Moving Bits with a Fleet of Shared Virtual Routers

Pradeeban Kathiravelu*† Marco Chiesa‡ Pedro Marcos§ Marco Canini¶ Luís Veiga*
*INESC-ID Lisboa/Instituto Superior Técnico, Universidade de Lisboa †Université catholique de Louvain
‡KTH Royal Institute of Technology §UFRGS/FURG ¶KAUST

Abstract—**The steady decline of IP transit prices in the past two decades has helped fuel the growth of traffic demands in the Internet ecosystem. Despite the declining unit pricing, bandwidth costs remain significant due to ever-increasing scale and reach of the Internet, combined with the price disparity between the Internet's core hubs versus remote regions. In the meantime, cloud providers have been auctioning underutilized computing resources in their marketplace as spot instances for a much lower price, compared to their on-demand instances. This state of affairs has led the networking community to devote extensive efforts to cloud-assisted networks — the idea of offloading network functionality to cloud platforms, ultimately leading to more flexible and highly composable network service chains.**

We initiate a critical discussion on the economic and technological aspects of leveraging cloud-assisted networks for Internet-scale interconnections and data transfers. Namely, we investigate the prospect of constructing a large-scale virtualized network provider that does not own any fixed or dedicated resources and runs atop several spot instances. We construct a cloud-assisted overlay as a virtual network provider, by leveraging third-party cloud spot instances. We identify three use case scenarios where such approach will not only be economically and technologically viable but also provide performance benefits compared to current commercial offerings of connectivity and transit providers.

I. INTRODUCTION

The massive amount of content shared on the Internet, along with the bandwidth requirements to provide higher Quality of Experience (QoE) for latency-sensitive applications such as online gaming and video conferencing, has resulted in ever-increasing bandwidth demand and an urgent desire for flexibility in interconnections by network operators.

Cloud-assisted networks have been recently proposed, to increase performance, flexibility, and reliability of wide area networks [31]. They leverage cloud resources to compose large-scale overlay networks. This approach is appealing because cloud platforms generally guarantee high levels of availability through various levels of Service Level Agreements (SLAs) [19]. Moreover, major cloud providers such as Amazon have built their own global backbone network [40]. Therefore, they do not rely on the transit providers for their data transfer. These cloud networks are well provisioned and maintained, which makes them better than the Internet paths regarding loss rate and jitter [32]. Based on these observations, companies such as Teridion [42] and Cloudflare [23] offer cloud-assisted networks for SaaS providers as a premium service for a higher price compared to using standard Internet-based connectivity.

However, the use of cloud-assisted overlays in a broader set of use cases, such as their use by end users or enterprises for regular data transfers and the economic viability of such cases, is not well studied. We argue that the feasibility of cloud-assisted overlay solutions deserves a broader study informed by three key factors that we identify as follows.

ISBN 978-3-903176-08-9

Geographical price disparity. First, the oligopoly of a limited number of connectivity providers and a substantial dependence on expensive long-haul links to the US or EU for international connectivity, have caused a higher price for IP transit in the remote Internet regions [20]. For instance, prices for 10 Gbps Ethernet (10 GbE) bandwidth remain up to 20 times more expensive in São Paulo and Sydney, compared to EU and USA. This disparity is even more pronounced in remote regions such as Central Asia and Sub-Saharan Africa. For example, as of 2014, while the transit cost per Mbps per month was 0.94$ in the US [37], it was 15$ in Kazakhstan and 347$ in Uzbekistan [35]. Even though the average IP transit prices at major Internet hubs have fallen by an annual 61% during the past two decades, this decline has been much less pronounced elsewhere [37]. Consequently, the economic viability and incentives of a cloud-assisted solution significantly depend on its geographical location.

On-demand bandwidth. Second, even though transit providers currently offer bandwidth with minimum commitment, as low as 10 Mbps [24], interconnection contracts shorter than one month are still uncommon. To rectify this limitation, companies such as Epsilon [4] and PacketFabric [8] are working towards making bandwidth a tradeable utility, steering up dynamic interconnections among their users. Furthermore, companies like Megaport [6] and Console Connect [3] provide scalable point-to-point connectivity to cloud and network providers. Aligning to these recent developments, the pay-per-use and per-second billing of cloud providers can enable cloud-assisted solutions to offer dynamic interconnections with no commitment to time or usage.

Low-cost cloud resources. Third, *spot markets* of cloud providers such as Amazon Web Services (AWS) and Google Cloud Platform (GCP) offer their spare compute instances, with the same resources and capabilities as their on-demand counterparts, at a much lower price. Nevertheless, spot instances can be suddenly interrupted with a notification period of up to two minutes. Applications that can tolerate the volatile nature of the spot instances can use them as an economical alternative to the on-demand ones. These spot markets, typically underutilized, have idle and affordable resources in multiple regions. The economic viability of cloud-assisted networks depends on the need for connectivity services that are more dynamic than the traditional ones. Resilient architectures built atop spot instances can bring cloud expenditures down enough to make cloud-assisted overlays profitable.

Given the above premises, we set out to investigate the following research questions: i) Can a cloud-assisted network built on several spot instances be a viable solution to realize an on-demand virtual Network-as-a-Service provider, i.e., a network provider built over multiple cloud offerings and that does not own any fixed or dedicated resources? ii) If so, what are its possible usage scenarios and under what conditions would this approach be economically sustainable for a network provider? iii) When would this approach be cheaper than existing alternative connectivity providers, including transit providers, Internet Service Providers (ISPs), and Multiprotocol Label Switching (MPLS) network providers? iv) how stable are today's spot instances? v) If not competitive on price, would this approach be able to provide higher performance and/or additional features than the alternatives?

We study the technological and economic viability of such a cloud-assisted network and propose $NetUber$ as an efficient architecture to realize it. $NetUber$ consists of a broker that i) purchases spot VMs from the cloud providers, and ii) creates an overlay network over the multiple spot VMs to function as a large-scale inter-region connectivity provider to the customers who would buy connectivity directly from $NetUber$. Thus, $NetUber$ operates as an on-demand virtual connectivity provider running atop virtual routers in the spot VMs. By leveraging the memory and CPU of the acquired spot instances, we further envision a deployment of auxiliary services such as compression-as-a-service [1] and encryption-as-a-service [2], offering an optional compressed or encrypted data transfer between the regions.

Our primary contributions are: i) an economic model to exploit spot markets for direct secured connectivity between pairs of endpoints, and ii) an inter-cloud approach that leverages spot VMs in building a reliable virtual connectivity provider. When compared to traditional flat-price connectivity providers, our extensive evaluation shows that i) $NetUber$ best suits the needs of small dynamic monthly transfers up to at least 50 TB and ii) $NetUber$ cuts Internet latencies up to a factor of 30%. We see our contributions as a first step towards a more systematic understanding of the next-generation overlay interconnection networks.

