2e services need to exert inter-domain control through programmable data plane elements, such as OpenFlow switches. 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. IXPs are presently the hubs of multiple services surpassing their initial goal of pure layer-2 switching fabrics [25]: *(i)* hosting route servers for ease of BGP-based peering [78], *(ii)* mobile peering with 3G providers for traffic convergence [1], or *(iii)* the adoption of SDN approaches for new inter-domain applications [49]—such as application-specific peering—are just a few examples. They are therefore open to hosting new services for their members, together with increasing their peering base.

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 3 for details). We base our 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 Fig. 1).

2.3 CXPs

Following the observation that IXPs provide ideal locations for data plane anchors, we introduce *Control Exchange Points* (CXPs), i.e., control points which stitch pathlets across multiple administrative domains to construct global paths. Here we discuss in detail how CXPs would operate and the existing or emerging control and data plane technologies a CXP implementation could rely on. We note that the full implementation of a CXP is beyond the scope of this work.

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. A

CXP works in parallel to traditional routing and can control parts of traffic independently from BGP, e.g., utilizing flow space isolation mechanisms [73]. CXPs use data plane anchors which classify and switch traffic, such as SDN switches [70]. Software Defined Internet eXchanges (SDX) as proposed by Gupta et al. [49] could constitute an IXP-based deployment possibility. CXP control planes can be built using PCEs [37]. 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 and PCEs could enable CXPs to respond dynamically to changing requirements over a set of IXP-mediated inter-domain connections. Besides public IXPs, anchors can be deployed at private peering points for augmenting geographical coverage, if required. Between data plane anchors, traffic is shipped on virtual links which are parts of e2e paths and act as *pathlets* [46]. Pathlets are provided by ISPs and may be 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.

Providing Pathlets. 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 [70, 73]. Within the ISP backbone, QoS guarantees are provided via traffic engineering and prioritization techniques [14, 88]. MPLS-TE [53] is one example technology. The ISP is responsible for providing cross-traffic isolation internally, keeping its management policies confidential. The CXP on the other hand, provides 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) should therefore accompany the ISP-originated pathlet advertisements. One way to achieve this is by annotating pathlets with Shared Risk Link Group (SRLG) IDs [29]. The incentive for ISPs to provide pathlets is the revenue generated when their pathlets are used for e2e services; any ISP can be a provider. As shown in Fig. 1, the ISPs of the source and the destination offer access pathlets to connect to ISP-adjacent data plane anchors, while the intermediary ISPs offer transit pathlets over their domains, between anchors.

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 reembedding, 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 a pathlet to the correct data plane anchor. A CXP monitors the bandwidth, latency and availability of a path for the duration of the client's reservation, using existing technologies and approaches [72, 77, 84]. 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 "hot-standby" backup path carrying traffic duplicates. Besides, the CXP may choose to better utilize the available pathlets by re-embedding paths and defragmenting the substrate resources.

	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 1: Properties of the graphs generated from the Euro-IX dataset at various scale-down factors (SDF); larger SDFs correspond to smaller CXP penetration and vice versa.

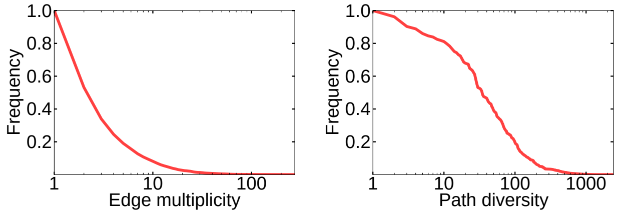
3. THE IXP MULTIGRAPH

In this section we measure and characterize the inter-IXP multigraph, i.e., the substrate on which inter-domain path brokers may operate. This analysis is necessary to understand where inter-domain control could be applied as well as the efficiency of incremental deployment, and is complementary to research related to scaling up CXP-like control planes [17] or investigating the trade-offs involved in logical centralization [64]. We thus 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, assuming that their member ISPs would offer the necessary pathlets, and *(ii)* how much path choice and diversity we can gain compared to classic BGP routing practices. We highlight this because currently, due to valley-free routing [41] and the prevalence of peer-to-peer links at IXPs [5], Internet paths normally cross at most one IXP. IXP-based path brokers simplify the use of paths that cross multiple IXPs.

3.1 Mapping the Inter-IXP Topology

We use four datasets to map the inter-IXP topology and the IPv4 address space: *(i)* the Euro-IX [35] and *(ii)* PeeringDB [75] databases, from which we obtained IXP membership data, *(iii)* the CAIDA AS relationship data [21, 65], and *(iv)* the CAIDA RouteViews AS-to-prefix data [22]. Due to space constraints we report results only for Euro-IX, which also provides geographic coordinates of IXPs (used to determine distances between IXP locations in Section 5) in contrast to PeeringDB. Analysis on PeeringDB data further corroborates our findings. Using a snapshot of the Euro-IX peering database [35], we extracted membership data for 6,542 ASes in 277 IXPs. After ignoring IXPs which 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. We derive simple graphs by collapsing multi-edges to single edges, annotated with the initial edge multiplicity.

We scale down the extracted 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 determine a suitable order based on a greedy heuristic, starting with the IXP having the largest address space coverage and in each iteration adding the IXP which yields the greatest number of non-overlapping addresses. 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. Fig. 1) , and *(ii)* connect the new IXP to other CXP-enabled IXPs at which these ISPs are present (via *transit* pathlets, cf. Fig. 1). We make this assumption, since our goal is to investigate the potential of an IXP-



(a)

(b)

Figure 2: CCDFs of edge multiplicity and path diversity

		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 2: AS-level policy models and their mean (μ) and median (M) path diversity, with added p2p links.

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. For such market analyses, investigating pathlet pricing and ISP participation, we refer the reader to works such as MINT [82] or RouteBazaar [34].

3.2 Properties of the Inter-IXP Multigraph

Table 1 gives an overview of the properties of the inter-IXP multigraph at different scales. The scale-down factor 32 corresponds to a small CXP deployment on 7 IXPs, while a factor of 1 involves all the 229 IXPs. We first observe average shortest path lengths between 1 and 1.9 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 in the initial topology with 7 IXPs. Fig. 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 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—over member ISPs—may be available.

