Network Hardware Virtualization for Application Provisioning in Core Networks

Ashwin Gumaste, Tamal Das, Kandarp Khandwala, and Inder Monga

The authors articulate the impact of NV on networks that provide customized services and how a provider's business can grow with NV. They outline a decision map that allows mapping of applications with technology that is supported in NV-oriented equipment. Analogies to the world of virtual machines and generic virtualization show that hardware supporting NV will facilitate new customer needs while optimizing the provider network from the cost and performance perspectives.

ABSTRACT

Service providers and vendors are moving toward a network virtualized core, whereby multiple applications would be treated on their own merit in programmable hardware. Such a network would have the advantage of being customized for user requirements and allow provisioning of next generation services that are built specifically to meet user needs. In this article, we articulate the impact of network virtualization on networks that provide customized services and how a provider's business can grow with network virtualization. We outline a decision map that allows mapping of applications with technology that is supported in network-virtualization--oriented equipment. Analogies to the world of virtual machines and generic virtualization show that hardware supporting network virtualization will facilitate new customer needs while optimizing the provider network from the cost and performance perspectives. A key conclusion of the article is that growth would yield sizable revenue when providers plan ahead in terms of supporting network-virtualization-oriented technology in their networks. To be precise, providers have to incorporate into their growth plans network elements capable of new service deployments while protecting network neutrality. A simulation study validates our NV-induced model.

INTRODUCTION

Provider revenues are growing primarily based on provisioning next generation services such as video, cloud, mobile backhaul, and data centers. Applications that dominate provider revenues are becoming aggressive in their network requirements [1]. If service providers do not reinvent themselves to meet application requirements, their revenue will decrease due to *over-the-top* (OTT) vendors capturing much of the newfound e-commerce revenue. For example, video distribution OTT vendors like Netflix, Amazon, Dropbox, and Salesforce are cashing in on raw bandwidth pipes provided by network operators, creating a constant feud between network providers and application providers. In the worst case scenario, a network provider could impede good quality service to application providers as they do not share revenues, given that the network is merely seen as a basic bandwidth pipe. This feud must be resolved for the larger sake of the ecosystem.

Another aspect of this feud is the drive to protect network neutrality (NN). Shown in [2] are multiple aspects of NN. While not throttling someone's service is a given, a more important aspect is how to create a new service that better facilitates the OTT operator while protecting NN. It is not a question of how long it would take for service providers (SPs) to support OTT services, but rather a question of how to support such a service. As elaborated in [1], it is about routing money, not packets.

This article studies the interaction between network providers and application providers through the use of network hardware virtualization. Network virtualization (NV) manifests itself as an excellent way to resolve this feud by facilitating the partitioning of the network hardware into qualitative domains that are responsible for providing specific service to the application provider. We see NV as an intermediate enabler for network functions virtualization (NFV), and in direct conjunction with SDN white-boxes.

In this article, we propose NV as an enabler toward solving the paradox between network operators and OTT application providers. Network operators reason that they have to invest in the network infrastructure, and licensing and maintaining the network, while application providers use the network and earn revenue from consumers. Customers of the network provider at times overuse the liberties provided by the network provider. Application providers, on the other hand, treat the network as a bunch of bandwidth pipes that pre-exist and do not see a reason to share their revenue. There are merits in both arguments from the perspectives of network and application providers. The deadlock needs to be resolved for both parties to maximize profit as well as serve the end user better.

This deadlock can be technically resolved by implementing NV. The idea is that by using NV in the network, a service provider c*an now customize services that suit the OTT application*. An OTT application provider now has the incentive to share revenue or buy a specific related service that better drives their application to their end user (the consumer).

The next obvious question is *how to implement NV in a network operator*. We begin by understanding application requirements at a broad level and mapping them to possible capabilities of networks to offer customized services. Webb

Digital Object Identifier: 10.1109/MCOM.2017.1500488CM

Ashwin Gumaste and Tamal Das are with the Indian Institute of Technology Bombay; Kandarp Khandwala is with the University of California San Diego; Inder Monga is with Lawrence Berkeley National Laboratory.

et al. [3] described ways in which an application can communicate to the network in terms of customization required for a particular application. However, rather than real-time application-level changes, most OTTs have specific and well-known requirements from the network [4]. *So can we model a network based on such requirements, mapping these requirements to NV partitions?*

