This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2947030, IEEE Transactions on Network and Service Management

An Economic Analysis of Cloud-Assisted Routing for Wider Area SDN

Prasun Kanti Dey, *Student Member, IEEE,* and Murat Yuksel, *Senior Member, IEEE*

***Abstract*—With the rapid growth of the Internet traffic and the intensity of online transactions taking place, it is expected that the current routing needs careful modifications and smart innovations to ensure effective and reliable end-to-end packet delivery. This involves new feature developments for handling traffic with reduced latency to tackle routing scalability issues in a more secure way and to offer new services at cheaper cost. Considering the fact that prices of DRAM (Dynamic Random Access Memory) or TCAM (Ternary Content-Addressable Memory) in legacy routers are not necessarily decreasing at a desired pace, cloud computing can be a great solution to manage the increasing computation and memory complexity of routing functions in a centralized manner with optimized expenses. Such cloud integration to routing is becoming plausible as cloud providers now offer various pricing schemes and provide large-scale computing infrastructure to meet the users' choice. Focusing on the attributes associated with existing routing cost models and by exploring a hybrid approach to SDN, we compare recent trends in cloud pricing (for both storage and service) to evaluate whether it would be economically beneficial to integrate cloud services with legacy routing for improved cost-efficiency.**

***Index Terms*—Network economics, Cloud computing, Cloud pricing, SDN, Routing, Scalability**

I. Introduction

THE Internet continues to witness a dramatic growth (3.88 Billion end users in June 2017 [2]) in traffic as people are more interested in diverse applications varying from watching high-quality videos, streaming music and playing online games to transferring bulk-data or making financial transactions online. According to *CISCO*, by 2021, more than 81% of total bandwidth to be consumed by video traffic [3] alone, which means the routers need to process more traffic and forces ISPs to fit the ever-growing number of prefixes in the existing routers in an efficient way.

Transition to IPv6 along with emerging Internet of Things (IoT) applications contribute to the extra space requirement in BGP routing tables. For example, IPv6 advertisements consume almost double the space (16 Bytes) than IPv4 routes in routing tables. Cisco Catalyst 6500 and 7600 series routers, like WS-SUP720-3BXL, can support 1 Million IPv4 routes, but only 512K IPv6 routes [4]. Research shows that BGP routing table growth has already exceeded the 700K mark [5] to ensure the network connectivity. Obvious measurement taken by ISPs, to cope up with the need for expanded FIB (Forwarding Information Base) table, is installing additional TCAMs or DRAMs in routers. But still, it is questionable how sustainable this approach will be in the future.

Moreover, configuration of traditional routers is tedious and error-prone due to human involvement as manual coordination between multiple of these routers require careful attention. To alleviate the situation, Software-Defined Networking (SDN) revolution became mainstream and data-centers embrace the SDN architecture because of its programmability, vendor-agnostic nature, and easier management. Flexibility to implement new innovations or patch a fix on the fly is another advantage of SDN as the forwarding plane is decoupled from the control plane, and as a result, individual development of these planes becomes natural as long as they are communicating via a standardized protocol (like OpenFlow [6]). Despite its benefits, concerns regarding SDN scalability [7] are uprising mainly due to its orchestration of a centralized controller.

Nevertheless, most of the SDN research encompasses use cases specific to data-centers, e.g., ElastiCon [8] deals with the static mapping of a switch to the controller, Avalanche [9] was developed to enable multicasting in switches, and a Network Virtualization Platform [10] was proposed for enterprise-level multi-tenant data-centers. The success of SDN in data-centers motivates researchers to extend the software-defined approach to find solutions in other scenarios like wide-area networks (WANs) that require high-end routers, firewalls, optimizers, and complex configurations for consistent performance. Centralized control, following the SDN technique, in Software-Defined WAN (SD-WAN) can exploit the holistic view of the network for dynamic load-balancing, handle various types of connectivity, and reduce complexity in management [11]. Yet, economics and scalability of wider area SDN deployments as in SD-WAN are not explored well.

From an economic perspective, ISPs' routing business model involves myriad items in the cost model that impact the price of network services [12]– geo-location, traffic amount, intellectual property or software licensing, and infrastructure or operational expenses required in the process to name a few. But, the primary contributing factor still remains the unit cost of router memory which is not reducing in accordance with Moore's law [13] to keep the routing cost manageable while BGP table size is constantly increasing.

In this context, to circumvent the scalability issues aroused from SDN implementation, and help ISPs maintain a pleasant economic outlook, cloud computing acts as a harbinger who brings a new opportunity by offering its computational power to handle complex and time-consuming tasks by relaxing the

Manuscript first submitted on Jun 2, 2018; revised on Feb 28, 2019 and July 4, 2019; accepted on Oct 4, 2019.

P. K. Dey and M. Yuksel are with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, 32816 USA (e-mail: prasun@knights.ucf.edu, murat.yuksel@ucf.edu).

A preliminary version of the work was published as short paper [1] in 2016 5th IEEE International Conference on Cloud Networking (Cloudnet).

resource constraints, sometimes practically eliminating them. The proliferation of cloud paves an excellent way [14] for expanding businesses by sharing and multiplexing resources statistically among its tenants to increase utilization without actually hurting the performance customers are receiving from the cloud at any given instant. *On-demand self-service*, capability of *rapid provisioning and release*, *precise measurement of usage* [15] make cloud a low-cost, programmable alternative to expensive routers for provisioning network connectivity and services that an SD-WAN adopter enterprise can opt for, allowing them to scale up as the demand increases [16]. Startups like VeloCloud, Aryaka are providing this capability to the enterprises and according to Juniper, communication service providers confirm that "SD-WAN is strongly aligned with the criteria for cloud success on many levels" [17].

While router hardware price is not declining constantly, the price of entry-level average public cloud services has been reduced to a staggering 66% during the period of 2014 to November 2015 and the price is expected to fall by 14% more by 2020 [18]. Although the price is dropping rapidly, it is expected to become stable soon as the market evolves and gains substantial maturity. Competitors will be encouraged to enhance their pool of resources by adding more high-end machines for no extra charge [19].

In this paper, we explore the economics of wider area SDN designs and characterize how sustainable SDN solutions can be at longer distances than inside of a data-center. In particular, we study how beneficial it may be to utilize cloud services for solving the increasing memory complexity of routers. We formulate the overall concept as "Cloud-Assisted Routing" (CAR), a potential solution to the scalability concerns of wider area SDN and compare it with legacy routing. The key question we aim to answer is that *can the partial placement of control and data plane routing functions to a remote cloud, reachable only via public Internet transit, be economically viable*? Major contributions of our work include:

- More than 30 years of DRAM price data showing the trend in router memory price (Section IV-A).
- A detailed empirical analysis and modeling of cloud storage and service prices for the last 8 years to audit the inclination of cloud related services' prices (Section IV-B, IV-C).
- Cost models for legacy routing and CAR to compare the effect of memory price vs. modified routing design with cheaper alternatives (Section III-A).
- Characterization of FIB size needed, with respect to the legacy routing, at a local router to attain a target cost reduction in CAR framework (Section V). This FIB size threshold outlines the region of operation where the CAR architecture is quantifiably cost effective.
- Understanding of the peering influence on the sustainability of wider area SDN concepts like CAR and future routing scalability under two extreme scenarios: no peering at all vs. complete peering (Section VI).

A preliminary economic analysis was presented in [1] and the architectural discussion was detailed in [20]. This paper extends by *a)* adding more data for DRAM and cloud prices, *b)* introducing the idea of Plutus point (optimum FIB cache size and associated maximum savings in CAR), *c)* stating the remarks on the ratio of total incoming traffic and FIB size for long-term savings, and *d)* commenting on what-if conditions based on heavy and light peering scenarios. Prior to delving into details, our paper elaborates the motivation and related works in Section II. We also describe CAR architecture in Section III followed by a detailed price comparison analysis for all variables associated with the cost models in Section IV. Next, we show the break-even points and compute the CAR savings in Section V. Section VI investigates the impact of peering decisions between two ISPs on CAR. Finally, we conclude the paper in Section VII with our observations and suggestions for the future optimizations.

II. Motivation and Related Work

Access time in Static Random-Access Memory (SRAM) is very small (~ 4 ns) compared to DRAM (~ 40 ns), which makes SRAM a perfect choice for router memory. Due to SRAMs power-hungriness and overheated nature, its usage is stringent (few Megabytes) and additional DRAMs (CISCO 4400 Series has 2-8 Gigabytes) [21] are being introduced to store routing tables. Again, on-chip memory (CPU cache or FPGA block RAM) usage has not increased as well because of being very expensive with respect to conventional off-chip memory (DRAM) [22]. As a result, research in this area mainly focuses on developing memory-management algorithms (e.g., SMALTA [23], FIFA [24]) to optimize IP lookup time by aggregating FIB [25].

Multibit-trie architectures such as Tree Bitmap [26] have attained much popularity in high-end routers (e.g, Cisco CRS-1 Multishelf System [27]) because of its faster updates and searching capability. But, this approach requires more memory and thus other tree-based architectures (e.g., FlashTrie [28], PopTrie [29]) have been explored to overcome the shortcomings. Another effort was taken by Rétvári et al. [30] that demonstrated whether it was possible to guarantee IP lookup performance by squeezing the existing router hardware memory to facilitate the ever-expanding FIB table.

Keeping technical complexity in FIB memory management aside, from pure business perspective alone, ISPs tend to form bilateral (zero-dollar) peering settlements or customer-provider relationships without considering the global view when it comes to maximizing their profit. These service level agreements (SLAs) sometimes result in inefficient routing because of prioritization [31], impose unnecessary network instability [32] due to conflict in peering policies, and thus, have adverse impact on overall Internet ecosystem and ultimately less aggregated market profit.

Network economics aims to explore strategies to minimize the Capital Expenditures (CapEx) and Operational Expenditures (OpEx) to maximize ISPs' profit. Ma et al. have investigated a game theoretic approach [33] to achieve an efficient, fair and optimal routing among a group of profit-sharing ISPs. A Cost-Aware (CoA) caching [34] scheme has also been proposed to see the feasibility of cost minimization that contradicts popular cache algorithms with fundamental in-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2947030, IEEE Transactions on Network and Service Management

tention of attaining maximum hit-ratio, and rather emphasizes specifically to reduce cost and offers economic incentives.