Fig. 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. Also, different ISPs may share the same physical cables (e.g., transatlantic fiber links). As Fig. 2a and Fig. 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 fails, multiple indirect paths crossing other ISPs and IXP anchors may be used to replace the lost connectivity.

3.3 Reaching Clients with a Handful of IXPs

To be successful, reaching a large client base is important for a CXP. Therefore, we address the question of how much of the IPv4 address space can be reached from IXPs and their members. Fig. 3a depicts the IP address coverage versus the number of participating IXPs, assuming a greedy strategy maximizing IP address coverage. We show results both for directly adjacent IXP members as well as those connected over a single intermediate ISP (one hop). We observe that we can serve over 1 billion IP 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 IP 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 IP addresses (>50 % of announced 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.

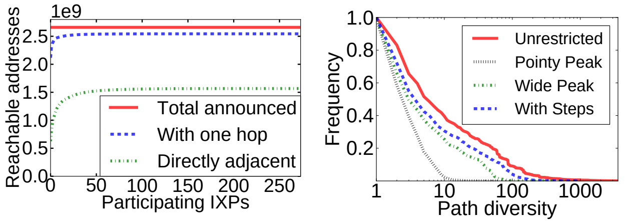
3.4 Rich Policy-Compliant Path Selection

We next evaluate the increase in path diversity gained when using a CXP-enabled IXP multigraph with relaxed peering policies as compared to valley-free routing of the AS-level topology. The most constrained policy corresponds to the traditional valley-free model [41] (scenario Pointy Peak); this allows the sequential composition of an uphill path (over customer-to-provider links), then at most one peer-to-peer (p2p) link, and a downhill path (over provider-to-customer links), resembling a mountain with a rather narrow peak. The upper bound on path diversity is achieved with the unrestricted policy scenario (scenario Unrestricted). 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, representing a scenario where there is exactly one CXP-mediated path traversed, passing over multiple IXPs. *(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.

To address the known deficiency in detecting p2p links using the current methodology to find AS-level links [5], 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. A given percentage of the derived links (cf. Table 2) is added to the graph, chosen uniformly at random; gradual addition is depicted with increasing percentages[3]. We estimate the corresponding policy-compliant AS-level path diversity, capitalizing on our prior work [57]. We use a sample size of 10K pairs of AS endpoints, selected randomly, with each AS weighted by the number of IPv4 addresses it announces over BGP.

Table 2 shows the mean and median path diversity observed for the various models and amounts of added p2p links, while Fig. 3b shows the distribution of path diversity for the models without added p2p 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

[3] Larger percentages were not investigated due to the memory limitations of the current NetworkX [71] min-cut implementations.



(a) Cumulative coverage of IPv4 address space originated by IXP members vs. participating IXPs

(b) CCDF of AS-level path diversity by scenario (with no added p2p links from IXP membership)

Figure 3: CXP potential via IXP deployment

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. Lastly, the small world properties of the Internet AS-level topology graph, also observed in the IXP multigraph abstraction (cf. Table 1), and our analysis of shortest path lengths 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 related reports [61].

3.5 Temporal Analysis of PeeringDB Graphs

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 CXP multigraph are valid *over time*. 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. 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.

We started with the evolution of the total number of the IXPs and ASNs which participate in the peering ecosystem, over time. We observed 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 witnessed an acceleration to a ∼115 IXPs/year rate of increase between the start of 2014 and the end of 2014. 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 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. 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, while the participating ASes have increased from ∼900 to ∼4000.

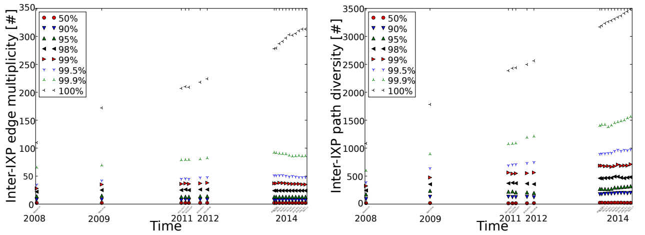
We next formed the actual corresponding IXP multigraph instances over time, and examined their sizes in terms of nodes and edges.We observed that the number of IXP nodes in the multigraph is increasing at a rate of ∼32 IXPs/year. We note here that this behavior is a bit different than the one that we observed for *all* the IXPs (nodes or not). 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. Examples of such IXPs are the ones in some remote parts of Africa, Australia, East Asia and South America. Larger ISPs that may peer concurrently at multiple IXPs around the world are usually not members of such small IXPs—at least in the beginning.

Moreover, we observed 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. By correlating this observation with the numbers of IXPs per ASN, 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. This is probably due to their more aggressive peering at geo-diverse public IXPs in the recent years. In total, the number of edges has evolved from ∼10k edges in 2008 to over 50k edges marking the start of 2015. Further correlation with the numbers of IXPs per ASN 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/year.

In Fig. 4a we examine the number of edges between *directly connected* IXP pairs. We 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 remaining 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.

In Fig. 4b, 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 Fig. 4a 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 summary, the IXP overlay graph is growing both in number of vertices and number of edges, thus improving connectivity. This is mainly due to the more aggressive peering of big players like



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

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

Figure 4: Edge multiplicity and edge-wise path diversity of the PeeringDB-based CXP multigraph over time.

Hurricane Electric, and the introduction of many new IXPs in remote parts of the globe during recent years. The edge multiplicity in the corresponding multigraph leads 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. In the presence of this densely connected "core", low path choice typically stems from badly connected stub ASes and not from a general graph property.

4. PATH STITCHING ALGORITHMS

As shown in Section 3 the IXP-based multigraph, on which CXPs 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, such as maximal latency or minimal bandwidth. These algorithms serve as the application logic of a logically centralized CXP controller, operating on the global view of the IXP multigraph for inter-domain path stitching.

The problem that we need to solve is complex for several reasons. *(i)* Requests from the large client base (cf. Section 3.3) dynamically arrive over time in a non-predictable manner, necessitating the use of online algorithms. *(ii)* While a single suitable e2e path can be found in polynomial time, the IXP-based graph offers rich choice (cf. Section 3.2, Section 3.4) and requires to carefully select which of the edges between two IXPs to use. *(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 edges, and preventing resource fragmentation. We formally introduce the e2e routing problem considered in this work as the QoS Multigraph Routing Problem (QMRP) in Section 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*, and subsequently one of them is *selected* for the actual embedding (cf. Section 4.2). Lastly, the framework is extended to support reconfigurations of pre-generated embeddings in order to accommodate further online requests.