To do so, we first explore if it indeed is feasible to model OTT requirements over an SP network, by isolating key services that would have strong business cases for implementing NV, and have key requirements that a provider can fulfill. To this end, the next section presents a table that manifests OTT requirements from the network including network technology choices [5]. For the sake of brevity, we focus only on the metro and core network, assuming that network pipes are essentially static entities with little scope for technology enhancement due to voluminous users, although with NFV even the access gear could be virtualized. Then we present a method for the application service provider (ASP) to interact with the SP and show how this can be implemented in four different technology classes, each using a software defined control plane. Following that we show how software defined networking (SDN) can be made to function in such a scenario, and the relationship between SDN and NV pertaining to the technology solutions. Finally, we capture results from a simulation model that validates our hypothesis.

DISPARATE NETWORKING REQUIREMENTS

In this section, we discuss application-level requirements of various domains and how these can be mapped to network equipment through NV. Shown in Table 1 is a list of key revenue-generating OTT services. For each service, Table 1 lists network-centric specifics desired by an ASP and plausible technology options to provision the service. Table 1 considers ASP businesses currently valued at US$1 billion plus [6]. The key driver toward ASP traffic is video. Since we ignore the access network, it is safe to say that the traffic is largely business-to-business (B2B) in nature, but can without loss of generality be extended to a business-to-consumer (B2C) model. For many of the applications, there are multiple technology solutions possible, and the ones that are commercially viable in a tier 1 provider network are listed in column 3.

The key question that Table 1 highlights is: *how can an SP provision a particular service requirement in the network?* To this end, a system must be designed that orchestrates interaction between the provider and the OTT ASP, adhering to tenets of NN. This interaction must be mapped onto network hardware so that service provisioning is possible. Our proposal is to create an SDN controller that would facilitate interaction between incoming traffic requests from ASPs mapping these onto provider hardware that adheres to NV principles. The key challenge in this approach is to *map the incoming demand into network-specific parameters that can be used for traffic engineering, bandwidth brokering, provisioning, and service support*; and *enable the network hardware to be able to provision new services with specific OTT needs.* The challenge in the latter is to be able to create services and differentiate them at the hardware layer.

Domain/OTT service	Requirement from network	Technology
Video services	Guaranteed bandwidth low jitter	IP/MPLS/WDM/CE
Mobile VAS	Unconstrained bandwidth low packet drop	MPLS/OTN/CE
Video advertising and merchandise delivery	Bandwidth on demand low jitter	MPLS/CE
Real-time events and entertainment delivery	Extreme multicast bandwidth on demand	MPLS/CE/WDM
Healthcare and tele-medicine	Low downtime, high bandwidth, security, low latency	MPLS/CE
Defense networks	Minimal downtime, low latency, security, virtualization, multicast, bandwidth	MPLS/OTN/CE/WDM
Finance and banking	Virtualization, minimal latency, security	IP/MPLS/CE/OTN
Educational networks	Multicast, high bandwidth	WDM
IT virtualization	Fast switching, resiliency	MPLS
Gaming services	Extreme interaction, multicast, low latency	MPLS/CE/IP

Table 1. Service-technology matrix.

The next section describes a solution using NV principles to partition SP hardware to meet ASP service goals. The advantage of NV is that it enables an SDN controller to realize the full potential of an SDN-centric network.

BUILDING A SOLUTION WITH VIRTUAL NETWORK EQUIPMENT PARTITION

In this section, we describe a method to implement NV to meet specific ASP requirements. We assume that a request for a service arrives into a SP domain and a *network management system* (NMS) communicates to an SDN controller that would provision services. The NMS can abstract specific requests into network-centric parameters with the goal of provisioning services. The NMS maps a service request onto an abstracted network topology by considering specific service parameters. These parameters are then mapped onto all the network elements (NEs) in the path to check service provisioning feasibility. To check feasibility, there must be a parameterized relationship between incoming service requests and the equipment deployed. The SDN controller maps an incoming request to a network-virtualized hardware. The idea is that every piece of hardware is further divided into service supporting modules that are parameter-driven and have a direct relation with an SDN controller populated flow table. Virtualization happens by the creation of multiple (virtualized) instances of the data-plane at each NE. Each such instance of the data plane enables OTT-service-specific feature implementation.