However, most of the network economics analysis are limited to either peering business relationship between multiple ISPs or how to shape the traffic to conciliate individual ISP budget for reducing the cost. Having said that, Motiwala et al. [35] present a cost model which considers the total volume of traffic flow and the cost of carrying them through the network, offering the operators an opportunity for traffic engineering in path selection by identifying the most expensive flows and routing them through an alternate, less-utilized and more economical transit. This is the closest to our proposed model. Indeed, they classify the main cost contributors into two categories, namely *Interconnect costs* that comprises transit fees, port costs or some fixed costs and *Backhaul costs* which represent circuit, capital, and operational costs altogether. We are interested in one of the components of *Backhaul costs*, i.e., router cost, to be precise.

Recent observations [36] on the cloud being cheaper, closer and higher quality (cloud challenges are reducing) attract networking community for a longer term. Cloud service providers are not only reducing the price but also they are investing more to offer newer feature sets and innovative services by developing their high-computation infrastructure that is capable of supporting a wide range of new applications. Features like load-balancing and auto-scaling have already become a common practice by the major providers like Amazon, Microsoft, or VMWare. Vendors are shifting towards *per-second* from *per-hour billing* schemes to provide more precise and detailed billing, which benefits enterprises with much flexibility [37]. Offerings of additional discount (up to 75%), albeit the requirement of a committed usage over the period of one to three years, make the cloud a lucrative choice to include while designing a cost-effective architecture.

Since the emergence of cloud computing, Amazon leads the industry with Amazon Web Services (AWS), while other big companies like Google and Microsoft have branded their own services as Google Cloud Platform and Microsoft Azure. All three provide file *storage capability* (Simple Storage Service (S3) by Amazon, Google Drive by Google, OneDrive by Microsoft) as well as facilitate services like *Software as a Service (SaaS)*, *Platform as a Service (Paas)* or *Infrastructure as a Service (IaaS)* and charge users accordingly. Being the dominant player in cloud computing market, Amazon sets the tune by continuously slashing cloud service price by 16% to 28% (varies by region and services) [38], [39] which, in turn, impels other competitors to follow the trend.

Customers like *Netflix* has already migrated to AWS to handle its 1000x growth in monthly streaming hours [40]. All of its video contents, business logic, data analysis and service availability is now hosted on Amazon. *AirBnb*, *Adobe* are also housed in AWS and are using EC2 (Elastic Compute Cloud) with other services for load balancing, simplified auto-scaling or efficient supervising purposes [41], [42].

Google cloud customers like *Snapchat* uses storage service for storing images, compute engine for image processing, and BigQuery for data analysis [43]. *Spotify* also uses BigQuery to analyze users' listening patterns to provide a better experience while *Best Buy* is using Google App Engine to reduce maintenance overhead for quick development and scalability [44].

Armbrust et al. [45] discuss the elasticity of the cloud system by showing an example of pay-per-use pricing for the cloud. In business, it is beneficial to have the ability to add or remove resources at any time. Methods of cloud pricing are always an issue between the user and cloud system [46]. But, earlier works primarily accentuate the computational expenses caused by distributed systems on the cloud instead of evaluating the storage price.

In addition to analyzing the router hardware cost in our model, we also need to understand the future cost-effectiveness of this trend of *delegation to cloud*. Our study investigates this issue and, specifically, we look at the consequences of delegating routing (network layer) functionality, partially, to cloud with a focus on FIB table caching and its causation of data packet delegation to cloud. In this regard, we also consider the transit cost incurred due to the packets delegated towards cloud using physical infrastructure. To augment our cost model, we use data transmission cost model based on the quality of service, proposed by Fishburn [47].

III. A Comparative Model

An architectural view of hybrid "CAR router" is illustrated in Figure 2. It is to be noted, CAR aims to find a middle ground where it can exploit both the local hardware to scale router performance and a completely cloud-based approach for a highly flexible routing service. Following the basic SDN architecture presented in Figure 1, wider area SDN deployments will require southbound API protocols to be implemented over longer distances. CAR is looking at the case when the southbound API may need to be implemented over public transit. Similar to how virtual memory systems use secondary storage to maintain full content, CAR uses the cloud to implement the full functionality of *Router X (RX)* and keeps *RX* as 'active' while *Proxy Router X (PRX)* as 'passive'. CAR follows a homogeneous approach of RouteFlow [48], previously known as QuagFlow. The differences are, in CAR, *a)* the controller sits on the cloud, and both *RX* and *PRX* can act as separate entities, and *b) RX* and *PRX* are capable of establishing BGP peering with others by themselves. CAR designers should follow these two principles [20]:

1) Computationally intensive but not-so-urgent tasks (e.g., BGP table exchange during peering establishment, shortest-path calculations, spanning all entries) should be offloaded to cloud as much as possible. It is *PRX*'s responsibility to store the full FIB table and in an unlikely event of *RX*'s failure to handle data and control plane functions, *PRX* should act as the default point of service. Checking the drop packet statistics in *RX*, periodically or after a certain time interval, *PRX* should send updated FIB information to *RX*.
2) Keep data plane mostly at the *RX* while some of the control plane operations such as on-demand route computations due to failures, collection of flow-level simple statistics or request for updated routing table will still be done at *RX*. However, CAR should orchestrate heavy routing optimizations at *PRX*.

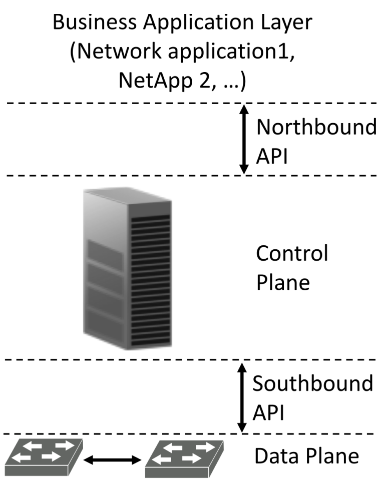


Fig. 1. Basic SDN architecture

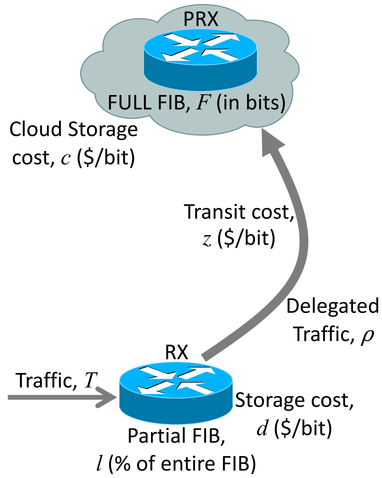


Fig. 2. CAR architecture

d	DRAM price
c	Cloud storage price
s	Cloud service price
z	Transit price
l	Percentage of FIB cache in *RX*
C	Traditional routing price
$\hat{C}$	CAR price (with cloud storage price only)
$\hat{C}_s$	CAR price (with cloud storage and cloud service price)
F	FIB size
T	Total incoming traffic to *RX*
ρ	Amount of traffic delegated to the cloud
ψ	First constant to represent temporal dynamics in router traffic
ω	Second constant to represent temporal dynamics in router traffic
γ	Scale factor
λ	Percentage of transit traffic

TABLE I
LIST OF SYMBOLS USED IN THE PAPER

Even though *PRX* can be designed to keep only the remaining not-so-popular prefixes that *RX* does not store, such implementation is not advisable since this will generate a higher volume of exchanged CAR messages between the proxy and the active routers. During every single *route update* phase, *RX* has to share its *partial* FIB with the cloud so that *PRX* can populate a new list of popular prefixes combining its own table and the table it received from *RX*. While in “full list stored in *PRX*” case, *RX* can refrain from sending its FIB table to *PRX* and reduce the CAR message size.

Due to the Internet’s *best effort* nature blended with heterogeneity of cloud service configurations, it is impossible to guarantee the latency and reliability of optimal cloud services, and therefore, related challenges need to be addressed before migrating routing functions to the cloud. CAR providers may incorporate multiple clouds in the design and use an intermediary, similar to CloudCmp [49] or Smicloud [50], to select the “best” public cloud provider based on *RX*’s location, response time and provider’s service cost. With a strategic selection of the *PRX*, latency degradation can be compensated.

Empirical evaluation shows that the background load (generated by multiple tenants) of a cloud provider may interfere with the perceived performance of latency-sensitive tasks [51], and already existing real-time multimedia applications such as cloud gaming are the worst sufferers from such incidents. However, dynamic utilization of the datacenters located near to end-users improves the latency dramatically, and, by preventing the tenants to go beyond their allowance (in the static reservation), services like Silo [52] can offer predictable latency. CAR can also capitalize on such techniques to minimize the latency caused by offloading routing service to the cloud.

As far as the ISPs’ CapEx and OpEx is concerned, utilizing multiple cloud vendors will be effective to achieve cost savings. With CAR attaining enough momentum and more ISPs embracing its design, it is possible to observe such scenario where *i)* multiple routers of an ISP and/or *ii)* multiple ISPs are being served by the same cloud vendor. In both of these cases, CAR architecture will not only help ISPs to save money in OpEx as well as CapEx but also will be able to perform *intra-* and *inter-domain* routing optimizations.

There is a tradeoff between the offloading cost and performance degradation. In general, as the routing functions are moved towards the cloud, the delay between the router and the full routing functionality increases but the offloaded functions relieve the router. Having a “fog computing” support closer to the routers will certainly help to add another level of caching to the routing functions. A key potential gain is to perform centralized optimizations and control tasks at the cloud [20], and implementing traffic engineering decisions there. Thus, the CAR designer’s decision here is to fine tune this tradeoff by choosing the right number of caching levels and their distances to the router(s), and the amount of resources at each level.

Akin to already mentioned key contributors of ISPs CapEx and OpEx in Section I and Section II each cloud provider identifies their own cost factors, therefore it may vary, but the most common ones [53] include *i)* server cost (per GigaByte data storage, RAM size), *ii)* computing cost (running Virtual Machine hours per vCPU), and *iii)* data transfer and network cost (upload/ download bandwidth, IPv4 or IPv6, number of VLANs). In the following sections, we shall particularly explore the cost models involving these factors and see how much they have evolved over the course of time.