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 $\mathsf{lat}_e \in \mathbb{R}_{\geq 0}$. 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. Fig. 1). 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 easier representation.

Integer Program 1: Optimal Flow Formulation (OptFlow)

$$\begin{array}{rlrr}
\max & \sum_{R \in \mathcal{R}} x_R & & \text{(OBJ)} \\
x_R = & \sum_{e \in \delta^+(\mathsf{s}_R)} P_R^e - \sum_{e \in \delta^-(\mathsf{s}_R)} P_R^e & \forall R \in \mathcal{R} & \text{(OF-1)} \\
0 = & \sum_{e \in \delta^+(v)} P_R^e - \sum_{e \in \delta^-(v)} P_R^e & \begin{array}{r}\forall R \in \mathcal{R}. \\ v \in V_G \setminus \{\mathsf{s}_R, \mathsf{t}_R\}\end{array} & \text{(OF-2)} \\
\mathsf{bw}_e \geq & \sum_{R \in \mathcal{R}} \mathsf{bw}_R \cdot P_R^e & \forall\, e \in E_G & \text{(OF-3)} \\
\mathsf{lat}_R \geq & \sum_{e \in E_G} \mathsf{lat}_e \cdot P_R^e & \forall\, R \in \mathcal{R} & \text{(OF-4)} \\
x_R \in & \{0, 1\} & \forall\, R \in \mathcal{R} & \text{(OF-5)} \\
P_R^e \in & \{0, 1\} & \forall\, R \in \mathcal{R}, e \in E_G & \text{(OF-6)}
\end{array}$$

We study how CXP operators can accept (and embed) as many requests as possible—a natural objective for any revenue-driven provider aiming at the 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, cf. *Integer Program 1* (OptFlow): the binary variable x_R decides whether request $R \in \mathcal{R}$ is embedded and the 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 and OF-2 induce a unit flow from s_R towards t_R if request $R \in \mathcal{R}$ is embedded (cf. [8]). $\delta^+(v)$ 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) of any pathlet is not exceeded. While the offline problem is interesting for optimizing existing allocations of requests in the background and further increase acceptance ratios (see Section 4.3), we are in general more interested in the online variant. In this context, each request R is known only at its arrival time, and the algorithm needs to compute an embedding (for the duration of the request) at that time.

4.2 Online 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 the set of all feasible paths. In the second stage one of these paths is selected for the embedding. We employ this approach as computing the *optimal* path under multiple objectives and constraints is generally NP-hard [42], while the algorithm might need to handle workloads of tens or hundreds of requests per second.

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 $\mathsf{s}_R$ to $\mathsf{t}_R$ or null
sample set of feasible paths $\mathcal{P}_R$
if $\mathcal{P}_R \neq \emptyset$ then
    select best path $P_R \in \mathcal{P}_R$ and embed $R$ accordingly
else
    return null
```

While investigating multiple path sampling strategies, we consider only a single selection strategy which aims at minimizing the utilization of the network (in order to provide room for many requests) and to secondarily penalize use of high-bandwidth and low-latency edges (since they are more scarce). We note that determining the best selection strategy is interesting in its own right, but lies outside the scope of this paper. 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 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.

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 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 requirement bw_R. Projecting the resulting multigraph onto a simple graph by replacing each set of edges with the minimal latency edge 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 this process would not find a feasible path even though such a path $P \in G$ exists. By replacing each edge of P with the minimal latency edge of the corresponding multi-edge 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 (cf. Korkmaz et al. [58]). In Section 4 of our accompanying technical report [60], three such algorithms are presented. The *Perturbed Dijkstra* (PD) algorithm is essentially a k-shortest paths variant [33], strictly minimizing 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. The run-time complexity of these algorithms is bounded by $\mathcal{O}\left(k \cdot (|E_G| + |V_G| \log |V_G|)\right)$, with k-many feasible paths to sample and $|V_G|$ nodes and $|E_G|$ edges to operate on. The algorithms can be used for different path sampling cases, ranging from purely deterministic variants (PD), to semi-randomized (GD) and fully randomized variants (GW).

Algorithm 2: 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 $\mathcal{P}_{\mathsf{confl}} \cup \{R^-\}$ by an offline algorithm
```

Integer Program 2: Heuristic Path Formulation (HeurPaths)

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

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

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

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

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

4.3 Adding 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 further. Such reconfigurations may be the only possibility to accept a request.

We thus extend the sample-selection scheme depicted as Algorithm 1 with the fallback scheme depicted as Algorithm 2. 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 and 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 set of *feasible* paths $\mathcal{P}_R$ of each request $R \in \mathcal{R}$ as input, computed previously via online path-sampling. 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^-, increasing the overall acceptance ratio.

We note that the proposed formulation is an adaption of the classic multi-dimensional knapsack problem [39] which, despite its NP-hardness, can be solved quite efficiently in practice using branch-and-bound solvers [50] when only dozens of paths are used for each request. In the evaluation (cf. Section 5), we use the *HeurPaths* program as follows: first we produce sets of paths for all requests (5

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 3: Online and offline/hybrid parameter space.

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.

5. ALGORITHMIC EVALUATION

We evaluate the performance of our algorithms in terms of Acceptance Ratio (AR), utilization i.e., the ratio of occupied bandwidth to the total available capacity[4], and computation time per request. To maximize the revenue, the acceptance ratio should be maximized while minimizing the resource utilization (so that there is room for more requests). Additionally, based on the ad-hoc online embedding of requests, the runtime should be low in order not to block the system. We use our custom CXP simulator [4], and the inter-IXP multigraphs described in Section 3. As discussed in Section 4, based on the multigraph nature of the IXP graph existing algorithms are not suitable and need to be adapted accordingly. In this section, we investigate the—empirical—trade-off between always choosing shortest paths (*Perturbed Dijkstra*) versus the more randomized versions *Guided Dijkstra* and *Guided Random Walk* (cf. [60]). Moreover, using the optimal Integer Program 1 as a baseline we show that our algorithms yield near optimal solutions quickly. Hence, our evaluation demonstrates how the orchestration on such graphs can be performed efficiently even on realistically sized scenarios. This is important for the application logic of potential SDN-based CXP implementations. We next elaborate on the setup and main insights yielded by the evaluation process.