We discuss the current potential for cloud-assisted networks in Section II. We then elaborate the potential deployments for the identified use cases of $NetUber$ in Section III. We illustrate the opportunities for $NetUber$ through an economic analysis in Section IV. We evaluate $NetUber$ against the current offerings in Section V. We discuss the related work in Section VI and conclude the paper in Section VII.

II. MAKING CLOUD-ASSISTED NETWORKS A REALITY: BACKGROUND AND MARKET ANALYSIS

A cloud-assisted network leverages the cloud VMs to host the virtual routers and the underlying cloud network infrastructure for its data transfer. In this section, we look at these counterparts from both economic and technological perspectives, and build a foundation for $NetUber$ design decisions based on our observations on cloud pricing trends.

A. Spot Markets

Large-scale dynamic overlays such as $NetUber$ require many cloud instances and a billing model that charges overlay operators for their actual usage of cloud resources. While the cloud providers offer per-second billing[1], the price of on-demand instances remains a concern for overlay operators. Despite a constant price reduction [18], building an overlay network with on-demand instances turns out to be more expensive than using traditional connectivity providers for the same Service Level Objectives (SLOs). Currently, largest cloud providers offer a cheaper and appealing alternative to on-demand instances. The so-called *spot instances* are spare computing resources, exactly identical to their on-demand counterparts, which are offered at a much lower price but can suddenly be interrupted by the cloud provider with a short notification. Amazon estimates that using spot instances can save up to 90% compared to the on-demand pricing [10]. Similarly, GCP preemptible instances [29] provide a flat rate of 80% of savings, and the recently announced Microsoft Azure low-priority instances [41] are expected to offer the same, compared to their respective on-demand offerings. Thus, we propose to use spot instances to minimize the cost as much as possible, while providing the same availability and performance guarantees.

In contrast, "reserved" instances refer to the cloud instances that are leased for an extended period such as 1 - 3 years, for a discounted price. Often reserved instances offer similar savings to spot instances. For example, 82% savings in Azure. However, they are unsuitable due to their time commitment that is unfit for the dynamic nature of the network demand.

Availability zones. Cloud data centers are present in several geographical locations that are known as the "cloud regions". Each region further consists of multiple "availability zones" that are physically separated servers in a given region. Availability zones provide resilience and fault-tolerance to the cloud regions. Each EC2 region has multiple availability zones that are isolated but connected to each other through low-latency links internal to the EC2 region network. Price fluctuations of EC2 spot market are inevitable and more vigorous for a few instance types in some availability zones at specific time frames. Even identical spot instances of different availability zones, inside the same region, often have different prices at times. In contrast, GCP preemptible instances have a fixed pricing unlike the dynamic pricing of AWS spot instances.

Price fluctuations. We monitored the availability, performance, and price fluctuations of EC2 spot instances over a time frame of three months (April - June 2017). Throughout our experiments, Linux r4.8xlarge spot instances remained the cheapest, yet memory-optimized EC2 instances with 10 GbE network interface. Each of these R4 instances consists of 32

---

[1] AWS and GCP moving to per-second billing from their per-hour and per-minute billings, starting from October 2017.

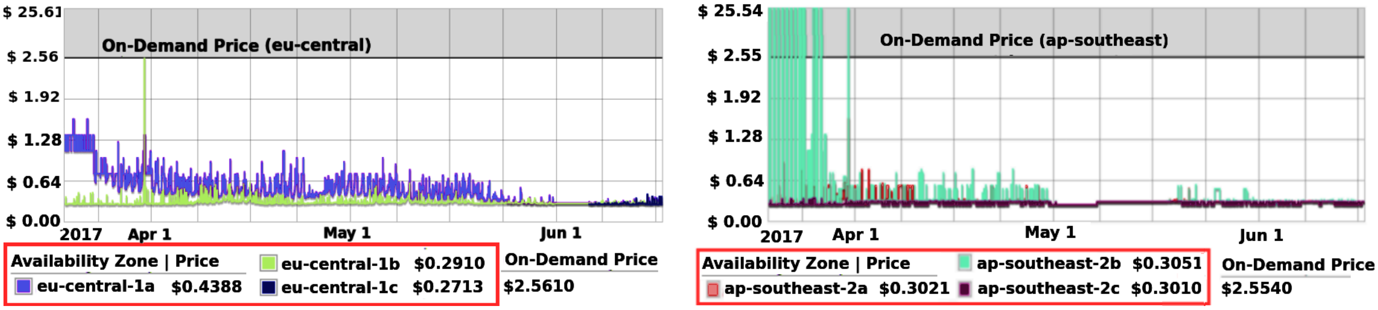

Fig. 1. AWS Linux r4.8xlarge Spot Instance Price in Frankfurt and Sydney

virtual CPUs and 244 GB of memory. They offer 10 Gbps bandwidth inside an AWS placement group (a logical grouping of EC2 instances inside an availability zone as a cluster). Network transfers outside a placement group are limited to 5 Gbps [13]. Figure 1 depicts the price fluctuations among the Linux r4.8xlarge instances of the availability zones of Frankfurt and Sydney regions during the period. We observed up to 89% of savings with spot fleets of r4.8xlarge instances in Sydney, Frankfurt, and North Virginia regions.

The spot price for Frankfurt remained relatively low and stable across all the availability zones. However, the spot price even exceeded the on-demand price in Sydney for availability zone 2b at times, while the other two availability zones remained cheaper for the spot instances. Instances of the zones eu-central-1c and ap-southeast-2c had a relatively steady and cheap price. While it is straightforward to opt for instances from the availability zones that have remained cheaper recently with a stable price, fluctuations in the future are unpredictable. Hence, while some spot instances belonging to a particular availability zone are being terminated, spot instances may be spawned in the other availability zones of the same region. This dynamic nature of the network poses the question of how the inter-region traffic should be re-adjusted to route to the current active instances.

Multiple spot instances can be spawned at once adhering to user specifications in target capacity and cost threshold, through approaches such as EC2 Spot Fleet [12] to mitigate the problem of scale and complexity in managing a large number of instances at once. Currently, EC2 spot instances are also available with predefined duration from one to six hours, 30 - 50% cheaper than the on-demand instances, making them less volatile and more reliable [10]. We observed that by leveraging multiple availability zones in each region, a stable overlay could be operated using the cheapest instances over time.

B. Cloud Data Transfer

Inter-region cloud data transfer prices remain relatively high as there is no “spot data transfer” that provides cheaper data transfer with a volatile bandwidth. The data transfer pricing exhibited a slower decline compared to that of IP transit. Around 20% and 25% of price reductions have been reported in 2010 [16] and 2014 [17] respectively, for data transfer out from the EC2 instances.