A. CAR vs. No CAR Cost Models

To begin our analysis, first of all, we have constructed a very naive cost model for CAR and traditional routing system based on DRAM price, transit cost, cloud storage, and service price. However, infrastructure costs like laying a sub-sea cable and equipment purchase for running the business or administrative overhead of technical staffs are ignored in this study as we are focusing on long-term consideration and these costs will eventually be distributed over time. For instance, we do not consider the fact that in 2016, US broadband providers invested approximately $76 Billion in network infrastructure and the total expenditure from 1996 to 2016 was $1.6 Trillion [54].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2947030, IEEE Transactions on Network and Service Management

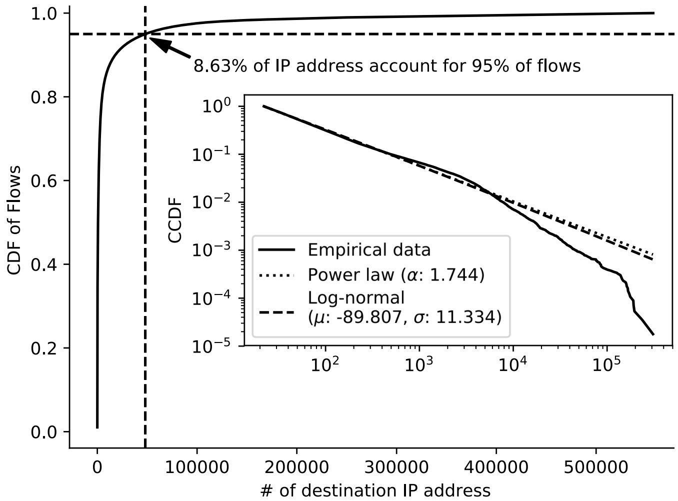


Fig. 3. IP prefix count vs. flow count

Let, d and c be the \$/bit cost of storage at the local DRAM and the cloud, respectively. Considering, F (in bits) as FIB size and l be the percentage of FIB that needs to be stored at the local router (basically, size of FIB cache) to sustain an acceptable average delay of forwarding time for packets towards the cloud. Here, "acceptable" means the new forwarding delay should be very close to the traditional router lookup delay. Assuming z (in \$/bit) being the delegated packets transmission cost to the cloud, models for the operational cost of traditional router C and CAR router $\hat{C}$ can be formulated as follows:

$$C = dF \qquad \text{(Eq. 1)}$$

$$\hat{C} = dFl + cF + z\rho(l) \qquad \text{(Eq. 2)}$$

where $\rho(l)$ is the amount of traffic delegated to the cloud and follows a Log-normal decay distribution function of l due to the significant locality in traffic. We have analyzed the *Anonymized Internet Traces* collected by CAIDA's *EQUINIX-NYC* monitor [55] during December, 2018 and observed the spatial (few popular prefixes) locality behavior (see Figure 3) displayed by the traffic. The analysis shows that 8.63% of destination IP addresses account for 95% of traffic flows at this major traffic exchange. Previous studies also support this claim of very high locality (10% prefixes account for 97% of the total traffic [56]) in the prefix lookups at a router. Assuming ψ and ω be the two constants for such temporal dynamics of traffic destination in routers and being the total incoming traffic towards RX, we can express ρ as:

$$\rho(l) = [1 - \psi \ln(l) - \omega]T \qquad \text{(Eq. 3)}$$

The more FIB entries we store, as l increases, the more bits will be needed at the local DRAM and less traffic, $\rho(l)$ will be delegated to the cloud. According to Figure 3, it is reasonable to expect that $l < 10\%$ will be enough to support most of the traffic locally and a very small amount of traffic will be delegated. Since the transit cost depends on the amount of delegated traffic, it will hence stay low as long as l is small, e.g., for $l < 10\%$ less than 5% of the traffic will incur transit cost. Thus, the transmission cost (third term in $\hat{C}$) of the delegated traffic will be fairly low due to its lower volume.

In the following sections we use exponential decay to model the cloud storage and transit costs as historical pricing data shows that these services are becoming a commodity. This is inline with the overall trend of bulk storage and communication prices declining exponentially in terms of the per unit price (e.g., \$/bit). This does not mean that the providers will give them for free as the volume of the services customers needs is also increasing. Given this, our analysis aims to reveal which one of the three terms in Eq. 2 will be the dominant factor in regulating a CAR router cost. In that sense, since the last two terms have exponentially decaying per unit prices c and z, they will not be the key factors in the overall CAR router cost as long as their multipliers F and $\rho(l)$ are under control. Since F also exists in the first term, the second term will not be the determining factor in comparison to the first term. As for the third term, we observe in Figure 3 that $\rho(l)$ decays according to a Log-normal distribution. An exponential decay would mean a faster decay, but the delegated traffic can still be kept small by properly exploiting the locality in the traffic as seen in Eq. 3 and Figure 3. As such, the Log-normal distribution of the delegated traffic decays very fast, and once a small portion of the FIB is cached, $\rho(l)$ will be kept small.

Therefore, the driving factor will be the first term, dFl, which is less than C. Thus, as long as l is managed properly via good prefix replacement algorithms (i.e., FIB caching algorithms), CAR routers will always be more cost-effective. It is worth noting that C will likely to have more terms in addition to dF due to the shifting of control planes tasks to remote platforms. Also, note that compressing FIB [57] does not really change the overall comparative analysis here since a similar study can be made involving SRAM costs.

IV. Price Comparison

Forecasting memory price is uncertain due to its market dependency. Supply-demand mismatch, environmental hazards or even company policy can impact on the price variation. Regardless, this price is certainly not comparable with the current price offered by cloud providers. A back-end cloud service (including storage facility) and a transit service towards the cloud are two key components of CAR. Historically, the prices for all of these services have been reducing as we shall see in the following discussion.

A. DRAM Price, d

We use memory price data from McCallum [58] for our study. This dataset contains memory prices from 1957 (transistor Flip-Flops) to 2017 (DIMM DDR3-1600), but we restrict our starting date from 1984. Figure 4 shows the decaying trend in DRAM prices with ups and downs on multiple occasions like 1988-1990 or 1996-1998 periods. In 2013, the price increased by 40% compared to its previous year which is not visible in this figure as the recent price of DRAM is extremely low compared to its initial predecessors. To get a more refined view on the recent years, we plot average DRAM prices for the last 10 years in Figure 5 separately. It is noticeable that the price started falling after 2014 and reached its lowest (\$3.55)

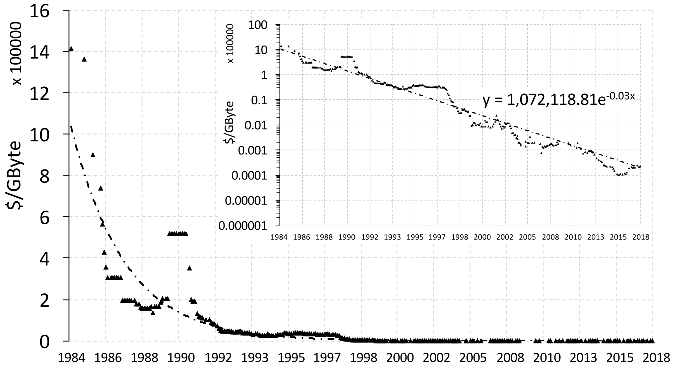


Fig. 4. Historical DRAM price trend

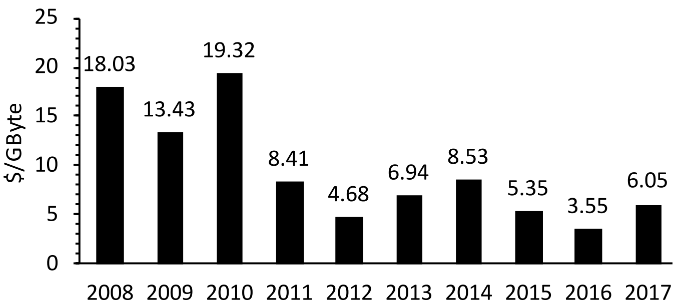


Fig. 5. Average DRAM price (2008 to 2017)

in 2016. After that, the price is increasing again and is reported to climb at least by 10% on average in 2018 [59].

Major DRAM makers like Samsung, Micron, SK Hynix are undergoing a transition, as they are competing to take the future lead and investing more to produce 18nm-class DRAM instead of 25nm or even 20nm wafers. Moreover, robust and continued demand from the mobile industry who are packing 4GB or 8GB of RAM in a smartphone contributes to the tight supply and thus leading to the most recent price hike.

Taking all of these into consideration, how much cheaper the (DRAM) memory price will become, in future, is unpredictable. According to the trend-line (in Figure 4), we expect the price to reduce exponentially with a small decaying exponent, but it will not necessarily be extremely economical in near future than what it is of today. Yet, based on the DRAM prices in 1984-2016 and favoring the traditional routers, we model the DRAM price with an exponential decay with respect to time, t:

$$d(t) = 1,072,118.81e^{-0.03t} \quad \text{(Eq. 4)}$$

B. Cloud Storage Price, c

Gartner introduces Magic Quadrant, a graphical representation of research providing a summarized insight about any given market, to position the competitors into four categories: leader, challenger, visionary and niche. According to them, AWS and Azure are the market *leader* and Google Cloud is marked as the top *visionary*, who is late to join the market (eight years later than Amazon) but can shift the momentum into its favor anytime [60]. We base on this criteria and will limit our discussion primarily on these three providers.

Cloud storage cost is advertised as per GB per month, even so, users have to pay some associated costs depending on the

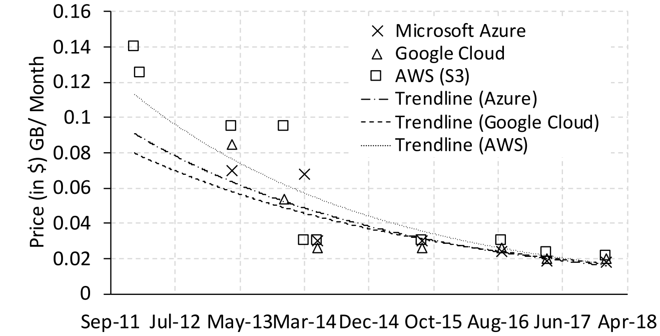


Fig. 6. Cloud Storage price-trend for three providers

providers' business strategies. Costs like storage request or transaction count, HTTP operations (i.e, GET, PUT) counts are often referred to as "hidden costs", although service providers mention them in their 'terms & conditions' [61]. For instance, both Amazon S3 and Microsoft Azure charge for both of these operations while Google does not charge for PUT requests. For simplicity, we discard these variable costs and base our analysis only on the storage costs.