5.1 Experimental Setup

The search space of our simulations is composed of the cross-product of the following parameter dimensions: *(i)* pathlet latencies and *(ii)* bandwidths, *(iii)* requested latencies and *(iv)* bandwidths, *(v)* graph sizes, *(vi)* maximal number of paths generated per request, *(vii)* number of requests per simulation run, and *(viii)* temporal characteristics of requests (e.g., durations). This search space has to be explored for each evaluated algorithmic variant. Due to its large volume, we constrain our search space so that the simulations may run within reasonable time frames (∼several weeks). Table 3 summarizes the used parameters. We next elaborate on the inter-IXP and endpoint-to-IXP pathlet latency and bandwidth model, the choice of the request endpoints and the temporal characteristics of the requests.

Latency. Pathlets connecting IXPs pass over ISP domains. To model pathlet latency in a geographically diverse ISP, we utilize the Hurricane Electric (HE) looking glass server [54] 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 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 $b = 26[\text{ms}]$. By least squares fitting, we model X as a normal distribution $N(\mu, \sigma)$ with $\mu = 0$ and $\sigma = 14[\text{ms}]$. We approximate the one-way latency as: $\mathtt{lat}_e = 1/2 \cdot rtt(d)$. and use this model for both access and transit pathlets (cf. Section 2). Request latencies are selected uniformly at random from four ranges (cf. Table 3) to evaluate looser to stricter requirements.

Bandwidth. We consider unitary requests, where each embedded request occupies the full bandwidth of the edge(s) it uses. This simplification removes the necessity to model offered and requested bandwidth and hence reduces the search space. In contrast, we rather focus on assessing our algorithms based on the topological characteristics of the inter-IXP substrate. In particular, we “fill” the multigraph with allocated bandwidth in order to discover its inherent potential for hosting arbitrary requests. Moreover, the chosen setup of uniform (and thus blocking) paths gives insights in how well the choice of shortest paths with respect to the hop count and the actual latencies are balanced to achieve the best resource utilization. If one were to always prefer shorter paths with respect to the hop count, the resource footprint would be minimized; however, this would reduce the availability of “mission-critical” pathlets of very low latency, hence reducing the acceptance ratio for latency-sensitive requests in the long run. Non-unitary request settings and corresponding simulations and effects are the subject of future work. For simplicity, non-access IXP-IXP pathlets are aligned with the unitary request bandwidth setting. In reality, their bandwidth is generally determined by ISP competition and auctioning [82].

Request Endpoints. We choose candidate IP addresses uniformly at random from the IPv4 address space adjacent to the members of the IXPs under examination. After we choose a source and destination address for a request, we retrieve their respective coordinates using the MaxMind GeoIP2 database [69]. These coordinates, together with the IXP locations, are used for geographical distance calculations between endpoints and IXPs. 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.

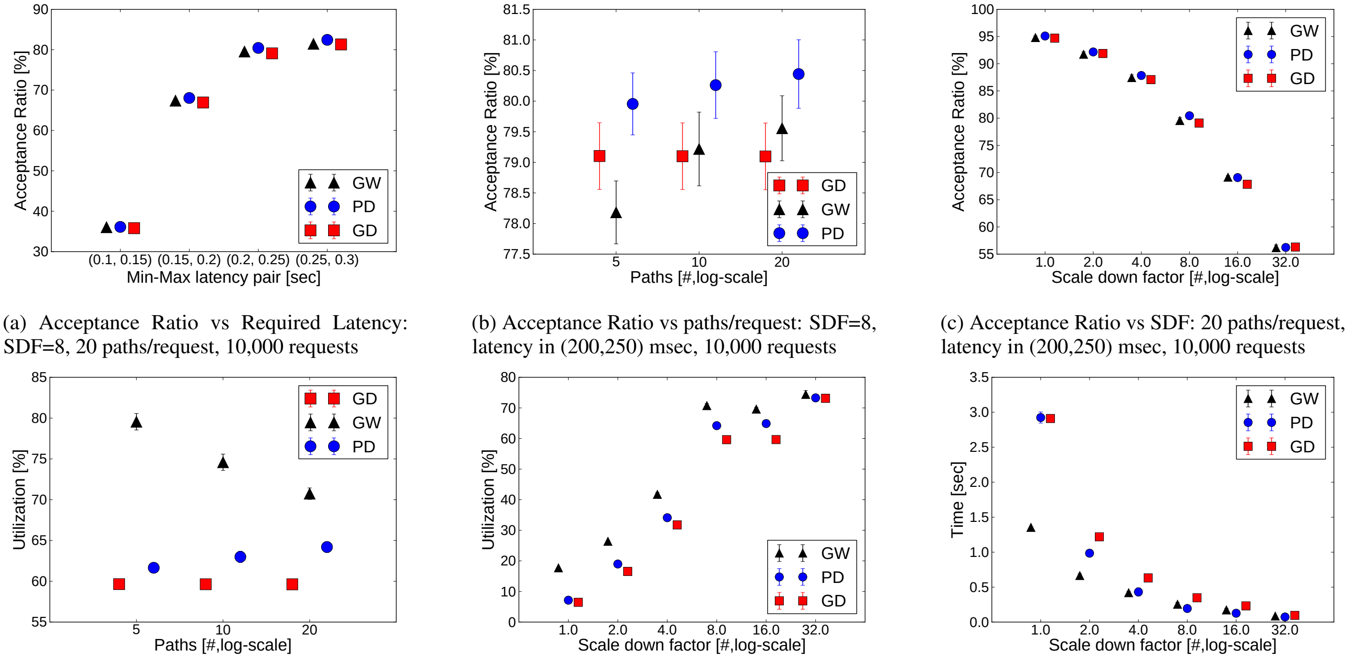
Online Requests. The requests arrive in order and are handled one-by-one in an online fashion. Each embedded request persists during the lifetime of the simulation (“infinite” duration), so that the peak load in the online case corresponds to the offline case, allowing for fair comparison at the corresponding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

5.2 Observations & Insights

Fig. 5 and Fig. 6 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. 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.We note that simple-graph approaches and baselines of previous work, not tailored to multigraph sampling (cf. Section 7), would have to operate on orders of magnitude larger substrates, e.g., using 2 “half-edges” and one AS node to simulate a pathlet, inducing biases. In such graphs $|V_G| = O(|E_G|)$, while here $|V_G| \ll |E_G|$.