METHOD TO IMPLEMENT NV IN SP-ASP (OTT) INTERACTION

We now describe how to implement NV in a provider network. A request that enters the network is provisioned through a network interface supported by the NMS. For each new incoming request, the NMS computes the optimal network resources to be allocated. To this end, the following steps are envisaged at the centralized NMS:

- A route is computed based on service requirements. Actual bandwidth allocation is computed along the route depending on the specified request and other requests at that instance.
- Each element along the computed route is examined from a service support perspective, whether it can satisfy *specific* requirements of the service.
- To compute the specific requirements of the service request, we propose the concept of or *virtualized network equipment partitions* (VNEPs) that enables a network equipment (e.g., a switch or router) to be partitioned to satisfy specific service parameters. An example of a VNEP is provided in the next subsection.
- If VNEPs are possible along the path to provision the request, all the network equipments are provisioned to meet the new request by the NMS through the SDN controller. Otherwise, an alternate path that maximally conforms to the VNE requirement (partially, if not fully) is provisioned.
- A VNEP created at a node may be moved to another node depending on resource availability over a period of time.

VNEP computation is now described in detail.

VNEP COMPUTATION

A VNEP is represented by the virtual partitioning of hardware such that each of the partitioned elements corresponds to fully functional entities, capable of performing all the functions as the larger hardware, but specific for a service request. The key to VNEP creation is to note that the overlaid software creates partitions by allocating hardware resources within a larger NE. Partitions could be created in switching elements, network processors, buffers, and packet classifiers. Partitions correspond to hardware resources as defined by the software and are made available strictly for a particular service or function.

Our conjecture (based on an analysis of existing network gear) is that a networking element can be divided into partitions, such that a partition can act as a completely independent networking element. We argue that the sum of parts — that is, the union of all *partitions* — does not necessarily add up to the *original element* for that *particular parameter*. Throughput, average latency, and packet loss rate are examples of parameters.

Let us consider an example: Assume a 60 Gb/s switch fabric with virtual-output-queued (VOQ) buffers, with 6-input lines and 6-output lines all at 10 Gb/s. Assume one of the lines is sending data at 2 Gb/s, the average packet size is 250 bytes, and the VOQ memory to store packets for contention resolution is 3 Mb. The maximum ingress-to-egress latency is observed as 300 s. However, we aim to estimate average latency, which is a function of the provisioned services at the other 5-ingress ports, the nature of the traffic, and type of switch fabric (cut-through, store-and-forward, shared memory, etc.).

Since the latency of a flow through a switch also depends on other flows, one way to control it is to bound the number of flows through the switch. A simple 4 × 4 cross-bar with VOQ (essentially a 12 × 4) switch (each port at 1 Gb/s) can take 4 flows each with 250-byte average packet size at full line rate (wirespeed operation), resulting in 1.2 μs switching, while the same switch will result in 2.4 μs port-to-port latency if the average packet size is 128 bytes [7]. Similarly, the switch will result in a latency of 3 μs if the packet size is 64 bytes [7]. The switch behavior becomes more erratic when the standard deviation between flows across multiple ports increases [8]. For example, the switch results in a port-to-port latency of 12 s for multicast traffic if the packet size is 64 bytes, and remaining ports have provisioned flows with packet size of 1500 bytes. The above discussion highlights the complex relationship between packet sizes, port counts, traffic distribution (random/unicast/multicast), and so on, implying that for carrier-class services, that is, with desired deterministic parameters of delay and jitter, predicting switch behavior is important but difficult. Even intricate queuing models (i.e., those deploying G/G/1 queues) tend not to converge in real time.

So our approach is to provision services without getting involved in the intricacies of computing switch-specific parameters in real time. *Our approach is technology-specific, given the enormous amount of technology deployments.*

In our approach, we partition a switch/router/optical-cross-connect into VNEPs that can individually provision services. The idea is to dynamically create a VNEP that will adhere to all the system-wide parameters for a particular service, with the constraint that the sum of all the VNEPs in an NE is less than the total capacity of the switch. The union of VNEPs is not linear. This implies that the system leads to overprovisioning, which, though undesired, is necessary to maintain many of the carrier-class attributes desired for OTT services.

VNEP creation and sizing involves the following steps:

1. An NE is viewed as the number of instances q_i of a particular parameter i such that $f(qi)$ denotes the performance criteria (e.g., bounded latency) for parameter i.
2. The value $f(qi)$ also takes into consideration another parameter whose performance criteria is $f(qi)$ is the number of instances of supporting j and which impacts $f(qi)$.
3. Note that it is mathematically nontrivial to compute $f(qi)$, and hence worst case provisioning metrics are used as acceptable practices.