Figure 6 evaluates the declining price trend for cloud storage from January 2012 to January 2018. One interesting observation we have found, albeit of its absence in the figure, AWS initially launched S3 in 2006 setting the price at \$0.15 per GB/month and continued to charge so (\$0.14 in 2010) until 2013, which is the year, Google publicly introduced its cloud services with a cheaper price for the first time. To match Google cloud storage plan, AWS halved the price; and since then, these two providers and Microsoft are battling to offer the lowest price to attract new customers.

Our proposed *CAR* architecture, to become efficient, needs faster storage services, hence we consider LRS-Hot (Local Redundant Storage) for Azure, S3 for AWS, and Google's Regional storage option. We discard other lower price options available from each of the vendors like LRS-Cold (Azure) with the price of \$0.0152, Glacier (Amazon) with \$0.004 and Coldline (Google) with \$0.007 as these are comparatively slower and will not grant frequent access that is needed for *PRX* in the CAR architecture. It may be very effective for CAR to have a multi-regional redundancy to offer intra-ISP optimization. But, Amazon's lack of similar service till date and without enough concrete evidence of benefits achieved from such approach, to maintain consistency, we have not included multi-regional storage price analysis in this paper.

If the current trend continues, each of the cloud providers will offer really low-cost unlimited storage. As the price goes down, providers will be motivated to promote more value-added services with a very little tweaking in their infrastructure. It will encourage them to implement routing feature on the cloud system, where the user will be able to rent a specific sized virtual router by paying a fixed (or on-demand) fee. It can be treated similarly to already existing cloud-based virtual machines. From the pricing data in Figure 8, we have deduced the following decaying cost equation for cloud storage:

$$c(t) = 0.12e^{-0.03t} \quad \text{(Eq. 5)}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2947030, IEEE Transactions on Network and Service Management

Fig. 7. Hourly price comparison for On-Demand (OD) vs. Reserved Instance (RI) for 1 or 3 years (Normalized to hours) cloud service

It would be interesting to see how much these vendors are willing to lower the price to compete with the new providers like Rackspace or Backblaze who are now offering similar services with almost one-fourth of the market price [62].

C. Cloud Service Price

Using memory and compute resources of a cloud provider also involves "servicing" price for the various labor needed to set up the cloud service. Comparison between cloud service price and actual physical router performance cost is not straight-forward. Even so, Newnan [63] identified the fixed cost for setting up the entire routing infrastructure, the variable cost of power and employee salary, the marginal cost of each additional performance improvement as the priority. If we opt for cloud service instead of managing the router by ourselves, it becomes easier to calculate the cost. Cloud providers charge hourly basis for on-demand service and offer discounts for year-long commitment. For convenience, we have normalized all prices to hourly-basis in order to compare them.

Among the yearly committed discount options, AWS Reserved Instance (RI) requires 1 year or 3 years of commitment and saves around 24%-75% depending on the duration, Azure has a similar policy with savings of 15%-45% and Google offers a flat 37%-55% discount per year on its sustained use policy. While IBM still does not have any yearly plan, it negotiates for a month-to-month agreement with about 10% reduced price. Discounts may vary based on the upfront payment method as well, for example, AWS allows no upfront, partial or full upfront payment options.

We plot a comparison between on-demand pricing and year-long commitment for four vendors in Figure 7. As AWS matures, we notice, it is not radically dropping the service price now like it was doing before ("on 44 different occasions over the last six to seven years", stated in 2014 [64]), and rather introduces multiple new instance types with better performance for same or even cheaper price over the period we conducted our research. We always prefer the latest instance type, as suggested by AWS, with similar capability and try to be consistent without remaining glued to a specific type. We choose *m2.xlarge* for the year 2014, *r3.large* for the year 2015 and continue with *r4.large* afterward as all of these have

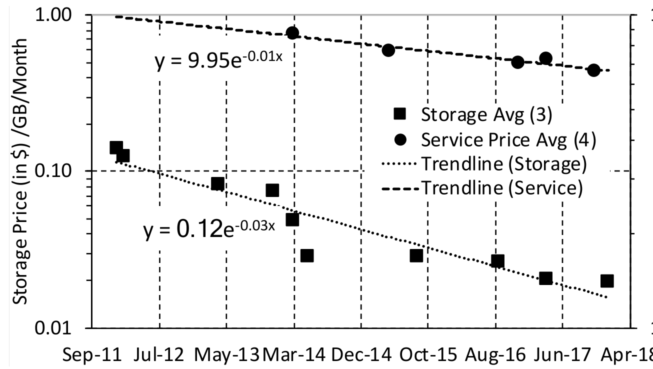


Fig. 8. Cloud storage vs. service price trends (log-log scale)

2 vCPU with 15 GB of RAM. For compatibility, we select *n1-highmem-2* from Google Cloud and *D11 v2* from Microsoft Azure as both of them have the same number of vCPU cores with RAM size of 13 GB and 14 GB, respectively.

Figure 8 illustrates trends of average cloud service price and cloud storage price simultaneously. It is clear, from the graph, service price trend-line is less steep than the storage one. This supports our observation of a market that is approaching to a steady state, where the cloud providers will not consider price reduction as their main selling point but will delve into developing new features to attract customers instead. Taking the average value to represent the entire business scenario, we get the following decay equation for cloud service price:

$$s(t) = 9.95e^{-0.01t} \tag{Eq. 6}$$

D. Transit Price, z

Packets delegated from the router will be transmitted over the actual physical link towards the cloud incurring transit cost. Depending on link type (i.e., dedicated or shared), the cost will vary. ISPs treat data transmission cost as the epitome of monumental costs involved in Internet business, as it, alone, asks for almost half of the long-haul network expenses [47]. However, the improvement of WDM (Wavelength Division Multiplexing) empowers network operators to constantly reduce the price per unit bandwidth by facilitating the expansion of transmission capacity without any extra fiber line setup.

To explore the trend in transit business, Fishburn and Odlyzko [47] considered two types of data demands. First one is delay insensitive (A) and another one is sensitive to delay (B). They also proposed two different services for these data types: *a)* separate network for A and B with different pricing schemes; and *b)* a single network with a unified price for both.

Analyzing this work and based on our research with prices for minimum commitment [65], [66], [67], we plot transit costs in Figure 9. We observe the decay trend in transit costs with larger R^2-ed value, supporting the well-fitting nature of our linear regression model. This decay trend will eventually promote more off-loading of packets to cloud for routing purpose. To ensure the validity of the model, we have also considered the existence of bias in the prediction by evaluating the residual plot (which we do not include due to space constraints) and found that the values were scattered randomly around zero with the residual center at zero.

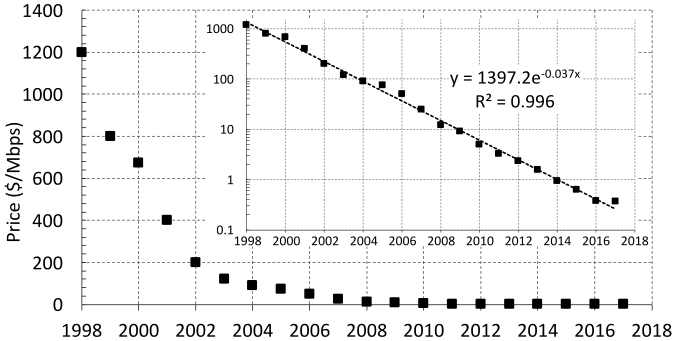


Fig. 9. Transit price trend

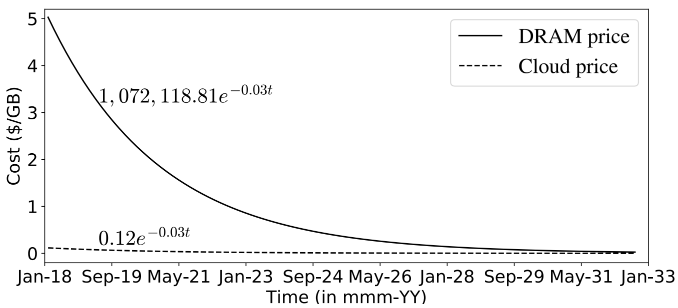


Fig. 10. Cloud storage price vs. DRAM price

For CAR, utilizing a dedicated transit system to delegate the packets to the cloud would be the best. Having said that, such dedicated service policy does not exist, except leased fiber lines. But, since this would limit the deployment of CAR to only those *RX*s that can have leased fiber line connections, we do not consider such leased transit service in our model. The only available option for high-bandwidth transit are the ones offered for generic use, which we consider. Fitting the data to an exponential decay gives us the transit cost equation as:

$$z(t) = 1,397.19e^{-0.037t} \quad \text{(Eq. 7)}$$

V. Economic Viability

Our modeling effort thus far gives an opportunity to explore the economic viability and scalability of cloud-assisted SDN architectures. We will now look at how legacy and cloud-assisted routing (CAR) will compare in terms of costs, find break-even points and explore regimes where one may be more beneficial in the emerging trends of Internet routing ecosystem.

Figure 10 plots the cost comparison between cloud storage and DRAM. According to the graph, the cloud price is cheaper and if the current trend continues, it will take at least 15 years for DRAM to catch up. Referring to our discussion in Section IV-A, DRAM price does not exactly align with the decaying trend, which means, according to this graph, cloud storage will certainly be more profitable compared to DRAM in future. Hence, usage of cloud storage for routing purpose will likely be economical and more common.