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. Fig. 5a, Fig. 5b, Fig. 5c), 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. Fig. 5d), across scales (cf. Fig. 5e). The reason for the prevalence of Perturbed Dijkstra

[4] This metric takes into account only inter-IXP pathlets.



(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 5: Moderate scale online simulation

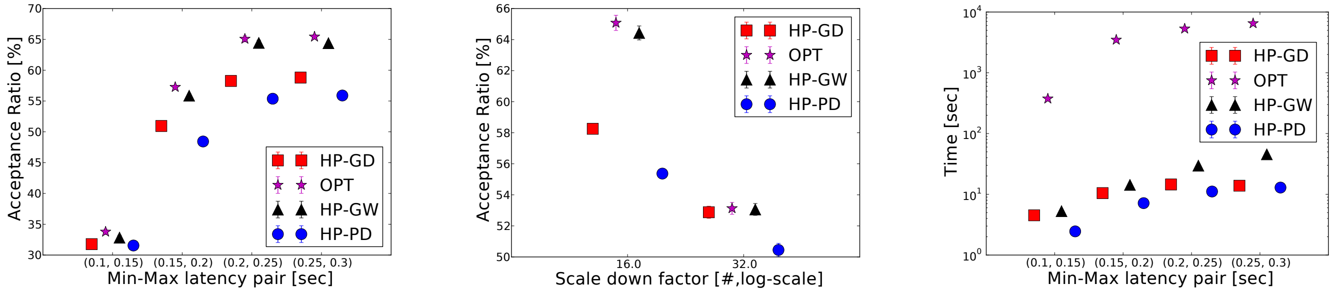
regarding acceptance ratios lies in its k-shortest path discovery; the edge-disjointness perturbation criterion, accompanied by the path selection function (cf. Section 4.2), counteracts its tendency to consume precious (latency-wise) paths and leads to good embeddings. Both Dijkstra approaches embed 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. In contrast, the Guided Walk, 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. Fig. 5b, Fig. 5d), since its progressive, random path sampling process benefits from exploring richer path sets (cf. Section 4.2).

How do hybrid variants behave w.r.t. acceptance ratios? HeurPaths with Guided Walk performs the best in terms of acceptance ratios and is very close to the offline optimal values. In contrast, HeurPaths with Perturbed or Guided Dijkstra leads to lower acceptance ratios as seen in Fig. 6a, with differences up to 10% for relaxed latency requirements. This is explained with the optimal latency seeking stages of these algorithms that do not couple 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. Fig. 6b) and latencies (cf. Fig. 6a).

How do offline, hybrid and online algorithms compare with each other w.r.t. 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 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. We did not include larger graph sizes for OptFlow due to run-time scaling issues, which 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. Fig. 5f). This is because the Guided Walk simply finds *feasible* paths quickly, without taking latency *optimality* into consideration and has lower computational complexity (cf. Section 4). In contrast, 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. Fig. 6c), and scales very poorly 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. Fig. 6a, Fig. 6b) at efficient run-times; the latter is evident in Fig. 6c, which presents the run-time of the Mixed Integer Programming computations versus the requested latencies. HeurPaths needs 10-100s to embed 10,000 paths. The path computations can be 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.

What is the effect of looser latency guarantees? The acceptance ratio (cf. Fig. 5a, Fig. 6a) 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,



(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: Small scale offline/hybrid simulation

then by 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 as we can observe in Fig. 6a.

What have we learned from online-offline cooperation? We have observed that using direct online-offline cooperation as described in Algorithm 2 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 utilization (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. In contrast, the pure online variants cannot see all the requests concurrently; therefore, they are doing their best to allocate each incoming request, or reject it when needed, without sacrificing utilization and jeopardizing future acceptance. We note 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 at large scales, when fast request allocation is desirable, especially if the incoming load of requests is high (e.g., due to higher CXP penetration). 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, which are close to its Dijkstra-based counterparts. Lastly, HeurPaths is a much better candidate for scaling up the hybrid version of the problem as opposed to OptFlow because it achieves similar acceptance ratios—in particular when combined with the Guided Walk—at much shorter run-times.

6. TELESURGERY AS A USE CASE

To get a better understanding of how CXPs can be used, beyond as plain multi-domain bandwidth brokers, we investigate the telesurgery [51, 68] use case. Telesurgery undoubtedly has stringent requirements on both availability and latency. Availability is essential for ensuring uninterrupted surgical operations and the patient's safety. Latency is important for making remote surgery feasible with real-time feedback [51]. In addition, the bandwidth requirements are generally high, e.g., for transmission of video streams [68].

Regarding availability, quick fail-over in case of emergencies is challenging [81, 87]. As a consequence, higher redundancy is needed *a priori* to achieve acceptable availability. One way to achieve this with CXPs, is 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 delivered to the application. A more efficient approach could be using Forward Error Correction (FEC) 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, the CXP immediately allocates a replacement, leaving the rest of the operational paths intact. Obviously, less reliable paths within ISPs mandate more redundancy to achieve high 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. Client flows can be dynamically assigned to (multiple) pathlets depending on the resources that are available within the "CXP cloud".

CXPs may also be able to find lower latency paths than traditional routing. If a path is subject to a triangle inequality [66] violation (the majority of paths are [9]) and there is a well-placed CXP anchor available to route over, the CXP can provide a path with lower latency. This implies the need for a broad CXP deployment footprint. While starting with selected IXPs as CXP anchors can serve as an initial step, it may not be sufficient for optimizing latency [6].

Finally, we note the following challenges related to SDN-based CXP implementations in the context of such use cases. *(i)* Controller distribution and placement; 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, while full distribution induces state consistency and concurrency challenges [64]. *(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 [72]. 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 rerouting. *(iii)* Path embeddings need to be protected against failures via CXP controller and anchor redundancy. These challenges are interesting directions for future SDN research at the inter-domain level, in the context of centralized pathlet stitching as a novel multi-IXP service.