The second point is supported by an example. Let i denote the service parameter for port-to-port latency. Assume a 60 Gb/s switch fabric supports 6 × 10 Gb/s connections with 250 bytes average packet size and $f(qi)$ = 3 μs. The same fabric will have an $f(qi)$ = 12 μs latency for the same number of flows if the average packet size reduces to 64 bytes. The delay increases sizably ($f(qi)$ = 50) if the number of flows increases to 60 × 1 Gb/s flows. So, now if we have to provision a service of 1 Gb/s with a latency within 3 μs, and another service of 5 Gb/s with a latency also within 3 μs,

In our approach, we partition a switch/router/optical-cross-connect into VNEPs that can individually provision services. The idea is to dynamically create a VNEP that will adhere to all the system-wide parameters for a particular service, with the constraint that the sum of all the VNEPs in a NE is less than the total capacity of the switch. The union of VNEPs is not linear.

how do we do so given that the packet size of the first service is, say, 128 bytes and the second one is, say, 64 bytes? Obviously, the second service will require more overprovisioning compared to the first one, that is, that although the second service is 5× the first service, in order to achieve similar parameters, the second service may have to be provisioned through the switch with 12× resources (buffers primarily) so that the switch can meet provisioning requirements. Now how do we arrive at the number 12×? This number is a function of both volume and quality: volume, as in how much more would the service take in every parameter's domain, and quality, as in what would be the impact of the service provisioning in other parameter domains.

Shown in Fig. 1a is the actual process for creating and allocating VNEPs. From an incoming request (*Req*(*i*)), we compute the corresponding partition's impact on other partitions. The SDN controller computes VNEPs for each service at each NE. The controller then sends specific information to each node to partition itself according to its VNEP computation based on the four use cases discussed at the end of this section.

Given that there are a large number of protocols deployed leading to a variety of equipment such as IP/multiprotocol label switching (MPLS) routers, optical transport network (OTN) cross-connects, carrier Ethernet (CE) switches, and wavelength-division multiplexing (WDM) gear, a key question is how to implement partitioning. It is publicly known that many vendors are in the process of SDNizing their current gear. The question we want to answer is: *how can equipment vendors achieve network virtualization at the data plane?*

To this end, we have identified network equipment from 10 vendors who are known to be committed to SDNizing their product portfolio. These 10 vendors combined have products across the aforementioned technologies (IP/MPLS etc.). Seven of these vendors have products in the layer 2/3 (L2/L3) space, while three products are from the optical space.

On studying the equipment of these seven chipsets as well as corresponding patents, it appears the architecture follows one or a combination of the following three strategies:

- A field programmable gate array (FPGA)-based switching core or an FPGA as a processing element
- An application-specific integrated circuit (ASIC) or merchant silicon-based switching core with an FPGA or a processor guiding the ASIC
- A network processor (NP)-based switching core

In Table 2, we captured the key chipsets that are used for creation of the products for the various equipment vendors. The table also shows how the data path can be partitioned.

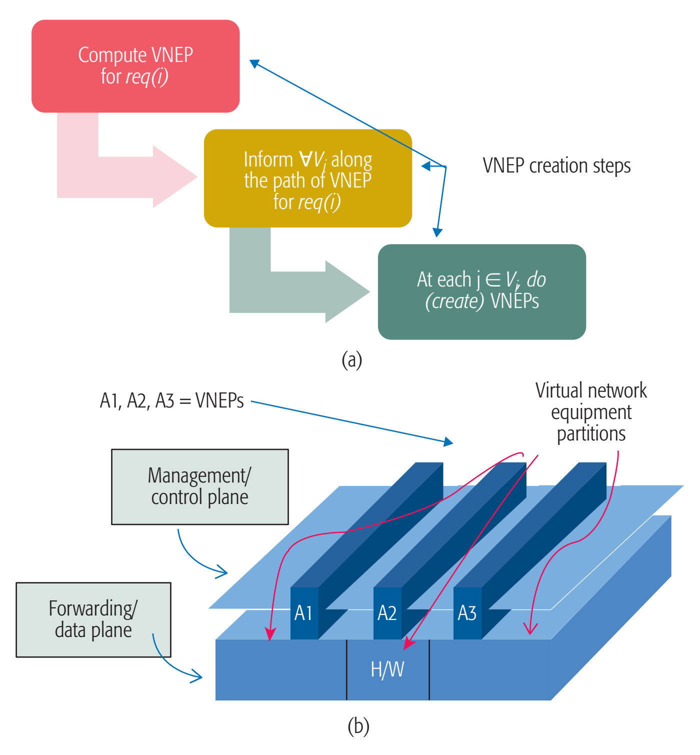

Figure 1. VNEP.