A. Break-even Points

Based on the cost models (Eq. 1) and (Eq. 2), we find the break-even points between CAR ($\hat{C}$) and No CAR (C) cases in

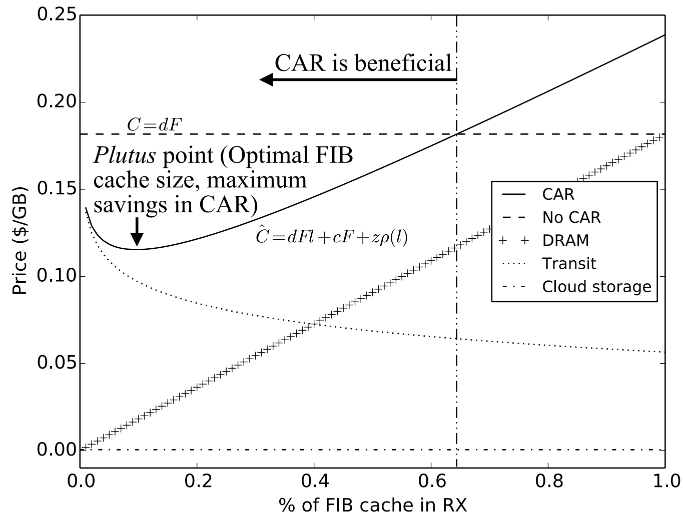


Fig. 11. CAR vs. No CAR after 15 years (at Jan 2033)

terms of the costs. Although Figure 9 plots the transit cost in $\$/Mbps$/month, for our calculation, we convert all the values to GB to match the units with cloud prices. As a parameter into this comparative analysis, we consider the percentage of FIB that needs to be stored in *RX*, i.e., the FIB cache size.

*1) **Considering No Labor Cost**:* Figure 11 reports the break-even between C and $\hat{C}$. To calculate ρ value using Eq. 3, we need to know the total traffic amount T, ψ and ω. From CAIDA Anonymized Internet Traces and Chicago (dirA) trace statistics [68] we identified the values for ψ and ω as 0.064 and 0.78 consecutively. Furthermore, we have assumed transit fee is charged for 20% of the total incoming traffic (2.31 Gb/s) towards *RX*. We shall discuss this assumption in detail and shall explore the feasibility of relaxing it, later in Section VI. For now, we see from the graph that $\hat{C}$ is lower than C when FIB cache size is zero, i.e, if we store the entire FIB table in the cloud and delegate all data traffic to *PRX* for processing, CAR is economically beneficial than No CAR. We observe that $\hat{C}$ continues to go down as FIB cache size increases. This is because $\hat{C}$ includes transit price for the delegated traffic and as *RX* stores more entry in FIB, fewer packets need delegation which, effectively, minimizes the total cost. After a certain point, $\hat{C}$ changes the direction and starts to climb up as FIB cache size increases. We name this turning point as *Plutus* after the Greek God of wealth as this marks the maximum profit for CAR. *Plutus* point does not only indicate the optimal FIB cache size, it also implies the maximum profit in CAR. As *point of tangent* alludes an (x,y) co-ordinate in geometry, *Plutus* point denotes the tuple of the optimal FIB cache size and the maximum profit. According to the graph, keeping around 10% of the entire FIB in *RX* will ensure the *Plutus* point.

The fact that we need more DRAM to store more FIB entries explains the increase in $\hat{C}$ after the *Plutus* point. Even though *RX* will be able to handle most of the traffic by itself and reduce the transit cost, DRAM price trumps the other variables in $\hat{C}$ and cause the rise up. In the meantime, C maintains the constant value as it is independent of the FIB cache size. The cross-over between these two models happen when FIB cache

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2947030, IEEE Transactions on Network and Service Management

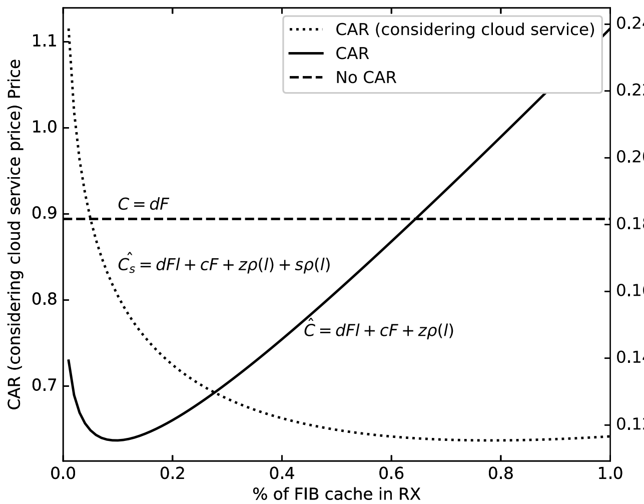


Fig. 12. CAR with storage and service price vs. No CAR

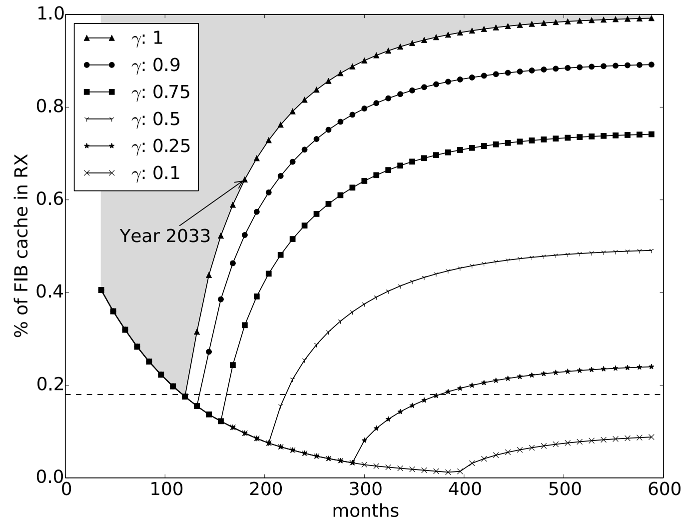


Fig. 13. CAR savings compared to legacy routing with no CAR in future

size is 64%, which means, $\hat{C}$ is more economical as long as we keep the FIB cache size less than or equal to around 64% of the total FIB size.

2) ***Considering Labor Cost:*** According to Newnan's engineering economic cost discussion [63] and Gartner [69], for a five-year life-cycle, the maintenance/support cost may supersede the initial setup cost. So, the engineering/labor cost for physical router maintenance may not be negligible. Though most of these data are business proprietary information, we can still say the labor costs will likely have a sizable impact on the router service pricing. We compare our proposed system, CAR, with an unrealistic approach to prove the cost-effectiveness. We assume zero maintenance cost for traditional routing, while CAR has both cloud service and storage cost. We think this comparison will help us to understand how beneficial CAR model will be even if we consider it in an uneven condition. Figure 12 considers three cases.

i) Traditional routing cost, (DRAM cost only)
ii) CAR cost (only cloud storage expense), and
iii) CAR cost, considering both cloud storage and service cost. For expressing CAR cost with service cost, we revise Eq. 2 as follows:

$$\hat{C}_s = dFl + cF + z\rho(l) + s\rho(l) \qquad \text{(Eq. 8)}$$

According to the plots portrayed in Figure 12, increasing the FIB cache size will be more profitable for $\hat{C}_s$ (case-iii) while it is not certainly the case for $\hat{C}$ (case-ii). This may seem contradictory to each other, but it is not. With FIB cache size reduction, we essentially delegate more packets to the clouds and thus injecting the additional cloud service cost.

Although this comparison does not show a break-even between $\hat{C}_s$ and C, we would like to emphasize that our cost model for C does not consider the recurring costs involved in traditional routing (e.g., human-labor and daily maintenance), which cannot be amortized over time. A fair comparison would require consideration of the labor and management costs of traditional routing and CAR's centralization benefits from possible management of multiple *RX*s being managed by the same cloud provider. Future work could explore these additional parameters involving inter-domain routing and business aspects spanning multiple autonomous systems.

B. CAR Savings

So far, we have explained how much savings CAR can offer at any specific time in future. We have extended our work in Figure 13 to determine how much buffer with respect to FIB cache size reduction a CAR engineer can enjoy over a period of time. Here, the scale factor represents the percentage of CAR savings over traditional routing, i.e., scale factor 1 means, there is no extra saving and both C and $\hat{C}$ are exactly the same (break-even). Assuming γ be the scale factor, we state the relation as

$$\hat{C} = \gamma C \qquad \text{(Eq. 9)}$$

Each point in individual curves represents the monthly break-even point (with a certain scale-factor) in future. For example, without any extra savings ($\gamma = 1$), after 180 months, graph plots 0.64 as break-even point for l (marked in Figure as the *Year 2033*), which is exactly what we have seen in Figure 11. This means, referring to our earlier detailed discussion, storing less than 64% of FIB in *RX* will be profitable for $\hat{C}$ in year 2033. The lower we set the γ at (to gain higher $\hat{C}$ savings), the more stringent FIB limitation is set for that specific month. In our analysis, we varied γ from 0.1 to 1 so that the impact of scale factor on maximum and minimum cost savings for CAR can be clearly demonstrated.

Another observation from here is that all the curves are concave and reach to their individual maxima. We shall discuss about these highest achievable values, the *threshold*, for each curve, and mathematically calculate them later on. FIB cache size can not be reduced beyond that threshold for a specific scale factor to achieve further savings.

Finally, we want to emphasize on the shaded area of the figure (beyond $\gamma = 1$ curve), anything on this area is not profitable for CAR at all and engineers are advised to plan accordingly for finding a suitable FIB cache size to serve their own purpose. The dotted line represents 0.18 in Y-axis of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2947030, IEEE Transactions on Network and Service Management

graph and this (18%) is the minimum FIB cache size at break-even point for $\gamma = 1$. Anything below this FIB cache size in *RX* will make CAR cheaper than traditional routing.

Eq. 9 brings us to the following remarks.