7. RELATED WORK

Internet QoS and Our Work. Quality-of-Service is an evergreen topic that has been discussed for decades [13, 86, 89], together with the challenges associated with its implementation [16, 85, 87]. Such challenges have hindered its Internet-scale adoption in parallel with classic best-effort IP routing and peering agreements [19]. Our work is complementary to existing work on end-to-end Internet QoS, which covers the spectrum from low-level implementation (queueing mechanisms, QoS-oriented MPLS, OpenFlow mechanisms) to high-level policies (SLAs, traffic isolation). We propose an IXP-centric model that can be used to support the deployment of inter-domain QoS in the context of centralized pathlet brokers and resource controllers (cf. Section 2.1). CXPs could capitalize on prior work for the implementation [18, 27, 28, 53] and monitoring [72] of QoS-enabled pathlets; the scheme assumes a given per-ISP QoS and focuses on what can be done assuming that ISPs provide guaranteed pathlets anchored to IXPs, irrespective of *how* they are implemented. We note that this work, based on the concept of logically centralized brokers offering Routing as a Service [63], is an alternative to the proposal of source-routed, composable path segments advocated e.g., in ARROW [76]. We believe though that using IXPs as the primary points where the path composition takes place could be common ground for the deployment of such proposals. Moreover, the CXP business model, involving the mediation of contracts between end-clients and pathlet providers, could benefit from works that facilitate the formation, establishment, and verification of end-to-end connectivity agreements based on cryptocurreny systems [24].

IXPs. Recently, a number of studies analyzed the important role of IXPs [25] in terms of: *(i)* the flattening of the Internet topology [30, 47], *(ii)* the prevalence of IXP-based peering links in the Internet ecosystem [5, 12], and *(iii)* performance improvements, such as the reduction of average Internet delays and path lengths [7]. The potential rise of SDN within IXPs, e.g., enabled by Software Defined Internet eXchanges (SDX) [49], coupled with the changing role of IXPs, could turn to be an avenue for inter-domain QoS services based on the CXP paradigm. Moreover, Hu et al. [52] investigated how a version of *on-demand* peering policy relaxation can take place at IXPs in order to recover from route failures. Our more general approach (cf. Section 3) actively uses the path diversity induced from different variants of routing policies, based on sequential composition of *inter-IXP* pathlets. Finally, we refer the reader to the work of Castro et al. [23] on remote peering at IXPs. Among other things pertaining to their study, the authors discuss the marginal utility of reaching extra ISPs in terms of the potential for offloading transit traffic. In contrast, we are investigating the incremental deployment of IXP-based penetration in terms of: *(i)* end-client coverage, and *(ii)* path diversity potential for connecting these end-clients under certain quality guarantees. However, the proliferation of remote peering practices means that the IXP-based multigraph tends to get even richer, with remote ISPs being able to offer pathlets (using other layer-2 resellers) to more IXPs than the ones in their direct vicinity.

QoS Routing and Embeddings. Finding suitable paths between a pair of endpoints is a classic problem in computer science, and has been studied intensively in the context of online call control [67], virtual-circuit routing [11, 15] and also specifically QoS provisioning [42]. 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 [33, 42] and the look-ahead scheme employed by Korkmaz et al. [58]. In contrast to stochastic QoS routing algorithms as presented by Orda [74], we assume QoS guarantees over the provided ISP pathlets. 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.) [42]. The heuristic offline variant of our problem (embed as many e2e paths as possible), is a variant of *unsplittable* flow problems [31] and is related to the VPN [48] and virtual testbed mapping [26] problems. For a good survey, we refer the reader to Fischer et al. [38]. Schaffrath et al. [79] also present a relevant virtualization architecture. The *hybrid online-offline* approach that enables the reconfiguration of existing e2e embeddings, was shown to increase acceptance ratios in the domain of virtual network embeddings by Fan and Ammar [36]. Frikha and Lahoud have recently proposed to precompute QoS paths to improve performance [40]. In contrast, the paths that have already been computed in our work are reused at a later stage (possibly in different contexts), thereby not introducing any additional computational overhead. Lastly, Ascigil et al. [10] debunk the conventional wisdom that logically centralized computations do not scale in terms of domain-level end-to-end Internet routes.

8. CONCLUSION

We proposed using IXPs for stitching inter-domain paths under the control of centralized routing brokers, which provide paths with end-to-end guarantees for mission-critical applications. We considered a novel abstraction of the Internet topology: the IXP multigraph. Based on our study using extensive peering datasets, we evaluated the potential of IXP-based pathlet stitching 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 of the Internet IPv4 address space. *(ii)* In terms of AS-level path diversity, we showed the potential of generalized routing policies applied on the dense IXP multigraph. We observed an increase of at least one order of magnitude in path diversity, i.e., multiplicity of edge-disjoint paths, as compared to BGP inter-domain routing practices. *(iii)* We exhibited the importance of having suitable path sampling algorithms that take advantage of the richness of the multigraph. 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 multigraph view. Lastly, we placed our analysis within the scope of a demanding application, namely telesurgery, and highlighted open challenges.

As supported by this multi-faceted evaluation of the potential of CXPs, we believe that providing guaranteed inter-domain services is not anymore as intractable as it has been in the past. The flattening Internet topology and the emergence of SDN provide new avenues for innovation on CXP-like approaches. In our on-going work we investigate ways to kick-start CXP markets. In particular, our goal is to still provide better than best effort paths across the Internet, even when major IXPs or many ISPs do not participate yet in the market.

9. REFERENCES

[1] AMS-IX Mobile Peering. https://ams-ix.net/services-pricing/mobile-peering/grx.

[2] Business Practice Technology & Production driving IP transformation. https://www.telekom.com/careers/inside-telekom/inhouse-consulting/centers-structure/198718.

[3] GEANT Bandwidth-on-Demand Provisioning Tool. http://geant3.archive.geant.net/service/autobahn/pages/home.aspx.

[4] Software accompanying the paper. https://bitbucket.org/vkotronis/cxp_experimentation.

[5] Ager, B. et al. Anatomy of a Large European IXP. In *Proc. of ACM SIGCOMM* (2012).

[6] Ahmad, M., and Guha, R. A Tale of Nine Internet Exchange Points: Studying Path Latencies through Major Regional IXPs. In *Proc. of IEEE LCN* (2012).

[7] Ahmad, M. Z., and Guha, R. Studying the Effect of Internet eXchange Points on Internet Link Delays. In *Proc. of SpringSim* (2010).

[8] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., 1993.

[9] Anderson, T. Networking as a Service, HOTI-21 keynote. http://www.youtube.com/watch?v=bEtq_4arFz0, 2013.