Shown in Table 2 are seven implementations of a switching plane used for L2/L3 equipment. Additionally, we have also considered three reconfigurable optical add/drop multiplexer (ROADM) implementations using liquid crystal on silicon (LCOS)-based wavelength selective switches (WSSs) of $1 \times M$ and $M \times N$ configurations and another WSS based on digital lightwave processing technology. The initial seven L2/L3 cases are shown in the table. Two types of FPGAs (FPGA 1 and 2), two types of network processors (NP 1 and 2) and three types of ASICs with FPGAs and NPs (ASICs 1–3) are compared.

The key takeaway from Table 2 is to show that irrespective of the technology deployed, it is indeed possible to create VNEPs. To this end, Table 2 showcases the sliceability parameter at what granularities can we slice a fabric. The impact of slicing is on the throughput (speed of the device) and latency. The memory capacity also has a direct impact on the throughput: the more slicing, the more memory is required; hence, latency suffers. Larger numbers of flows require either more interconnected fabrics (multicard designs) or use of large ASICs (columns 7 and 8). The latency is impacted by sliceability as well as protocol (quality of service [QoS], more processing, etc.).

VNEP Partitioning Analogous to Virtual Machine Creation and Migration: VNEP creation and NV using VNEPs is analogous to virtual machine (VM) creation and migration in hardware. Shown in Fig. 1b is the analogy of the forwarding plane in an NE with a VM hypervisor. VMs can be dynamically created in a processing environment. The same analogy is used for VNEP creation, whereby VNEPs are, like VMs, created on the fly and use the switch fabric resources independently. As in Fig. 1b, VNEPs are created by the control plane (SDN-based) and implemented within the NE through NV.

From the perspective of Table 2, we can create VNEPs as slices in different implementations of L2/L3 equipment or as independent optical switches virtually superimposed on a ROADM, as shown next.

	FPGA 1	FPGA 2	NP1	NP2	ASIC 1	ASIC 2	ASIC 3
Sliceable or not	Yes (425K logic blocks)	Yes (693K logic blocks)	Yes	Yes	Yes	Yes	Yes
Min. SW granuality IO (Mb/s)	1	≤ 1	1	0.064	0.128	1	0.064
SW granularity every component (Gb/s)	240	800	120	640	40	1280	50
How many parallel lines (10 Gb/s)	24 (16 standard 8FX)	80 (GTX)	12 × 10G OR 3 × 40G	64 × 10G and 16 40G	4 × 10G and 24 × 1G and 12 × 2.5G	12 8 × 10G	2 × 25G
Switch capacity	360 Mp/s	1600 Gb/s	≤ 2 GHz	1.25 GHz	60 Mp/s	1440 Mp/s	30 Mp/s
Average latency	400 ns	500 ns	NA	NA	NA	150–650ns	120 ~750ns
Protocol	L2/3/4	L2/3/4	L2/L3	L2/L3/L4	L2/L3	L2/L3	L2
Memory capacity	50 Mb	52 Mb	4 MB	7.5 MB	8.5 MB	12 Mb	–
Number of switching blocks	Variable	Variable	120,000	240,000	40,000	1,280,000	50,000

Table 2. VNEPs in pragmatic network elements.

USE CASES

Use Case 1: IP/MPLS-over-WDM: For IP/MPLS overlay and WDM ROADM underlay, IP/MPLS label switched routers (LSRs) are partitioned based on supported flows, and WDM ROADMs are partitioned to support non-blocking connections. VNEPs in the ROADM require support of colorless, directionless, and contentionless (CDC) as well as gridless properties. A VNEP in an LSR is an MPLS tunnel.

Use Case 2: MPLS-over-OTN with WDM: In the case of MPLS-over-OTN with WDM underlay, VNEP partitions take into consideration OTN pipes at MPLS-LSR interfaces that further feed into a WDM network. We assume services are sub-wavelength granular, implying wavelength assignment as a multi-service aggregation and provisioning problem. Partitioning happens at the LSR forwarding plane and OTN-based ODU (optical data unit) switch fabric.