Remark 1: In order to achieve a long-term scaling factor of γ for fixed total traffic T arriving at RX, the FIB cache size l should be set to γ as long as FIB table size F monotonically increases:

$$\lim_{t\to\infty} l(t) = \gamma \qquad \text{(Eq. 10)}$$

Proof: Using C and $\hat{C}$ expressions from Eq. 1 and Eq. 2, we can re-write Eq. 9 as following:

$$d(t)Fl + c(t)F + z(t)(1 - \psi \ln(l) - \omega)T = \gamma d(t)F$$

Equating it for γ, we get:

$$\gamma = \frac{d(t)Fl + c(t)F + z(t)(1 - \psi \ln(l) - \omega)T}{d(t)F} \qquad \text{(Eq. 11a)}$$

Now substituting the values from Eq. 4, Eq. 5, Eq. 7,

$$= l + \frac{0.12e^{-0.03t}F + 1397.19e^{-0.04t}(1 - \psi \ln(l) - \omega)T}{1072118.81e^{-0.03t}F} \qquad \text{(Eq. 11b)}$$

$$= l + \frac{0.12}{1072118.81} + \frac{1397.19}{1072118.81}\frac{T}{F}(1 - \omega - \psi \ln(l))e^{-0.01t} \qquad \text{(Eq. 11c)}$$

Second term in the above equation will be very negligible and ignoring this value we get,

$$= l + [-0.0013\frac{T}{F}\psi \ln(l) + 0.0013\frac{T}{F}(1 - \omega)]e^{-0.01t} \qquad \text{(Eq. 11d)}$$

Since, $\lim_{t\to\infty} e^{-0.01t} = 0$, we finally get:

$$\gamma = l$$

This means, in future, if total incoming traffic towards *RX* remains constant, FIB cache size in *RX* and the CAR savings will have a linear relationship between them. For instance, if we want CAR cost to be 50% cheaper than that of traditional routing, storing 50% of full FIB as cache would be enough to achieve the savings.

Remark 2: As long as the ratio of the total incoming traffic T and full FIB size F does not increase more than 12.75% per year, Eq. 10 will be always true.

Proof: See Appendix A.

VI. Peering Influence

Throughout our analysis, we assumed that 20% of the total traffic will incur transit cost and the rest will be transmitted using either public or private peering. This is based on the fact that Cloudflare, a prominent content-delivery network (CDN) that operates 122 data centers across 58 countries around the globe, observed a significant shift towards peering from the year 2014 to 2016 and expected this trend to grow even more. According to them, 40% of their traffic goes through peered network relationship in North America, which is the lowest in peering as they observed, while Europe and Asia have 60% and Africa has 90% peered traffic [70]. One question that arises is how CAR and No CAR costs will change as future trends of peering increases.

Figure 14a plots multiple $\hat{C}$ graphs for varying amount of transit traffic in the year 2033. It identifies the *Plutus* points, optimal FIB cache sizes and associated maximum profits, for individual graphs as well. The higher the transit percentage is, the more *Plutus* point is shifted towards the right, meaning less traffic delegation to *PRX* will be economically beneficial which is self-explanatory as more transit cost will be charged for more traffic to *PRX*.

To get a better understanding on the relationship between transit traffic amount and its associated $\hat{C}$ values at *Plutus* point, we vary the transit percentage from 0 to 100 and charge that corresponding amount of traffic for transit to calculate $\hat{C}$. Observing the behavior of this relationship presented in Figure 14b we can, conservatively, claim that if ISPs continue to peer more and carry traffic among themselves without charging extra, owning a smaller FIB cache will be sufficient for CAR providers to offer cheaper routing services. As the amount of transit traffic rises up, FIB cache size increases almost linearly, yet, CAR is capable of saving at least 50% of FIB at *Plutus* point when there is no peering and the entire data traffic needs to pay the transit cost.

A. On Plutus Point

To evaluate the peering influence on *Plutus* point at any given time, we introduce a new variable, λ that represents the percentage of the delegated traffic that will require transit cost.

Proposition 1: $\forall \lambda \in \mathbb{R} \mid 0 < \lambda < 1$, the optimal FIB cache size l^* at the *Plutus* point is

$$l^* = \frac{\lambda \psi z T}{dF} \qquad \text{(Eq. 12)}$$

Proof: Taking the first order derivative of Eq. 2 with respect to l and equating it to zero, we get

$$\frac{d}{dl}[dFl + cF + z\rho(l)] = 0 \qquad \text{(Eq. 13a)}$$

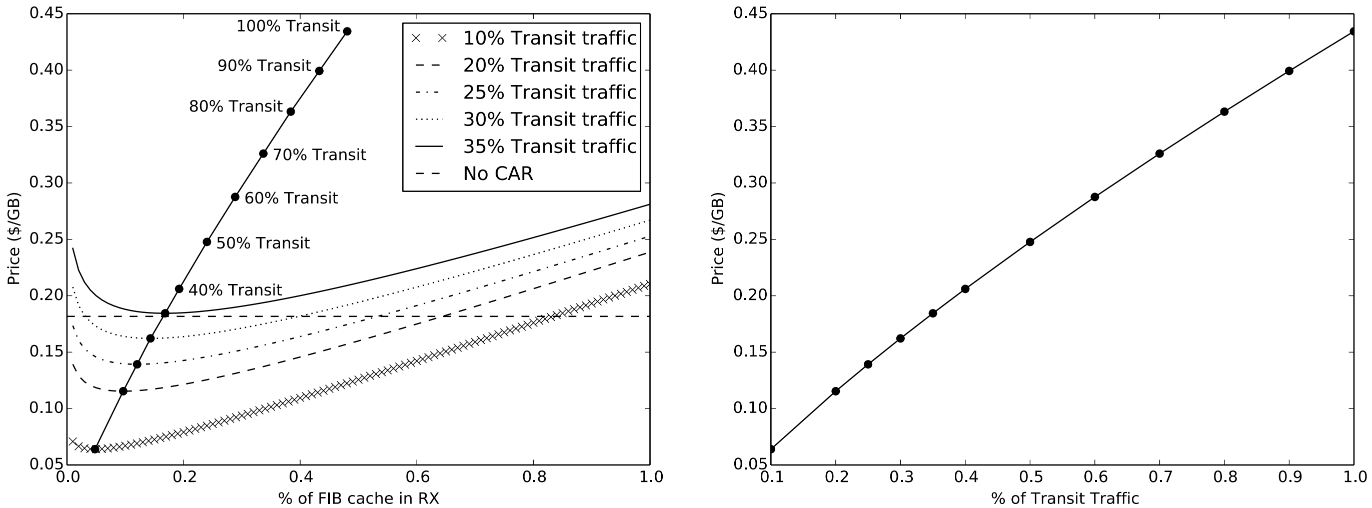
Substituting $\rho(l)$ with $\lambda\rho(l)$ and using Eq. 3,

$$dF + \lambda z \frac{d}{dl}[T - \psi T \ln(l) - \omega T] = 0 \qquad \text{(Eq. 13b)}$$

Hence, to formulate the equation for l^*, we re-write Eq. 13b in the following closed-form expression:

$$l^* = \frac{\lambda \psi z T}{dF}$$

Our observations from Eq. 12 are multifold. First, the ratio of transit cost z and DRAM cost d converges to zero if transit cost keeps declining. If that is the case, then l^* will get closer to zero as well. Second, based on how peering relationship between the ISPs evolve in future, it is possible that transit percentage λ may also increase or decrease. Third, as total traffic T continues to grow, with same λ value, larger l^* will be obtained and force the architecture to ensure a modest FIB cache size to mitigate the traffic increase impact. However, as



(a) At varying FIB cache size

(b) At Plutus Point based on Transit traffic amount

Fig. 14. CAR pricing

T keeps increasing, bigger full FIB will be required to store the entire BGP table and the ratio of total traffic T and FIB size F will play an important role to determine how CAR cost grows over time. Finally, with λ increasing, $\hat{C}$ at *Plutus* point is also increased and after a certain value of λ, $\hat{C}$ exceeds C (see in Figure 14a for *35% Transit* curve), and CAR becomes no longer economically beneficial beyond that point.

B. On Optimal CAR Design

Eq. 2 calculates the CAR cost for any FIB cache size. By using l^* instead of l, we can get the CAR cost when optimal FIB cache size is judicially picked by the CAR designer. This gives us a chance to observe the peering influence at optimal CAR design.

Proposition 2: If ξ_1, ξ_2, ξ_3 are three positive constants and f is a logarithmic function of transit percentage λ, CAR cost at optimal FIB cache size $\hat{C}(l^*)$ will be:

$$\hat{C}(l^*) = \lambda\xi_1 - \lambda\xi_2 f(\lambda) + \xi_3 \qquad \text{(Eq. 14)}$$

Proof: See Appendix B.

Now, according to Eq. 14, CAR providers can consider $\hat{C}(l^*)$ as a function of λ (transit percentage) alone, and use this condition to maximize their profit. Next, by setting λ values to the extreme, we get the boundary conditions for $\hat{C}(l^*)$.

Proposition 3: In a hypothetical environment in future where ISPs are peering heavily with each-other (i.e., $\lambda \to 0$), CAR cost at optimal FIB cache size $\hat{C}(l^*)$ will be the direct product of cloud storage cost c and the total FIB size F.

$$\hat{C}(l^*) = cF \qquad \text{(Eq. 15)}$$

Proof: See Appendix C.

This equation is independent of DRAM cost d and exactly resembles with Eq. 1 except for the fact that, d has been replaced with c (cloud-storage cost). This forecasts for a future where ISPs, in a completely peered environment among themselves, will be able to store the full FIB table in the cloud and delegate the entire traffic to cloud without any extra charge since there will be no transit cost involved at all. If this actually happens, then cloud providers can extend their footprint into routing business aggressively by emerging themselves as new candidates for ISP market and eliminate the existing ones totally or embrace CAR architecture and form partnerships to progress in a more conventional way.

Proposition 4: For light peering (i.e., $\lambda \to 1$), if transit price z does not continue to drop, delegation to the cloud will cost additional charge, which in turn will mandate CAR providers to deal with the incoming traffic locally instead of collaborating with the cloud. In such a case, the CAR cost at optimal FIB cache size will be:

$$\hat{C}(l^*) = zT[1 - \psi \ln(\frac{\psi zT}{dF}) - \omega] - zT\psi + cF \qquad \text{(Eq. 16)}$$

Proof: See Appendix D.

As T (total traffic through a router) increases, which is expected to be, ln function gives larger value and can even produce ∞, mathematically. However, based on processing speed, queue size and consumption of power, we can safely ignore this possibility as every router will have its own threshold limit and an electronic device can not perform indefinitely.

The multiplier of zT, in the first term of Eq. 16, is ρ (see Eq. 3), and the maximum value of it can be 1, as a router can not delegate more traffic than it actually receives. To make this possible, ψ value can never be equal to zero for two reasons. First, potential heavy hitters (popular prefixes) will always exist in the router to exhibit temporal dynamics and thus preventing ψ from being zero; and second, even for a capricious router with its arbitrary list of prefixes, if ψ becomes zero, ln value will be *undefined* and the entire equation will become indeterminate.