While it is typical for small enterprises and home users to connect to the cloud servers through the public Internet, for large-scale data transfers, a dedicated connection or co-locating with a cloud Point of Presence (PoP) is recommended for throughput and cost efficiency. Such a direct connection avoids depending on a third-party connectivity provider such as ISPs, which incur more costs and also limit the scale of data transfer (for example, typically ISPs offer up to 1 TB per month for home users abiding by the data rate promised in the data plan). Cloud data transfers through private direct connections thus avoid the bottleneck caused by the Internet-based connectivity between the user data centers and cloud servers. The virtual network overlay users must have an existing connection to the cloud provider or set it up directly with the cloud provider or its partners. The costs of setting up the Direct Connect is pay-per-use and not more expensive than the alternatives with the same throughput. It is configured and paid directly by each cloud user to the cloud provider.

Currently, cloud providers such as AWS and GCP do not charge for the data transfer into an instance either from the Internet or the other regions. They charge for data transfer out of a region, which differs based on the destination: the Internet, a server connected by Direct Connect, and to any other region. Currently, AWS typically charges the inter-region traffic independent of the destination region (with a few exceptions for nearest regions such as cheaper transfer between the US East regions - North Virginia and Ohio). On the other hand, GCP clusters the regions into four groups (worldwide, Asia, China, and Australia) to give differentiated rpricing based on the group of egress/destination region.

Cloud region price disparity. Figure 2 illustrates the disparity of pricing among the AWS regions to transfer a unit of data (1 TB), for transfers up to 10 TB. Larger transfers become cheaper per unit, with the price reaching almost half when the total volume reaches 500 TB. For example, it decreased from 92.16$/TB to 51.20$/TB for transfers out from regions 1 - 8 (Canada and EU-based regions, and US-based regions except for GovCloud) to the Internet. The cloud data transfer options such as Direct Connect are cheaper than sending data from the cloud to a customer server directly through the Internet. The discrepancy in pricing among the regions is visible (which is comparable to the IP transit price disparity); regions 1 - 9 (US, Canada, and EU) remain much cheaper than the others.

We observe that the current pricing of data transfer for the cloud providers is not supportive of a more comprehensive adoption of an Internet-scale cloud-assisted network, due to the lack of a differentiated pricing model based on the current lo-

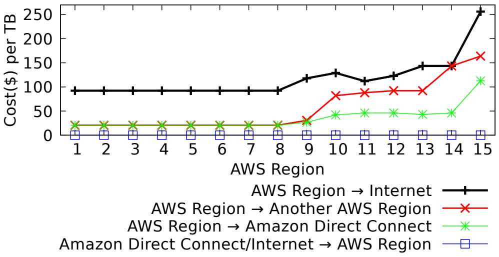

Fig. 2. Data Transfer Cost for AWS

cal time or demand for bandwidth. Moreover, cloud providers charge for the data transfers by the volume of data transferred, rather than by the data rate, unlike transit providers or ISPs. Regardless of the throughput, the cloud user pays the same amount based on the amount of data transferred. Therefore, there is no incentive for the cloud users to opt for a slower data rate even when their application is delay-tolerant. Furthermore, not considering the cloud overlay network scenario, cloud providers discourage long-distance inter-region data transfer to counter communication delays due to poor SaaS design.

Based on our observations and subsequent analysis on the cloud data transfer offerings and the availability of cheaper spot VMs, we deduce that the data transfer will contribute with the most significant share to the expenses for the Internet-scale overlays deployed over multiple regions. Thus data compression and minimizing data transfer path lengths can enable a cost-efficient execution of the cloud-assisted network.

III. TOWARDS NETUBER DEPLOYMENTS

In this section, we will look into the deployment architecture of three primary use cases of $NetUber$: i) a cheaper point-to-point connectivity between two regions for data transfers, ii) a premium connectivity between multi-cloud regions for end users for faster data transfer and better SaaS performance, and iii) a connectivity provider with additional network services.

A. Economical Point-to-Point Connectivity

$NetUber$ leverages spot instances to offer short-term or small-scale direct access between two geographically separated endpoints, as an economical alternative to enterprise MPLS networks. $NetUber$ has no dedicated servers. Its broker is hosted in a set of spot instances per region. Cloud monitors such as AWS alarms are leveraged to ensure that each region has at least one broker instance that is active and not scheduled for termination. For a stable overlay, we need some instances in each region, based on the bandwidth demand and the number of active instances at any given time. At any moment, the broker purchases instances from the availability zone that has the cheapest of the higher performance instances in a region. Over time, instances are spawned and maintained across multiple availability zones. Therefore, the expensive ones to maintain are terminated at the earliest.

Each $NetUber$ cloud instance hosts a virtual router. Each virtual router can dynamically connect to the virtual routers of certain spot instances of another region through the overlay, based on the users' connectivity or data transfer requests. Consider a scenario where a customer chooses to send data from her server s_o to the destination server s_d. These servers are in the cloud provider regions r_o and r_d respectively and are connected to the cloud provider through a dedicated connection such as Amazon Direct Connect. Figure 3 illustrates a representation of a sample $NetUber$ deployment that offers a direct connection between s_o and s_d. $NetUber$ is composed of many spot VMs in multiple regions, connected through the overlay network of virtual routers.

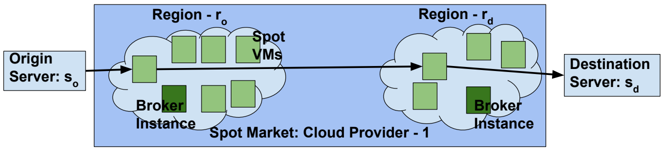

Fig. 3. $NetUber$ Deployment with a Single Cloud Provider

The broker instances monitor the resource utilization of the current VMs purchased in the spot market. They alter the spot fleet policies to bid for more instances, when the existing VMs are not sufficient (measured with a margin, to avoid performance degradation) to address the demand for connectivity and when the price for the spot instances are profitable to $NetUber$. The technical challenges include, i) initializing a newly spawned instance to operate as a virtual router in a short time, and ii) ensuring that the instances can be connected and identified through an overlay, other than their physical address, as spot instances remain volatile. The list of spot instances of the virtual routers in each region can be provided through an accessible and reliable location in the cloud provider such as an S3 bucket. The broker handles the updates to the list of instances, consisting of new and terminated spot instances.

B. Higher Performance Point-to-Point Interconnection

On the intra-domain traffic, an ISP can seek the shortest path as it controls the network. Since Border Gateway Protocol (BGP) decisions are mainly policy-oriented, it may not result in the selection of the best or the shortest path. With the cloud instances, $NetUber$ can choose to intelligently route the traffic towards the VMs in the exact regions, minimizing the number of hops and path length.