Use Case 3: CE+OTN-over-WDM: In this case, we partition the CE switch fabric into discrete switching chunks so that an Ethernet switched path (ESP) is mapped onto an OTN ODU port. The VNEPs are portions of the CE switch fabric implemented.

Use Case 4: IP-over-CE+OTN-over-WDM: In this case, IP routers are at select locations as an overlay with a CE underlay, all over a ROADM-based WDM network. Whenever a service has granularity that is near a wavelength's full capacity (10/100Gb/s), it is routed all-optically by the ROADM. Whenever a service can be routed at layer 2 through the use of an ESP, it is done so using the CE network used for aggregation and switching. However, when layer 2/1 provisioning is not possible, the service is handled exclusively through the IP layer. VNEP information created by the centralized controller is used to partition switching resources at any or all of the CE/IP layers that use FPGAs/ASICs/NPs.

INTERACTION BETWEEN SDN AND NV

Figure 2 shows a switch architecture to implement SDN with NV with a controller connected to the switch's northbound interface. The switch could support L2/L3 protocols, and the interfaces would be mapped onto wavelengths. Incoming flows are segregated at the input buffers (which are further segregated to support VOQs). Flow headers are worked on by a control state machine (CSM) that also populates SDN tables. All protocol functioning happens at the controller. To support scalability, we assume that the controller runs on a VM.

The architecture in Fig. 2 can have multiple manifestations including the use of FPGAs/ASICs/NPs. In one embodiment, we assume an IP/MPLS LSR in which the CSM and SDN flow tables (SDNFTs) are implemented in an FPGA, while other modules are implemented in an ASIC. In another CE design, the SDNFT, CSM, and VOQs are implemented in an NP, while the switch fabric and memory are implemented in an ASIC. Yet another design includes a smaller CE device that has the entire design except the SDNFT in an FPGA, with the flow tables in a TCAM ASIC.

We propose the following three policies for VNEP partitioning:

Policy 1: Throughput Maximization: In this policy, VNEP computation maximizes the throughput at every NE. This is a non-carrier-class policy implying that the port-to-port latency per NE is non-deterministic. This implies an additive increase of throughput, and hence, whenever a new request arrives at the SDN control plane, a VNEP is created with a view to maximize network-wide throughput. The CSM partitions the hardware as per the specifications of Table 2.

Policy 2: Latency-Bounded Partitioning: In this policy, a VNEP is created such that the corresponding service is guaranteed to meet end-to-end latency requirement through every NE by bounding latency. This policy requires double optimization: route selection and associated appropriate amount of partitioning at a node.

Policy 3: Latency-Sensitive Service Maximization (LSSM): In this globally active policy, the approach is to maximize the number of services