[10] Ascigil, O., Calvert, K. L., and Griffioen, J. N. On the Scalability of Interdomain Path Computations. In *Proc. of IEEE IFIP Networking Conference* (2014).

[11] Aspnes, J. et al. On-line Load Balancing with Applications to Machine Scheduling and Virtual Circuit Routing. In *Proc. of ACM STOC* (1993).

[12] Augustin, B., Krishnamurthy, B., and Willinger, W. IXPs: Mapped? In *Proc. of ACM IMC* (2009).

[13] Aurrecoechea, C., Campbell, A. T., and Hauw, L. A Survey of QoS Architectures. *Multimedia Systems 6*, 3 (1998), 138–151.

[14] Awduche, D. et al. Overview and Principles of Internet Traffic Engineering. RFC 3272, Informational, 2002.

[15] Awerbuch, B. et al. Competitive Routing of Virtual Circuits with Unknown Duration. In *Proc. of ACM-SIAM SODA* (1994).

[16] Bell, G. Failure to Thrive: QoS and the Culture of Operational Networking. In *Proc. of ACM RIPQoS Workshop* (2003).

[17] Berde, P. et al. ONOS: Towards an Open, Distributed SDN OS. In *Proc. of ACM HotSDN Workshop* (2014).

[18] Bistarelli, S. et al. Unicast and Multicast QoS Routing with Soft-constraint Logic Programming. *ACM TOCL 12*, 1 (2010).

[19] Boucadair, M. et al. Considerations of Provider-to-Provider Agreements for Internet-Scale Quality of Service (QoS). RFC 5160, Informational, 2008.

[20] Buldyrev, S. et al. Catastrophic Cascade of Failures in Interdependent Networks. *Nature 464*, 7291 (2010), 1025–1028.

[21] CAIDA. AS Relationships Dataset (acquired on 2013-11-01). http://www.caida.org/data/as-relationships/.

[22] 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.

[23] Castro, I., et al. Remote Peering: More Peering without Internet Flattening. In *Proc. of ACM CoNEXT* (2014).

[24] Castro, I., et al. Route Bazaar: Automatic Interdomain Contract Negotiation. In *Proc. of HotOS XV* (2015).

[25] Chatzis, N. et al. There is More to IXPs than Meets the Eye. *ACM SIGCOMM CCR 43*, 5 (2013), 19–28.

[26] Chowdhury, M., Samuel, F., and Boutaba, R. PolyViNE: Policy-based Virtual Network Embedding Across Multiple Domains. In *Proc. of ACM VISA Workshop* (2010).

[27] Civanlar, S. et al. A QoS-enabled OpenFlow Environment for Scalable Video Streaming. In *Proc. of IEEE GC Workshops* (2010).

[28] Cruvinel, L., and Vazão, T. Improving Performance for Multimedia Traffic with Distributed Dynamic QoS Adaptation. *Comput. Commun. 34*, 10 (2011), 1222–1234.

[29] Dahai, X. et al. Failure Protection in Layered Networks with Shared Risk Link Groups. *IEEE Network 18*, 3 (2004), 36–41.

[30] Dhamdhere, A., and Dovrolis, C. The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh. In *Proc. of ACM CONEXT* (2010).

[31] Dinitz, Y., Garg, N., and Goemans, M. X. On the Single-Source Unsplittable Flow Problem. *Combinatorica 19*, 1 (1999), 17–41.

[32] 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*, vol. 7990. Springer, 2013.

[33] Eppstein, D. Finding the k Shortest Paths. *SIAM Journal on computing 28*, 2 (1998), 652–673.

[34] Esquivel, H. et al. RouteBazaar: An Economic Framework for Flexible Routing . Tech. rep. 1654, Univ. of Wisconsin - Madison, 2009.

[35] Euro-IX. European Internet Exchange Association. https://www.euro-ix.net/. Dataset acquired on 2014-04-09.

[36] Fan, J., and Ammar, M. H. Dynamic Topology Configuration in Service Overlay Networks: A Study of Reconfiguration Policies. In *Proc. of IEEE INFOCOM* (2006).

[37] Farrel, A. et al. A Path Computation Element (PCE)-Based Architecture. RFC 4655, Informational, 2006.

[38] Fischer, A. et al. Virtual Network Embedding: A Survey. *IEEE Communications Surveys Tutorials 15*, 4 (2013), 1888–1906.

[39] Fréville, A., and Hanafi, S. The Multidimensional 0-1 Knapsack Problem-Bounds and Computational Aspects. *Annals of Operations Research 139*, 1 (2005), 195–227.

[40] Frikha, A. and Lahoud, S. Performance Evaluation of Pre-computation Algorithms for Inter-domain QoS Routing. In *Proc. of ICT* (2011).

[41] Gao, L., and Rexford, J. Stable Internet Routing Without Global Coordination. In *Proc. of ACM SIGMETRICS* (2000).

[42] Garroppo, R. G., Giordano, S., and Tavanti, L. A Survey on Multi-constrained Optimal Path Computation: Exact and Approximate Algorithms. *Comput. Netw. 54*, 17 (2010), 3081–3107.

[43] Geleji, G. et al. A Performance Analysis of Inter-Domain QoS Routing Schemes Based on Path Computation Elements. In *Proc. of HONET* (2008).

[44] Gill, P., Schapira, M., and Goldberg, S. A Survey of Interdomain Routing Policies. *ACM SIGCOMM CCR 44*, 1 (2013), 28–34.

[45] Giotsas, V. et al. Inferring Complex AS Relationships. In *Proc. of ACM IMC* (2014).

[46] Godfrey, P. B. et al. Pathlet Routing. In *Proc. of ACM SIGCOMM* (2009).

[47] Gregori, E. et al. The Impact of IXPs on the AS-level Topology Structure of the Internet. *Comput. Commun. 34*, 1 (2011), 68–82.

[48] Gupta, A. et al. Provisioning a Virtual Private Network: A Network Design Problem for Multicommodity Flow. In *Proc. of ACM STOC* (2001).

[49] Gupta, A. et al. SDX: A Software Defined Internet Exchange. In *Proc. of ACM SIGCOMM* (2014).

[50] Gurobi Optimization. Gurobi Optimizer Reference Manual. http://www.gurobi.com, 2014.

[51] Haidegger, T., and Benyó, Z. *Extreme Telesurgery*. InTech, 2010.