Currently, a few cities and geographical regions host the cloud regions for multiple providers. For example, North Virginia, Mumbai, London, São Paulo, Tokyo, and Singapore host both AWS [11] and GCP [28] regions. As of now, Ohio, North Carolina, Seoul, Canada central, and Ireland are AWS regions but not GCP regions; Iowa, Belgium, South Carolina, and Taiwan are GCP regions that are not AWS regions.

Figure 4 elaborates this scenario with the cloud provider 1 having presence in regions r_o and r_i, and the provider 2 with presence in regions r_i and r_d. None of the providers are present in both r_o and r_d, while both providers are present in r_i. $NetUber$ functions as a mediator between the two cloud providers to enable data transfer from $r_o \rightarrow r_d$ by interconnecting between the cloud providers in r_i, through the Internet-based connectivity or a direct/dedicated connectivity offered by the cloud providers between the clouds.

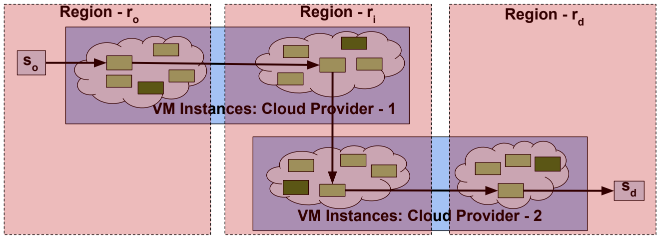

Fig. 4. Deployment Across Multiple Cloud Providers

The latency of the $NetUber$ inter-cloud interconnection remains minimal as multi-clouds of the same region are proximate to each other: they may share the same co-location facilities, or potentially be interconnected via a direct connection between the servers of the cloud vendors. For example, AWS Direct Connect can offer a direct interconnection between a pair of AWS and GCP instances in r_i. This architecture provides a higher performance point-to-point connectivity for the end users for data transfers to a geographically remote region, instead of connecting directly through an ISP.

$NetUber$ needs to consider operational differences between the cloud vendors for a stable execution. For example, currently, an AWS instance will be terminated by AWS when the current spot price exceeds the bid, with a 2-minute notice. GCP offers a 30-second notification. An AWS spot instance is terminated either by the user or by AWS when the current spot price exceeds the user bid price or when the spot resource pool in an availability zone is over-utilized. GCP terminates every spot instances 24 hours after they were started, in addition to the same conditions as AWS spot instance termination. $NetUber$ avoids shutting down instances on its own for the sake of stability, except for terminating the instances after supporting a significant spike in bandwidth demand.

A SaaS provider can use $NetUber$, instead of having geo-replicated deployments in multiple cloud regions which can be technically more challenging and more expensive. Furthermore, $NetUber$ supports more regions beyond those supported by any single cloud provider. For example, SaaS applications hosted in r_d can be accessed by an end user in r_o through $NetUber$ more reliably than through the public Internet. Thus, a SaaS provider can exploit $NetUber$ to create a point of presence in multiple regions while hosting the application in just a single region. Hence, $NetUber$ can be a potential cost-efficient alternative to geo-replicated solutions.

C. A Provider of Network Services

Virtual Network Functions (VNFs) such as packet scrubbers, transcoder, firewalls, load balancers, and proxies, can be hosted in the spot VMs of $NetUber$ as SaaS to perform middlebox actions to alter the data flow transferred atop the overlay. For example, forwarded data can be encrypted or compressed at an instance before the inter-region transfer, if prompted by the user, as additional services. Encryption enhances the security of data transferred, while compression allows an economic transfer, with minimal latency as data can be compressed in-memory in the spot instances. We can host caching services in $NetUber$ instances to optimize or limit WAN traffic. $NetUber$ can also be used for content distribution atop the overlay or mitigation of distributed denial of service (DDoS) attacks on the customer networks.

Hosting VNFs and SaaS on top of an overlay such as $NetUber$ is straightforward as these applications directly consume the cloud resources. As elaborated in Section II, $NetUber$ relies upon highly-optimized 10 GbE spot instances (i.e., R4), which are ideal to support all the above computation-intensive network functions, as opposed to smaller unstable spot instances. These optimized instances have abundant memory (244 GB memory each) and CPU resources. Thus, with a relatively stable memory, computing, and networking resources across the regions, $NetUber$ can be used as a framework for third-party network services on a cloud platform. We omit elaborations on such services for the sake of brevity.

IV. ECONOMIC FEASIBILITY OF A CLOUD-ASSISTED VIRTUAL CONNECTIVITY PROVIDER

Various pricing models have been proposed for connectivity providers [33], content providers [30], and clouds [38]. Cloud vendors list their VM prices at an hourly rate though they charge per second. $NetUber$ follows the same pricing scheme since it acquires its instances on a per-second basis. Since the connectivity providers list their charges per-month, we assume one month as the total time in our models, for a fair comparison. While a cloud-assisted network provider may be able to negotiate a discounted price with the cloud providers for a large-scale spot resource acquisition, we limit ourselves as regular users for the sake of a realistic evaluation. A cloud vendor itself may choose to operate a $NetUber$-like overlay. However, our interest is limited to the decoupling of the overlay provider from the cloud provider.

Equation 1 defines $\lambda_{o,d}$ as an end-to-end unit data transfer cost from s_o to s_d. Currently, cloud providers do not charge for incoming data from the Internet or another region. The $NetUber$ customers incur a cost (charged per used port-hours, at an hourly rate, by AWS Direct Connect), D_o and D_d, to connect the servers s_o and s_d to the cloud provider. Thus,

$$\lambda_{o,d} = \lambda_{s_o,r_o} + \lambda_{r_o,r_d} + \lambda_{r_d,s_d} = D_o + \lambda_{r_o,r_d} + \lambda_{r_d,s_d} + D_d \quad (1)$$

By substituting the values for the two (also can be generalized for more than two) cloud providers, Equation 1 can be extended to include the multi-cloud scenario depicted in Figure 4. $\lambda^{(1)}$ and $\lambda^{(2)}$ denote the unit data transfer costs by the cloud providers 1 and 2 respectively. The instance of the $cloud\ provider$ 2 in r_i is just an external server connected through the public (Internet-based) or a dedicated direct connectivity for the $cloud\ provider$ 1, whereas it functions as the origin cloud server from the perspective of the $cloud\ provider$ 2. Thus, the cost associated with the $cloud\ provider$ 1 is denoted by $\lambda^{(1)}_{r_o,r_i} + \lambda^{(1)}_{r_i,s_i}$, whereas the cost associated with the $cloud\ provider$ 2 is denoted by $\lambda^{(2)}_{r_i,r_d} + \lambda^{(2)}_{r_d,s_d}$. D_i denotes the cost of connecting the instances of both cloud providers at the region i. Hence, Equation 2 denotes the total cost.