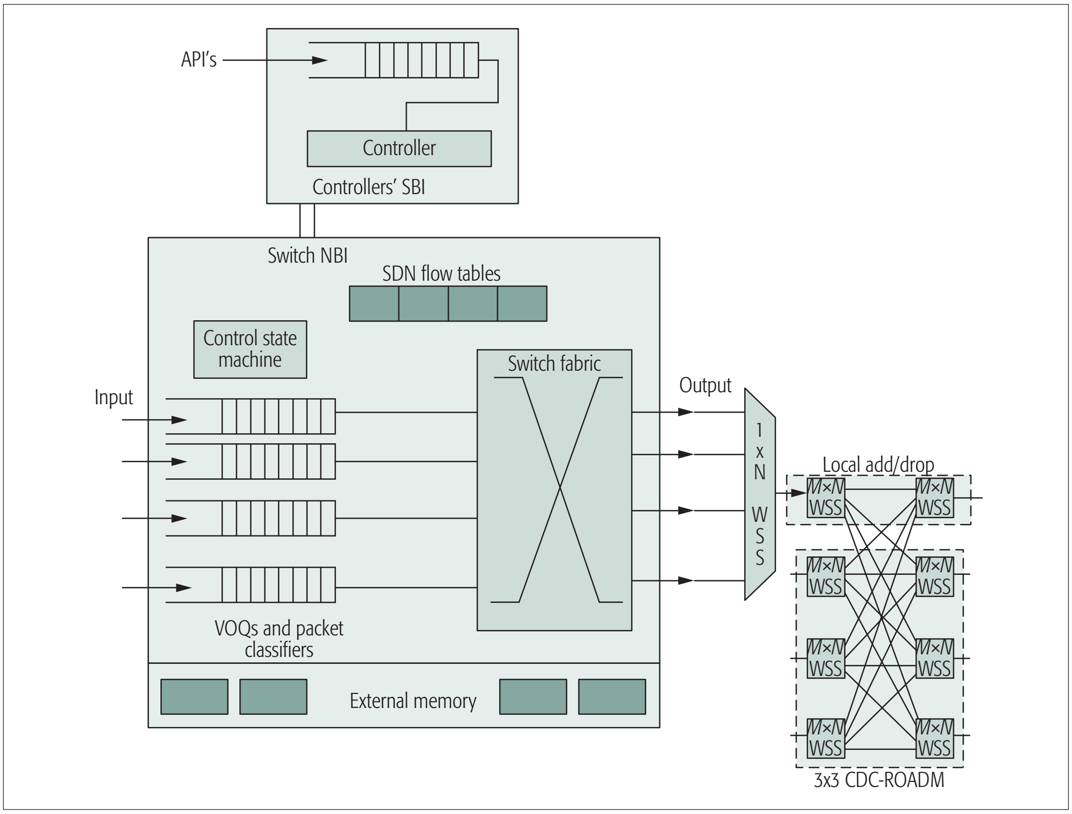

Figure 2. Switch Architecture to Implement SDN.computation (top) and VM migration analogy (bottom).

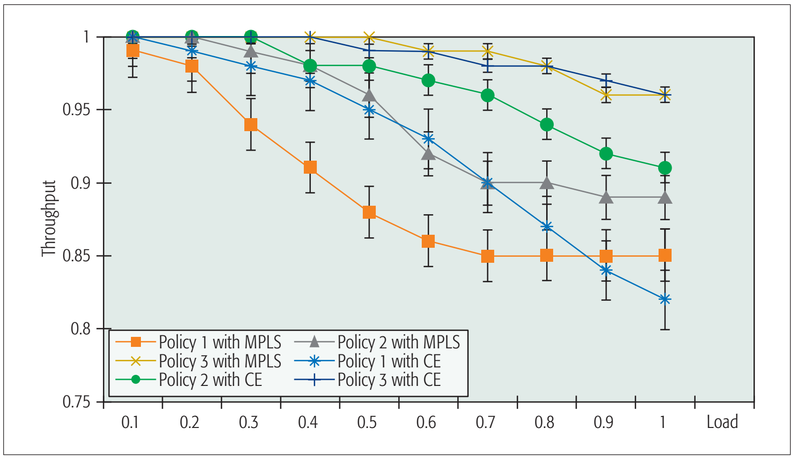

Figure 3. Throughput as a function of load for different policies.

through an NE. The controller creates VNEPs such that they balance each other in terms of parameterized requirements. For example, services with similar delay and bandwidth requirements are load balanced. The controller also provides for equal cost multiple paths (ECMPs) to load balance the service.

In our simulation model, we rationalize service requirements based on their utility to the network (revenue for the provider) and normalize the utility over delay constraints. We then provision services such that the delay constraints are met, while bundling as many services together as possible. The LSSM policy is a greedy heuristic, and its complexity is fourth-order polynomial in terms of number of links in the network; hence, its functioning depends on graph size.

SIMULATION MODEL AND HYPOTHESIS VERIFICATION

A simulation model was built to test our VNEP hypothesis as a method to facilitate interaction between SPs and ASPs. We model a provider network with two autonomous systems (ASs) and five metropolitan regions, with each region divided randomly into 20, 40, 60, 80, and 100 access regions. The backbone and metro networks use fiber, while the access networks could be wireless/fiber/coaxial cable-based. Our goal is to evaluate the impact of NV over different technologies by provisioning OTT services. To this end, the simulation model implements each technology solution using proposed VNEP creation policies.

Each access region has between 10,000 and 100,000 subscribers, and is connected to a metro network with multiple metros backhauled to a core network (wholly viewed as a single AS). The point of presence (POP) connecting the access to the metro supports ROADMs. The overlay depends on the technology being simulated; we study IP/MPLS, MPLS, OTN, and CE technologies, and the seven cases of Table 2 are deployed randomly. The control plane is implemented as an SDN overlay that consists of controllers, one for an AS of 10,000 users and hierarchically arranged thereafter.