For any $0 < \psi < 1$ value, $zT > zT\psi$ will be always true. However, zT and $zT\psi$ will be very close to each other unless ψ becomes exceptionally small, which will be a rare phenomenon for the Internet. Now, if z does not continue to drop as we have seen in Section IV-D, CAR providers have to face a harsh environment where they are bound to pay for every data traffic they delegate since there is zero peering.

Finally, the minimum value of ρ can be zero if router does not delegate any traffic at all considering the transit situation. This is the worst case scenario for CAR architecture where *RX* avoids delegating towards *PRX*.

VII. Summary and Future Work

In this paper, we have introduced a new hybrid approach of SDN that leverages the computational power of the cloud and keeps the intelligence of router to some extent for reducing the FIB size and eventually offer monetary benefits to ISPs. Our primary interest is to show how much FIB reduction CAR can offer since DRAM price is not consistent enough to rely on in present condition. We have presented two cost models: one for traditional routing and the other for CAR, in which we have included the related variables that impact cost for each service. We have also shown the trends for these associated variables separately and tried to predict how they will behave in future, in order to equip the Internet providers with a better understanding of the nature of these variables.

We have shown that cloud storage is cheaper than DRAM price and transit cost is following an almost consistent decay. We then compare the economic viability of CAR with respect to traditional routing by finding a break-even between the two cost models. Although we initially focus only on storage cost, we have also considered the cloud service cost to replicate labor cost (as in traditional routing) and observed its effects on the break-even points. Later we demonstrate how much savings CAR can offer, regarding FIB cache size, in future by using scale factor against traditional routing and we have found that it is not possible to achieve unlimited savings for any given scale factor.

Finally, we have considered the Internet peering impact on our proposed architecture as it is specifically important for CAR providers to know how much luxury they can afford in delegating traffic to the cloud via paid transit. We showed an example scenario for the year 2033, where at least 65% of traffic needs to be routed through a peered network so that adopting CAR will be economical. Our analysis indicates that, with heavy peering, cloud providers will have an authoritative say in setting the market price that may bring some turbulence in ISP business. In contrast, with light peering, transit cost needs to be very small, otherwise, CAR providers will not be interested in delegating the packets to the cloud, and instead, will store the entire FIB table locally.

We believe there is a considerable scope for further research in this area and some of the key research questions include developing a failure-resilient architecture, and intra- and inter-cloud optimization using multiple *RXs* and *PRX*s. Regarding future works on economic analysis, developing a cost model for labor cost in traditional routing and to develop a model for future traffic should get priority. A more detailed private data (e.g., pricing of custom designs and services of DRAM and cloud) would enable development of more sophisticated pricing models. Further, a thorough modeling of multi-level caching of routing functions and exploration of how CAR solutions may impact an ISP's CapEx and OpEx in real world settings is needed. We plan to develop a simple prototype of CAR architecture using Quagga router and connecting it with public cloud services to observe the feasibility of the overall design. Finally, the tradeoff between the offloading cost and the performance degradation due to the packets being delegated to the cloud needs to be treated diligently.

Appendix A
Proof of Remark 2

We can re-write Eq. 11d as:

$$\gamma = l + [-0.0013\psi \ln(l) + 0.0013(1-\omega)]e^{-0.01t}\frac{T}{F} \quad \text{(Eq. 17)}$$

To maintain Eq. 10, the following must be true.

$$\frac{T}{F} < e^{0.01t}$$

Following table includes some threshold values that $\frac{T}{F}$ can obtain after a certain period of time to maximize CAR profit.

t (in months)	1	12	24	36	48
$e^{0.01t}$	1.01	1.1275	1.2713	1.4333	1.616

TABLE II
Thresholds for $\frac{T}{F}$ ratio

Appendix B
Proof of Proposition 2

To obtain the optimal CAR design case, we substitute l^* to Eq. 2:

$$\hat{C}(l^*) = dFl^* + cF + z\rho(l^*) \quad \text{(Eq. 18a)}$$

Substituting $\rho(l)$ with $\lambda\rho(l)$ and using Eq. 3,

$$\hat{C}(l^*) = dFl^* + cF + \lambda z[1 - \psi \ln(l^*) - \omega]T \quad \text{(Eq. 18b)}$$

Using l^* from Eq. 12, we get:

$$\begin{aligned}
\hat{C}(l^*) &= \lambda\psi zT + cF + \lambda z[1 - \psi \ln(\frac{\lambda\psi zT}{dF}) - \omega]T && \text{(Eq. 18c)}\\
&= \lambda zT[1 + \psi - \omega] - \lambda zT\psi \ln(\frac{\lambda\psi zT}{dF}) + cF && \text{(Eq. 18d)}\\
&= \lambda\xi_1 - \lambda\xi_2 f(\lambda) + \xi_3 && \text{(Eq. 18e)}
\end{aligned}$$

Here, ξ_1, ξ_2, ξ_3 are constants while f is a function of λ and can be interpreted as following:

$$\begin{aligned}
\xi_1 &= zT[1 + \psi - \omega] && \text{(Eq. 18f)}\\
\xi_2 &= zT\psi && \text{(Eq. 18g)}\\
\xi_3 &= cF && \text{(Eq. 18h)}\\
f(\lambda) &= \ln(\frac{\lambda\psi zT}{dF}) && \text{(Eq. 18i)}
\end{aligned}$$

APPENDIX C
PROOF OF PROPOSITION 3

Using $\lambda = 0$ in Eq. 18d,

$$\hat{C}(l^*) = \xi_3 \quad \text{(Eq. 19a)}$$
$$= cF \quad \text{[From Eq. 18h]} \quad \text{(Eq. 19b)}$$

APPENDIX D
PROOF OF PROPOSITION 4

Using $\lambda = 1$ in Eq. 18d,

$$\hat{C}(l^*) = zT - zT\psi - zT\omega - zT\psi \ln(\frac{\psi zT}{dF}) + cF$$
$$= zT[1 - \psi \ln(\frac{\psi zT}{dF}) - \omega] - zT\psi + cF$$

ACKNOWLEDGMENT

The work presented here was supported in part by NSF grant 1647189 and 1321069.

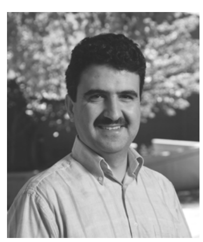
REFERENCES

[1] P. K. Dey and M. Yuksel, "On the break-even point between cloud-assisted and legacy routing (short paper)," in *Cloud Networking (Cloudnet), 2016 5th IEEE International Conference on*. IEEE, 2016, pp. 154–157.
[2] (2017, Dec.) Internet World Stats. [Online]. Available: http://www.internetworldstats.com/stats.htm
[3] (2017, Sep.) Cisco Visual Networking Index: Forecast and Methodology, 2016-2021. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html
[4] (2014, Aug.) CAT 6500 and 7600 Series Routers and Switches TCAM Allocation Adjustment Procedures. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/switches/catalyst-6500-series-switches/117712-problemsolution-cat6500-00.html
[5] (2018, Jan.) BGP Routing Table Analysis Reports. [Online]. Available: https://bgp.potaroo.net
[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
[7] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in software-defined networking (sdn)," *Computer Networks*, vol. 112, pp. 279–293, 2017.
[8] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an elastic distributed sdn controller," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 7–12.
[9] A. Iyer, P. Kumar, and V. Mann, "Avalanche: Data center multicast using software defined networking," in *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*. IEEE, 2014, pp. 1–8.
[10] T. Koponen, K. Amidon, P. Balland, M. Casado, A. Chanda, B. Fulton, I. Ganichev, J. Gross, P. Ingram, E. J. Jackson *et al.*, "Network virtualization in multi-tenant datacenters." in *NSDI*, vol. 14, 2014, pp. 203–216.
[11] O. Michel and E. Keller, "Sdn in wide-area networks: A survey," in *Software Defined Systems (SDS), 2017 Fourth International Conference on*. IEEE, 2017, pp. 37–42.
[12] H. He, K. Xu, and Y. Liu, "Internet resource pricing models, mechanisms, and methods," *Networking Science*, vol. 1, no. 1, pp. 48–66, 2012.
[13] M. M. Waldrop, "The chips are down for Moore's law." *Nature*, vol. 530, no. 7589, pp. 144–147, 2016.
[14] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica *et al.*, "Above the clouds: A berkeley view of cloud computing," Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Tech. Rep., 2009.
[15] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.
[16] What is Software-Defined WAN (or SD-WAN or SDWAN)? [Online]. Available: https://www.juniper.net/us/en/products-services/what-is/sd-wan/
[17] J. Hodges, "SD-WAN Implementation & Differentiation Layer Strategies," Heavy Reading | White Paper, Feb. 2017. [Online]. Available: https://www.juniper.net/assets/us/en/local/pdf/whitepapers/2000666-en.pdf
[18] (2016, Jan.) Pricing the Cloud 2016 Edition - 2016 to 2020. [Online]. Available: https://www.prnewswire.com/news-releases/pricing-the-cloud-2016-edition---2016-to-2020-300206954.html
[19] Y. Mor. (2015, Nov.) Analyzing AWS EC2 Price Drops over the Past 5 Years. [Online]. Available: https://www.cloudyn.com/blog/analyzing-aws-ec2-price-drops-over-the-past-5-years/
[20] P. K. Dey and M. Yuksel, "CAR: Cloud-assisted routing," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE Conference on*. IEEE, 2016, pp. 100–106.
[21] (2015, Jun.) Memory Troubleshooting Guide for Cisco 4000 Series ISRs. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/routers/access/4400/troubleshooting/memorytroubleshooting/isr4000_mem.html#pgfId-1080950
[22] T. Yang, G. Xie, Y. Li, Q. Fu, A. X. Liu, Q. Li, and L. Mathy, "Guarantee ip lookup performance with fib explosion," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 39–50.
[23] Z. A. Uzmi, M. Nebel, A. Tariq, S. Jawad, R. Chen, A. Shaikh, J. Wang, and P. Francis, "Smalta: practical and near-optimal fib aggregation," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*. ACM, 2011, p. 29.
[24] Y. Liu, B. Zhang, and L. Wang, "Fifa: Fast incremental fib aggregation," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1–9.
[25] X. Zhao, Y. Liu, L. Wang, and B. Zhang, "On the aggregatability of router forwarding tables," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.
[26] W. Eatherton, G. Varghese, and Z. Dittia, "Tree bitmap: hardware/software ip lookups with incremental updates," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 97–122, 2004.
[27] G. Varghese, *Network Algorithmics: an interdisciplinary approach to designing fast networked devices*. Morgan Kaufmann, 2005.
[28] M. Bando and H. J. Chao, "Flashtrie: hash-based prefix-compressed trie for ip route lookup beyond 100gbps," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.
[29] H. Asai and Y. Ohara, "Poptrie: A compressed trie with population count for fast and scalable software ip routing table lookup," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 57–70.
[30] G. Rétvári, J. Tapolcai, A. Kőrösi, A. Majdán, and Z. Heszberger, "Compressing ip forwarding tables: towards entropy bounds and beyond," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 111–122.
[31] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of internet path selection," in *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4. ACM, 1999, pp. 289–299.
[32] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating internet routing instabilities," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 205–218, 2004.
[33] R. T. Ma, D. M. Chiu, J. Lui, V. Misra, and D. Rubenstein, "Internet economics: The use of shapley value for isp settlement," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 3, pp. 775–787, 2010.
[34] A. Araldo, D. Rossi, and F. Martignon, "Cost-aware caching: Caching more (costly items) for less (isps operational expenditures)," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1316–1330, 2016.
[35] M. Motiwala, A. Dhamdhere, N. Feamster, and A. Lakhina, "Towards a cost model for network traffic," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 1, pp. 54–60, 2012.
[36] (2017, Nov.) Comparing Cloud Instance Pricing: AWS vs Azure vs Google vs IBM. [Online]. Available: https://www.rightscale.com/blog/cloud-cost-analysis/comparing-cloud-instance-pricing-aws-vs-azure-vs-google-vs-ibm
[37] (2017, Dec.) Is the cheapest cloud pricing flexible or granular? [Online]. Available: https://www.cio.com/article/3243092/cloud-computing/is-the-cheapest-cloud-pricing-flexible-or-granular.html
[38] S. Ragner. (2016, Nov.) The cloud price war continues: Amazon cuts its cloud storage prices, again. [Online]. Available: http://www.zdnet.com/article/the-cloud-price-war-continues-amazon-cuts-its-cloud-storage-prices-again/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2947030, IEEE Transactions on Network and Service Management