$$\lambda^{(1,2)}_{o,d} = D_o + \lambda^{(1)}_{r_o,r_i} + \lambda^{(1)}_{r_i,s_i} + D_i + \lambda^{(2)}_{r_i,r_d} + \lambda^{(2)}_{r_d,s_d} + D_d \quad (2)$$

We formulate the total cost from all the vendors for $NetUber$ C, in Equation 3. C consists of the cost associated with acquiring the spot VMs and the cost of data transfer. $c_{v,r,i,t}$ defines the cost for an instance from the cloud vendor ($v \in V$) from a region at a given time step between t_0 and t_f. Since the spot price continues to fluctuate, the cost to acquire the required number of spot instances ($i \in I$) in each of the regions ($r \in R$) is calculated as a time integral over its execution time, and summed for all the instances from each region of all the cloud vendors.

The data transfer cost is billed by the cloud provider per the volume of data transferred. Therefore, it is calculated by a time integral of data rate b_t through a cloud path to its completion. Since $\lambda_{o,d}$ denotes the unit data transfer cost involving all the cloud paths, we calculate the total data transfer cost from the first $NetUber$ instance that receives the user traffic, $\forall i \in I_{r_o}$, for each region of all the cloud vendors.

$$C = \sum_{v\in V}\sum_{r\in R}\left[\sum_{i\in I}\int_{t_0}^{t_f} c_{v,r,i,t}dt + \sum_{i\in I_{r_o}}\int_{t_0}^{t_f}(\lambda_{o,d}b_t)dt\right] \quad (3)$$

We observed that the data rate of the inter-region data transfers b_t is proportional to the network interface (β) of the instance. β is 10 Gbps in the r4.8xlarge instances used by $NetUber$. However, it is impossible to reach the full network interface capacity in the inter-region data transfer. Cloud data transfers between regions have a degradation from the promised network interface. We define the degradation in the data rate of inter-region data transfer as a ratio of the network interface of the pair of VM instances, $\chi_t \in (0, 1)$. The actual data rate $b_t = \beta \times (1 - \chi_t)$.

Data compression at the source can significantly reduce the costs, given that cloud providers do not charge for the incoming data. Many cloud compression tools, general purpose or optimized for specific file formats, make lossless compressions feasible at the time of the cloud transmission [46]. By compressing the data before the inter-region transfer, we can significantly increase throughput or the actual data transferred per unit time. We define a compression ratio γ_t as the percentage of size reduction from compression without incurring data loss. γ_t and χ_t vary with time, unpredictable to $NetUber$. Equation 4 illustrates the effective data rate b.

$$b = \frac{b_t}{1-\gamma_t} = \frac{\beta \times (1-\chi_t)}{(1-\gamma_t)}; \chi_t, \gamma_t \in (0, 1) \quad (4)$$

$NetUber$ proposes to charge its customers based on their requested bandwidth (b), the length of the bandwidth usage (τ), and a direct unit (per time unit, per unit data rate) cost ($\Lambda_{o,d}$) to acquire the instances and data transfer from the cloud provider. c_{v_*,r_i} defines the cost of acquiring intermediate instances from any vendor v_*. c_{v_*,r_i} is 0 if $v_o = v_d$, as this makes $NetUber$ overlay with just one cloud vendor. β defines the network interface of the instance. To find the instance cost per unit data rate, we divide the cost of instances by the capacity of their network interface. $NetUber$ defines the charge Γ for its customer as a cost function (Equation 5). The cost function ensures that the total income of $NetUber$ from all its users for their connectivity demands remain higher than its cost of acquiring spot instances and data transfer costs.

$$\Gamma = f(\tau, b, \Lambda_{o,d}), \textbf{where} \quad (5)$$

$$\Lambda_{o,d} = \beta^{-1} \times (c_{v_o,r_o} + c_{v_d,r_d} + \sum_{if(v_o<>v_d)} c_{v_*,r_i}) + \lambda_{o,d}.$$

V. EVALUATION

In this section, we aim to answer two questions: i) when is $NetUber$ more cost-efficient than connectivity providers?, and ii) how does the performance of $NetUber$ compare to using direct Internet paths?

Prototype deployment. We deployed a prototype of $NetUber$ on multiple r4.8xlarge optimized spot instances (each with 10 GbE network interface) in each AWS region, as virtual routers. We performed an initial evaluation of various origin and destination regions. We reached around 50 Mbps between two instances in any regions with a single TCP connection. We achieved 1.2 Gbps of maximum stable inter-region bandwidth between the pair of 10 GbE instances with parallel connections. Thus, we deployed our prototype on at least 9 pairs of r4.8xlarge spot instances to achieve 10 Gbps bandwidth between two regions. We confirmed that the maximum bandwidth (1.2 Gbps) obtained was independent of the origin and the destination regions. By choosing a spot instance r_o in the overlay and placing it along with the origin server s_o in the same placement group, $NetUber$ ensures that s_o utilizes the complete 10 GbE rate when sending data to the overlay. Since spot instances are billed per second, we spawned the instances only when necessary. Instances are shared across users for multiple data transfers (at the same time, or at different time intervals, based on the data rate required for the transfer). Thus, the entire data rates of the instances are exploited, with minimal underutilization.

Infeasibility of using smaller instances. There are alternatives to the R4 instances that we used in $NetUber$: smaller spot instances and on-demand instances. The smaller instances such as C3 instances have a moderate network. We found two issues with these moderate network instances: i) we need to acquire a lot of them, which is more complicated to maintain due to the need for a substantial number of parallel connections, and ii) they are very unstable. We noted that with the cost of acquiring the number of 10 GbE instances, we could have around 2 - 4 Gbps, yet unpredictable, inter-region bandwidth with numerous moderate-network spot instances. However, the moderate network instances offered no promised guarantees for the bandwidth, unlike the R4 instances (r4.8xlarge and r4.16xlarge have 10 GbE and 25 GbE interfaces, respectively).

We were able to obtain the R4 spot instances promptly while having to wait for up to one hour for the moderate network instances. The r4.8xlarge of $NetUber$ instances stayed alive throughout the experiments which lasted up to 3 months, while smaller instances shut down frequently. Thus, we observe that it is possible to have a stable overlay over the 10 GbE spot instances, whereas currently, it is not feasible over the smaller ones. There was no difference in the quality of paths between the on-demand and spot instances. Thus, we observed that using smaller on-demand instances provide worse data rate or a much higher cost than using the 10 GbE spot instances for a functional prototype.