[52] Hu, C. et al. A Measurement Study on Potential Inter-Domain Routing Diversity. *IEEE TSNM 9*, 3 (2012), 268–278.

[53] Hunt, R. Review: A Review of Quality of Service Mechanisms in IP-based Networks - Integrated and Differentiated Services, Multi-layer Switching, MPLS and Traffic Engineering. *Comput. Commun. 25*, 1 (2002), 100–108.

[54] Hurricane Electric Internet Services. Network Looking Glass. http://lg.he.net/. Dataset acquired on 2014-04-10.

[55] ICT-PACE. PACE: Next Steps in PAth Computation Element (PCE) Architectures. http://www.ict-pace.net/.

[56] Katz-Bassett, E., Madhyastha, H. V., John, J. P., Krishnamurthy, A., Wetherall, D., and Anderson, T. E. Studying black holes in the internet with hubble. In *NSDI* (2008), pp. 247–262.

[57] Klöti, R. et al. Policy-Compliant Path Diversity and Bisection

Bandwidth. In *Proc. of IEEE INFOCOM* (2015).

[58] KORKMAZ, T., AND KRUNZ, M. Multi-constrained Optimal Path Selection. In *Proc. of IEEE INFOCOM* (2001).

[59] KOTRONIS, V., DIMITROPOULOS, X., AND AGER, B. Outsourcing the Routing Control Logic: Better Internet Routing Based on SDN Principles. In *Proc. of ACM HotNets* (2012).

[60] KOTRONIS, V. ET AL. 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.

[61] KÜHNE, M. Update on AS Path Lengths Over Time. https://labs.ripe.net/Members/mirjam/update-on-as-path-lengths-over-time, 2012. RIPE NCC Article.

[62] LABOVITZ, C. ET AL. Internet inter-domain traffic. *ACM SIGCOMM CCR 41*, 4 (2011), 75–86.

[63] LAKSHMINARAYANAN, K., STOICA, I., AND SHENKER, S. Routing as a Service. Tech. rep. ucb-cs-04-1327, UC Berkeley, 2004.

[64] LEVIN, D. ET AL. Logically Centralized?: State Distribution Trade-offs in Software Defined Networks. In *Proc. of ACM HotSDN Workshop* (2012).

[65] LUCKIE, M. ET AL. AS Relationships, Customer Cones, and Validation. In *Proc. of ACM IMC* (2013).

[66] LUMEZANU, C. ET AL. Triangle Inequality and Routing Policy Violations in the Internet. In *Proc. of PAM* (2009).

[67] MARBACH, P., MIHATSCH, O., AND TSITSIKLIS, J. Call Admission Control and Routing in Integrated Services Networks Using Neuro-dynamic Programming. *IEEE JSAAC 18*, 2 (2000), 197–208.

[68] MARESCAUX, J. ET AL. Transcontinental Robot-assisted Remote Telesurgery: Feasibility and Potential Applications. *Annals of Surgery 235*, 4 (2002), 487.

[69] MAXMIND, INC. GeoLite2 Free Downloadable Databases. http://dev.maxmind.com/geoip/geoip2/geolite2/. Dataset acquired on 2014-04-10.

[70] MCKEOWN, N. ET AL. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM CCR 38*, 2 (2008), 69–74.

[71] NETWORKX. NetworkX. http://networkx.github.io/.

[72] OBERORTNER, E. ET AL. Monitoring Performance-related QoS Properties in Service-oriented Systems: A Pattern-based Architectural Decision Model. In *Proc. of EuroPLoP* (2012).

[73] ON.LAB. OpenVirtex: Programmable Virtual Networks. http://ovx.onlab.us/.

[74] ORDA, A. Routing with End-to-end QoS Guarantees in Broadband Networks. *IEEE/ACM TON 7*, 3 (1999), 365–374.

[75] PEERINGDB. PeeringDB. https://www.peeringdb.com/. Dataset acquired on 2014-01-20.

[76] PETER, S., ET AL. One Tunnel is (Often) Enough. In *Proc. of ACM SIGCOMM* (2014).

[77] PRASAD, R. ET AL. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network 17*, 6 (2003), 27–35.

[78] RICHTER, P. ET AL. Peering at Peerings: On the Role of IXP Route Servers. In *Proc. of ACM IMC* (2014).

[79] SCHAFFRATH, G. ET AL. Network Virtualization Architecture: Proposal and Initial Prototype. In *Proc. of ACM VISA Workshop* (2009).

[80] SPRINT. SLA Performance for Global MPLS. https://www.sprint.net/sla_performance.php?network=gmpls.

[81] SPRINTSON, A. ET AL. Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks. In *Proc. of IEEE INFOCOM* (2007).

[82] VALANCIUS, V. ET AL. MINT: A Market for INternet Transit. In *Proc. of ACM CONEXT* (2008).

[83] VASSEUR, J., AND LE ROUX, J. Path Computation Element (PCE) Communication Protocol (PCEP). RFC 5440, Standards Track, 2009.

[84] WANG, J., ZHOU, M., AND LI, Y. Survey on the End-to-End Internet Delay Measurements. In *High Speed Networks and Multimedia Communications*, vol. 3079. Springer, 2004, pp. 155–166.

[85] XIAO, X. *Technical, Commercial and Regulatory Challenges of QoS: An Internet Service Model Perspective*. Morgan Kaufmann, 2008.

[86] XIAO, X., AND NI, L. M. Internet QoS: A Big Picture. *Netwrk. Mag. of Global Internetwkg. 13*, 2 (1999), 8–18.

[87] YANUZZI, M. ET AL. On the Challenges of Establishing Disjoint QoS IP/MPLS Paths Across Multiple Domains. *IEEE Communications Magazine 44*, 12 (2006), 60–66.

[88] ZHANG, Z.-L. ET AL. Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services. *SIGCOMM CCR 30*, 4 (2000), 71–83.

[89] ZHOU, X., WEI, J., AND XU, C.-Z. Quality-of-service Differentiation on the Internet: A Taxonomy. *J. Netw. Comput. Appl. 30*, 1 (2007), 354–383.

[90] ZHUANG, Y. ET AL. Future Internet Bandwidth Trends: An Investigation on Current and Future Disruptive Technologies. Tech. rep. tr-cse-2013-04, Polytechnic Institute of NYU, 2013.