[39] J. Barr. (2017, Dec.) Amazon EC2 Price Reduction in the Asia Pacific (Mumbai) Region. [Online]. Available: https://aws.amazon.com/blogs/aws/category/price-reduction/
[40] (2016, Feb.) Completing the Netflix Cloud Migration. [Online]. Available: https://media.netflix.com/en/company-blog/completing-the-netflix-cloud-migration
[41] (2018) Airbnb Case Study. [Online]. Available: https://aws.amazon.com/solutions/case-studies/airbnb/
[42] (2018) Adobe Case Study. [Online]. Available: https://aws.amazon.com/solutions/case-studies/adobe/
[43] M. Weinberger. (2017, Feb.) Snap will pay Google at least $400 million a year for the next 5 years. [Online]. Available: http://www.businessinsider.com/google-cloud-platform-wins-snapchat-2017-2
[44] (2018) Best Buy Slashes App Development Time and Resources with Google App Engine. [Online]. Available: https://cloud.google.com/customers/best-buy/
[45] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
[46] H. Wang, Q. Jing, B. He, Z. Qian, and L. Zhou, "Distributed systems meet economics: pricing in the cloud," 2010.
[47] P. C. Fishburn and A. M. Odlyzko, "Dynamic behavior of differential pricing and quality of service options for the internet," *Decision Support Systems*, vol. 28, no. 1, pp. 123–136, 2000.
[48] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, C. N. Corrêa, S. C. de Lucena, and M. F. Magalhães, "Virtual routers as a service: the routeflow approach leveraging software-defined networks," in *Proceedings of the 6th International Conference on Future Internet Technologies*. ACM, 2011, pp. 34–37.
[49] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: comparing public cloud providers," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 1–14.
[50] S. K. Garg, S. Versteeg, and R. Buyya, "Smicloud: A framework for comparing and ranking cloud services," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. IEEE, 2011, pp. 210–218.
[51] S. K. Barker and P. Shenoy, "Empirical evaluation of latency-sensitive application performance in the cloud," in *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*. ACM, 2010, pp. 35–46.
[52] K. Jang, J. Sherry, H. Ballani, and T. Moncaster, "Silo: Predictable message latency in the cloud," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 435–448.
[53] (2018) AWS (Amazon Web Services) vs Microsoft Azure vs Google Cloud Platform. [Online]. Available: https://www.cloudorado.com/vs/aws_vs_azure_vs_google
[54] P. Brogan, "Broadband Investment Continues Trending Down in 2016," Oct. 2017, USTELECOM (The Broadband Association). [Online]. Available: https://www.ustelecom.org/sites/default/files/documents/Broadband%20Investment%20Trending%20Down%20in%202016.pdf
[55] CAIDA. (2018) The CAIDA UCSD Anonymized Internet Traces - 20181220. [Online]. Available: http://www.caida.org/data/passive/passive_dataset.xml
[56] C. Kim, M. Caesar, A. Gerber, and J. Rexford, "Revisiting route caching: The world should be flat." in *PAM*, vol. 9. Springer, 2009, pp. 3–12.
[57] Y. Liu, S. O. Amin, and L. Wang, "Efficient fib caching using minimal non-overlapping prefixes," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 1, pp. 14–21, 2013.
[58] J. C. McCallum. (2018) Memory Prices (1957-2017). [Online]. Available: http://jcmit.net/memoryprice.htm
[59] (2017, Nov.) [Market View] TrendForce Reports 3Q17 DRAM Revenue Hit a New High With a 16.2% Sequential Increase; DRAM Prices to Rise by 10% on Average in 4Q17. [Online]. Available: https://www.dramexchange.com/WeeklyResearch/Post/2/4828.html
[60] R. Bala, A. Chandrasekaran, and J. McArthur. (2017, Jul.) Magic Quadrant for Public Cloud Storage Services, Worldwide. [Online]. Available: https://www.gartner.com/doc/reprints?id=1-47K8W1K&ct=170725&st=sb
[61] (2013, Jan.) Top tips for doing a cloud storage cost analysis. [Online]. Available: http://searchstorage.techtarget.com/tip/Top-tips-for-doing-a-cloud-storage-cost-analysis
[62] G. Budman. (2017, Apr.) Backblaze B2 Drops Download Price By 60%. [Online]. Available: https://www.backblaze.com/blog/b2-drops-download-pricing/
[63] D. G. Newnan, T. Eschenbach, and J. P. Lavelle, *Engineering economic analysis*. Oxford University Press, 2004, vol. 2.
[64] E. Kim. (2014, Jul.) Amazon Web Services Had 44 Price Cuts And You Can Expect More To Come. [Online]. Available: http://www.businessinsider.com/amazon-web-services-had-44-price-cuts-2014-7
[65] W. B. Norton, "Internet transit prices-historical and projected," *Report, DRPeering International, Aug*, 2010. [Online]. Available: http://drpeering.net/white-papers/Internet-Transit-Pricing-Historical-And-Projected.php
[66] (2016, Jul.) IP Transit pricing summary (Flat-rate 1Gbps-on-GigE pricing: $380/mo (3-year term)). [Online]. Available: http://henet.p2knowledgebase.com/view/blog/price-reductions-hurricane-electric-ip-transit-network-bandwidth-services_35
[67] (2017, Aug.) IP Transit pricing summary (Flat-rate 1Gbps-on-GigE pricing: $380/mo (3-year term)). [Online]. Available: http://henet.p2knowledgebase.com/view/short_post/WnL8B
[68] (2016, Apr.) Trace Statistics for CAIDA Passive OC48 and OC192 Traces. [Online]. Available: http://www.caida.org/data/passive/trace_stats/
[69] A. Lerner. (2014, May) Save Money on Networking Support/Maintenance Costs. [Online]. Available: https://blogs.gartner.com/andrew-lerner/2014/05/15/sweetmoolahwithoutunclericoh/
[70] N. Rao. (2016, Aug.) Bandwidth Costs Around the World. [Online]. Available: https://blog.cloudflare.com/bandwidth-costs-around-the-world/

Prasun Kanti Dey completed his B.Sc. in CSE from Bangladesh University of Engineering and Technology (BUET), Bangladesh in 2011 and M.Sc. in CSE from University of Nevada - Reno (UNR), Reno, NV in 2016. He is currently pursuing the Ph.D. degree in Computer Engineering at University of Central Florida (UCF), Orlando, FL. He was a software engineer in Samsung R&D Institute Bangladesh (SRBD) after finishing his undergraduate studies. Now, he is working with his supervisor Dr. Murat Yuksel as a research assistant in UCF. His research interests include network management and security, network measurement and economics, SDN, cloud-computing and software-design. He is a member of both ACM and IEEE. He likes to read novels, travel places and is passionate about photography.



Murat Yuksel is an Associate Professor at the ECE Department of the University of Central Florida (UCF), Orlando, FL. Prior to UCF, he was with the CSE Department of the University of Nevada - Reno (UNR), Reno, NV as a faculty member until 2016. He was with the ECSE Department of Rensselaer Polytechnic Institute (RPI), Troy, NY as a Postdoctoral Associate and a member of Adjunct Faculty until 2006. He received a B.S. degree in computer engineering of Ege University, Izmir, Turkey in 1996. He received M.S. and Ph.D. degrees in computer science of RPI in 1999 and 2002 respectively. He worked as a software engineer at Pepperdata, Sunnyvale, CA and a visiting researcher at AT&T Labs and Los Alamos National Lab. His research interests are in the area of networked, wireless, and computer systems with a recent focus on big-data networking, UAV networks, optical wireless, public safety communications, device-to-device protocols, economics of cyber-security and cyber-sharing, routing economics, network management, and network architectures. He has been on the editorial board of Computer Networks, and published more than 100 papers at peer-reviewed journals and conferences and is a co-recipient of the IEEE LANMAN 2008 Best Paper Award. He is a senior member of IEEE, senior and life member of ACM, and was a member of Sigma Xi and ASEE.