A. Economical Alternative to Connectivity Providers

To evaluate the cost efficiency of *NetUber*, we compare the data transfer cost of *NetUber* from Frankfurt to Sydney, against the offerings of 2 connectivity providers in the EU/US regions, marked as CP-1 and CP-2 in Figure 5. Due to cost restrictions, our extensive study only covers this pair of EC2 regions. However, based on the past approximate spot instance pricing details we gathered, we believe that our approach can be generalized to other pairs of regions. We also consider a compressed data transfer with *NetUber* in the evaluations, accounting for the potential compression-as-a-service deployment in the *NetUber* instances. We report the cost of the instances as the average price during the period.

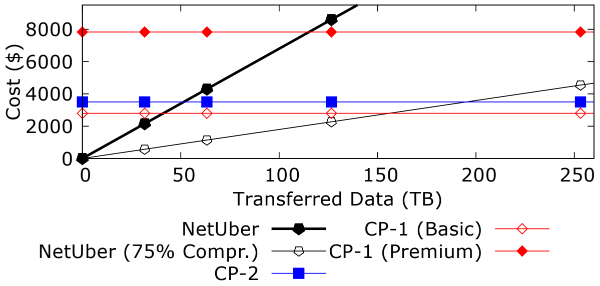

Fig. 5. Monthly Fee for 10 GbE Flat Connectivity

CP-1 is an infrastructure provider with an extensive, geographically-distributed infrastructure that offers connectivity as an alternative to transit providers. It provides two options: a basic scheme to connect to regular networks choosing the cheapest paths, and a more expensive premium scheme to connect to premium networks (connecting with large IXPs and premium networks) for better throughput and shortest paths. CP-2 is a transit provider. We obtained these price quotes via private email queries. As these quotations are not public (as typically transit providers do not list the prices in public), we must refrain from disclosing the providers. We include the costs of acquiring instances, data transfer costs, and the AWS Direct Connect cost for a continuous data transfer of the given volume for *NetUber*.

Up to 75% of lossless compression has been reported in compressing streaming data in real time [1]. We demonstrate a similar (0.75) or higher γ_t with in-memory compression at r_o. With these values, b = β · (0.12) / (1 - 0.75) = 0.48 · β = 4.8 Gbps, from Equation 4. While the inter-region data transfer achieves only 0.12 · β (or 0.48 · β with compression), AWS Direct Connect reaches the complete data rate of β. In Figure 5, we plot the minimum price of *NetUber* as the cost charged by EC2 for the spot instances and the data transfer. Here we consider both regular (γ_t = 0; χ_t = 0.88) and compressed (γ_t = 0.75; χ_t = 0.88) *NetUber* transfers.

3164.0625 TB per month (10 Gbps = 10/8 · 3600 · 24 · 30 GB/month) of data can be transferred between a pair of instances with 10 GbE interface. For volumes of data transfers up to at least 50 TB, *NetUber* always offered a competitive price (compared to the benchmarked connectivity providers) and remained globally profitable. When 75% compression is assumed, *NetUber* was still cheaper up to 200 TB. Thus, by leveraging the availability of abundant memory in the r4.8xlarge instances, we can employ enhancements profitable to *NetUber* or additional VNFs that can be executed as value-added services on the data traffic.

TABLE I
PING TIMES (MS): REGULAR INTERNET VS. *NetUber*

Origin → **Destination**	**Direct**	***NetUber (via)*** Improvement
Vladivostok → São Paulo	362.72	307.08 *(Tokyo)* 15.34%
Hobart → Mumbai	347.22	248.41 *(Sydney)* 28.46%
Seoul → São Paulo	321.72	299.31 *(Seoul)* 6.97%
Tashkent → Singapore	351.61	258.57 *(Mumbai)* 26.46%
Nairobi → Tokyo	403.87	386.37 *(Mumbai)* 4.33%
Frankfurt → Tokyo	296.87	237.34 *(Frankfurt)* 20.05%
Thuwal → Tokyo	346.01	324.30 *(Frankfurt)* 6.27%
Prague → São Paulo	224.90	221.40 *(Frankfurt)* 1.56%
Nuuk → Sydney	415.02	352.46 *(Canada)* 15.07%
Fairbanks → Mumbai	441.57	435.64 *(Canada)* 1.34%
São Paulo → Paris	239.45	210.72 *(São Paulo)* 12.00%
Tacuarembó → Montreal	203.42	186.01 *(São Paulo)* 8.56%

B. Higher Performance Point-to-Point Interconnection

NetUber is not always cheaper. But can it perform better when it is equally or more expensive than using the standard Internet-based connectivity? To assess the performance, we compare the round-trip time latency (ping time) between two endpoints that connect through *NetUber* against the latency using Internet-based connectivity of ISPs.We sent pings between the endpoints, first through the standard connection, and then via *NetUber* by entering the overlay through the nearest AWS region. For the geographically distributed servers, we used RIPE ATLAS Probes [15] and our physical servers.

We benchmark *NetUber* along with ISPs for faster Internet routes, against using just an ISP, as the chosen ATLAS Probes are connected by default to the Internet via an ISP. We repeated the evaluations ten times and listed the average ping times (in milliseconds) in Table I, along with the AWS region that the ping is routed through for *NetUber*, as well as the improvement when using *NetUber*. We measured the performance improvement as latency reduction, leaving other properties such as jitter and loss rate for future work. In all the cases, we observe that going through *NetUber* overlay offered better latency (up to 30% improvement) than directly connecting through the ISP, as long as a cloud region exists relatively near to the origin server, en route to the destination. The ISP-based connectivity remained the bottleneck in throughput as it reached up to only 50 - 75 Mbps.

The results indicate that even without dedicated connections to the cloud provider, an ISP user can resort to *NetUber* for better latency and throughput for data transfer and accessing SaaS hosted in a far cloud region, rather than directly connecting through the user's ISP. However, we predict a better latency and throughput when a dedicated connection such as Amazon Direct Connect connects the servers to the overlay. Similarly, *NetUber* can also be used in conjunction with

FTTH and community-based initiatives [14] for faster Internet routes, as they are not widespread. One can even use $NetUber$ selectively, such that only the transfers with a cloud region en route would go through the $NetUber$ overlay to achieve better latency. We leave further discussion on these for future work.

VI. RELATED WORK

A. *Cost Efficiency with Cloud Infrastructure*

Cloud-based infrastructures can increase the cost-efficiency of interconnections while minimizing the deployment time. Voxility [44] leverages its vast distributed infrastructure to provide network services and end-to-end interconnection, cheaper than the transit providers with more flexible agreement options for short-term and small-scale interconnections. CloudDirect [22] offers auxiliary services such as backup and disaster recovery atop cloud offerings. Cloud resources of $NetUber$ can be leveraged for more than just connectivity, including network services such as caching, content distribution to multiple local subscribers, and data analytics over wide-area networks [43].

Various approaches have been proposed, to reap the economic benefits, while addressing the technical challenges inherent to the volatile nature of spot instances. A third party or a broker leveraging resources from multiple cloud providers, and reselling them in a vertical market, has been found to be beneficial for both the broker as well as the cloud providers [36]. This multi-cloud infrastructure is in line with that of $NetUber$. Dynamic bidding policies are developed to support deadline-constrained jobs in spot instances [49]. Temporal multiplexing of burstable instances (to have a constant higher availability of CPU cycles) and spatial multiplexing of spot instances (to have reliable connectivity with redundancy in the path) can be performed inside a single AWS region with minimal overhead [26]. A trusted third party such as Google Fi [9] can function as a virtual ISP by exploiting the resources of multiple ISPs [50]. However, no comprehensive study has been conducted to realistically determine the feasibility of a virtual ISP that leverages the resources of spot instances, as a regular cloud user. $NetUber$ exploits the differentiated pricing of various availability zones for a relatively stable overlay.

Bidding in the spot markets of multiple regions can minimize the costs of CPU-intensive workloads, increasing the availability of the Internet services [34]. While $NetUber$ bids in multiple regions to acquire VMs in geographically distributed locations to host the virtual routers, it cannot use migrations between VMs in different regions for cost efficiency, as it will, in turn, increase the bandwidth consumptions and the number of hops. Cloud brokerage services have been built on spot instances with scheduling and reservation mechanisms, to minimize computing costs for jobs with a strict deadline, up to 57% [47]. Cost efficiency and performance of inmemory caches have been improved in the cloud, by deploying in spot instances while exploiting burstable instances for a backup [45]. But the scope of those research work is limited to computing, while $NetUber$ focuses on network connectivity, data transfer, and additional VNFs on the path.

B. *Decoupling the Internet*

Software-Defined Internet Architecture (SDIA) [39] decouples the architecture of the Internet from the infrastructure, by modifying the way interdomain tasks operate, through SDN and MPLS. SDIA and $NetUber$ share the goal of connecting endpoints on the Internet regardless of the underlying infrastructure. However, $NetUber$ focuses on network virtualization and does not alter how the underlying physical network works. Consequently, $NetUber$ can be deployed on existing cloud providers without any modifications to the cloud networks. Jingling [27] separates the network functions outside the network towards external “Feature Providers”. While Jingling delegates network functions to third parties, $NetUber$ virtualizes the entire network with virtual routers running atop spot VMs, by a third party broker. Both $NetUber$ and Jingling do not have control over the exact physical location of the system. Thus, specifying policies of the end users and identification of the cloud instances should be done through service layer instead of the physical address.

Virtual connectivity providers that do not control the infrastructure have been proposed, following an approach similar to that of $NetUber$ [25], [48], [21]. Cabo decouples ISPs into infrastructure providers and service providers, with concurrent networks that run multiple virtual routers atop each physical router, thus virtualizing links between any two virtual nodes [25]. Slicing the home networks can enable various service providers to reduce the costs and overhead associated with deployment and management, by sharing a common infrastructure [48]. However, $NetUber$ focuses on leveraging the cheap spot instances, and thus offers an economical approach to deploy on a large scale.

There have been industrial efforts following the same goal with $NetUber$, aiming at a fast direct interconnection between two endpoints, without relying on traditional connectivity providers in the region. PacketDirect [7] is an SDN-based platform that reduces the time to set up interconnections through its SDN-based framework. MPLS providers such as iTel [5] connect multi-location decentralized offices with a private layer-2 network, a unified connection to the whole organization. These providers differ from the traditional MPLS networks that merely provide connectivity between two endpoints, thus still requiring Ethernet connections for each office.

VII. CONCLUSION

Connectivity providers limit their agreements regarding minimum length and scale, preventing customers with short-term (in the scales of minutes, opposed to months), or minimal bandwidth requirements. $NetUber$ aims to address this as a virtual connectivity provider, built as a cloud-assisted overlay, running atop spot instances purchased from cloud providers for a low price. $NetUber$ aims for affordable end-to-end network connectivity for anyone, while not owning the infrastructure. In this paper, we built a case on why a virtual connectivity provider without any fixed resources may not just be technologically feasible, but also be economically sustainable.

We presented case studies with $NetUber$ as i) an economical alternative to connectivity providers for data transfers up to 50 TB, ii) a higher performance connectivity as an alternative to ISPs for end-to-end inter-region data transfer and accessing SaaS hosted in far regions without geo-replication, and iii) a provider for network services on top of a cloud-assisted overlay. Our analysis of the spot instance prices shows that cloud-assisted overlay network costs depend on a variety of factors, including geographical locations and current demand. We observe the enhancements in performance in comparison to ISPs and cheaper data transfer for small decentralized enterprises. As a future work, we envision an Internet-scale economic analysis and deployment of $NetUber$ on top of multiple cloud infrastructures.

Acknowledgements: We are thankful to Márcio Santos and Virgil Truica for their insights. This research was supported by European Union's Horizon 2020 research and innovation programme under the ENDEAVOUR project (grant agreement 644960), and national funds through Fundação para a Ciência e a Tecnologia with reference UID/CEC/50021/2013.

REFERENCES

[1] Compression as a service is now available in the cloud! boostedge.net for isps, telcos and enterprises ubfast.com for end-users!, 2013. Available at http://www.businesswire.com/news/home/20130207006266/en/Compression-Service-Cloud!-Boostedge.Net-ISPs-Telcos-Enterprises.

[2] Encryption as a service, 2014. Available at http://www.cloudlinktech.com/wp-content/uploads/downloads/2014/07/Data-Encryption-as-a-Service_R1.pdf.

[3] Console - the cloud connection company, 2017. Available at https://www.consoleconnect.com/.

[4] Epsilon Telecommunications Limited – Connectivity Made Simple, 2017. Available at http://www.epsilontel.com.

[5] iTel MPLS (IP VPN) High Performance Connectivity, 2017. Available at http://signup.itel.com/hubfs/Sales_Resources_InfoSheets/iTel_MPLS_Info_Sheet.pdf.

[6] Megaport, 2017. Available at http://megaport.com/.

[7] PacketDirect, 2017. Available at https://www.packetfabric.com/packetdirect/.

[8] Packetfabric, 2017. Available at https://www.packetfabric.com/.

[9] Project fi, 2017. Available at https://fi.google.com/about/.

[10] Amazon. Amazon ec2 spot instances, 2017. Available at https://aws.amazon.com/ec2/spot/pricing/.

[11] Amazon. Aws regions and endpoints, 2017. Available at http://docs.aws.amazon.com/general/latest/gr/rande.html.

[12] Amazon. How spot fleet works, 2017. Available at http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-fleet.html.

[13] Amazon. Placement groups, 2017. docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html.

[14] B4RN. Broadband for the rural north, 2018. https://b4rn.org.uk/about-us/our-network/.

[15] V. Bajpai, S. J. Eravuchira, and J. Schönwälder. Lessons learned from using the ripe atlas platform for measurement research. *ACM SIGCOMM Computer Communication Review*, 45(3):35–42, 2015.

[16] J. Barr. Aws outbound data transfer prices reduced by \$0.02/gb, 2010. Available at https://aws.amazon.com/blogs/aws/aws-data-transfer-prices-reduced/.

[17] J. Barr. Aws data transfer price reduction, 2014. Available at https://aws.amazon.com/blogs/aws/aws-data-transfer-price-reduction/.

[18] J. Barr. Aws blog. category: Price reduction, 2017. Available at https://aws.amazon.com/blogs/aws/category/price-reduction/.

[19] S. A. Baset. Cloud slas: present and future. *ACM SIGOPS Operating Systems Review*, 46(2):57–66, 2012.

[20] B. Boudreau. Global bandwidth & ip pricing trends, 2017. Available at http://www2.telegeography.com/hubfs/2017/presentations/telegeography-ptc17-pricing.pdf.

[21] C. X. Cai, F. Le, X. Sun, G. G. Xie, H. Jamjoom, and R. H. Campbell. Cronets: Cloud-routed overlay networks. In *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on*, pages 67–77. IEEE, 2016.

[22] CloudDirect. Move to Cloud ID - quickly, easily and securely, 2017. Available at https://www.clouddirect.net/.

[23] Cloudflare. Cloudflare argo, 2017. Available at https://www.cloudflare.com/argo/.

[24] Cogent. Cogent ip transit, 2017. Available at http://www.cogentco.com/en/products-and-services/ip-transit.

[25] N. Feamster, L. Gao, and J. Rexford. How to lease the internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1):61–64, 2007.

[26] A. Gandhi and J. Chan. Analyzing the network for aws distributed cloud computing. *ACM SIGMETRICS Performance Evaluation Review*, 43(3):12–15, 2015.

[27] G. Gibb, H. Zeng, and N. McKeown. Outsourcing network functionality. In *HotSDN'12*, pages 73–78. ACM.

[28] Google. Google cloud platform - cloud locations, 2017. Available at https://cloud.google.com/about/locations/.

[29] Google. Preemptible vm instances, 2017. Available at https://cloud.google.com/compute/docs/instances/preemptible.

[30] P. Hande, M. Chiang, R. Calderbank, and S. Rangan. Network pricing and rate allocation with content provider participation. In *INFOCOM'09*, pages 990–998. IEEE.

[31] O. Haq and F. R. Dogar. Leveraging the power of cloud for reliable wide area communication. In *HotNets'15*, page 19. ACM.

[32] O. Haq, M. Raja, and F. R. Dogar. Measuring and improving the reliability of wide-area cloud paths. In *WWW'17*, pages 253–262.

[33] H. He, K. Xu, and Y. Liu. Internet resource pricing models, mechanisms, and methods. *Networking Science*, 1(1):48–66, 2012.

[34] X. He, P. Shenoy, R. Sitaraman, and D. Irwin. Cutting the cost of hosting online services using cloud spot markets. In *HPDC'15*, pages 207–218. ACM.

[35] P. Lovelock. Unleashing the Potential of the Internet in Central Asia, South Asia, the Caucasus and Beyond. *ADB Consultant's Report*, pages 27–28, 2015.

[36] A. Ludwig and S. Schmid. Distributed cloud market: Who benefits from specification flexibilities? *ACM SIGMETRICS Performance Evaluation Review*, 43(3):38–41, 2015.

[37] B. W. Norton. Internet transit prices - historical and projected, 2014. Available at http://drpeering.net/white-papers/Internet-Transit-Pricing-Historical-And-Projected.php.

[38] R. Pal and P. Hui. Economic models for cloud service markets: Pricing and capacity planning. *Theoretical Computer Science*, 496:113–124, 2013.

[39] B. Raghavan, M. Casado, T. Koponen, S. Ratnasamy, A. Ghodsi, and S. Shenker. Software-defined internet architecture: decoupling architecture from infrastructure. In *HotNets'12*, pages 43–48. ACM.

[40] D. Richman. Amazon web services' secret weapon: Its custom-made hardware and network, 2017. https://www.geekwire.com/2017/amazon-web-services-secret-weapon-custom-made-hardware-network/.

[41] M. Scurrell. Batch computing at a fraction of the price, 2017. Available at https://azure.microsoft.com/en-us/blog/announcing-public-preview-of-azure-batch-low-priority-vms/.

[42] Teridion. Teridion, 2017. Available at https://www.teridion.com/.

[43] R. Viswanathan, G. Ananthanarayanan, and A. Akella. Clarinet: Wan-aware optimization for analytics queries. In *OSDI'16*, pages 435–450. USENIX Association.

[44] Voxility. Voxility - The secure infrastructure for your amazing Cloud Service, 2017. Available at https://www.voxility.com/.

[45] C. Wang, B. Urgaonkar, A. Gupta, G. Kesidis, and Q. Liang. Exploiting spot and burstable instances for improving the cost-efficacy of in-memory caches on the public cloud. In *EuroSys'17*, pages 620–634. ACM.

[46] Y. Xing, G. Li, Z. Wang, B. Feng, Z. Song, and C. Wu. Gtz: a fast compression and cloud transmission tool optimized for fastq files. *BMC bioinformatics*, 18(16):549, 2017.

[47] M. Yao, P. Zhang, Y. Li, J. Hu, C. Lin, and X. Y. Li. Cutting your cloud computing cost for deadline-constrained batch jobs. In *ICWS'14*, pages 337–344. IEEE.

[48] Y. Yiakoumis, K.-K. Yap, S. Katti, G. Parulkar, and N. McKeown. Slicing home networks. In *HomeNets'11*, pages 1–6. ACM.

[49] M. Zafer, Y. Song, and K.-W. Lee. Optimal bids for spot vms in a cloud for deadline constrained jobs. In *CLOUD'12*, pages 75–82. IEEE.

[50] L. Zheng, C. Joe-Wong, J. Chen, C. G. Brinton, C. W. Tan, and M. Chiang. Economic viability of a virtual isp. In *INFOCOM'17*. IEEE.