The simulation model works as follows: Randomly generated service requests have specific QoS parameters. Services are organized into two levels — services and sessions. Services are exponentially distributed with a mean holding time of 6 months, while session holding time is exponentially distributed with a mean time equivalent to a

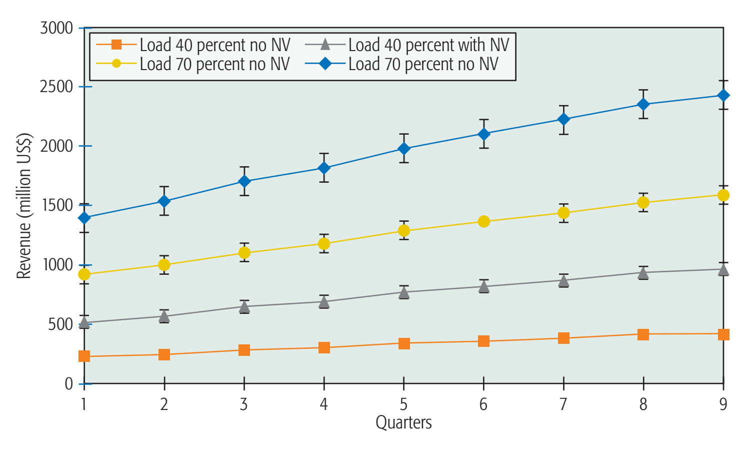

Figure 4. SP revenue with and without ASP revenue-sharing through NV.

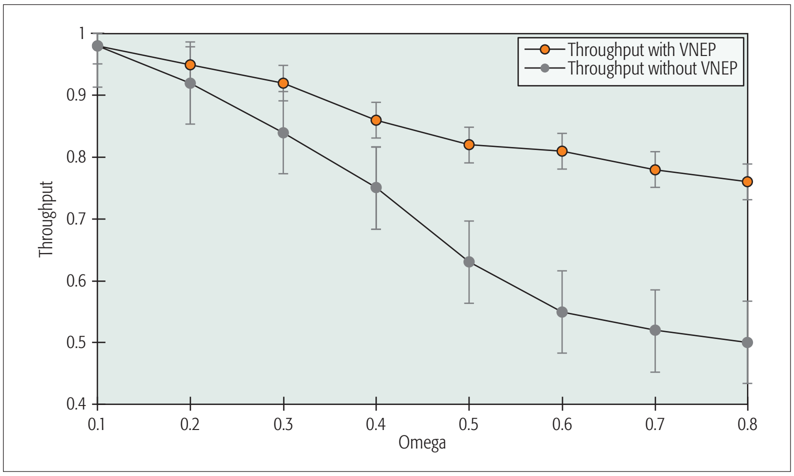

Figure 5. Throughput versus ratio of MP2MP/P2P traffic.

100 MB video file download session. The service is guided to the appropriate controller, which uses one of the three VNEP creation policies and evaluates whether provisioning is possible. Services are lumped through pre-assigned aggregation policies. Once a service is provisioned, we compute service and switch statistics.

Load is computed as average occupancy of all the services to the maximum allowable input rate across all the ingress ports. MPLS LSRs have 1 Gb/s and 10 Gb/s interfaces and a net switching capacity of 640 Gb/s [9]; CE switches have 1 Gb/s and 10 Gb/s interfaces and 80 Gb/s fabric that is stacked to create a 640Gb/s node [7]. ODU switching is assumed at ODU0/1/2e [10]. Transport wavelengths can be generated by an MPLS/CE/IP forwarding plane and support 10 Gb/s, 40 Gb/s, and 100 Gb/s. Cost is computed as in [11] for both capital expenditure (CAPEX) and operational expenditure (OPEX), while we assume that for provisioning OTT services, the OTT ASP shares 20 percent of its revenue with the SP.

Figure 3 compares all three policies used for VNEP computation using MPLS and CE technologies using the FPGA, FPGAs+ASIC, and NP+ASIC approaches. We show throughput vs. load with error bars indicating stability of results. MPLS and CE were chosen as likely candidates for cost considerations. A peculiar behavior is that policy 3 has the best throughput, while being able to take service latency into consideration.

Figure 4 highlights the effect of ASP revenue sharing through NV on the SP revenue. It shows that there is sizable incentive for ASPs to share their revenue as the providers would be able to grow the network, thereby facilitating larger and qualitatively superior reach for the ASPs. Figure 4 is generated as follows: We first measure ASP revenue without NV and no revenue sharing. NV is implemented using the most popular approach, FPGA+ASIC, and uses the LSSM policy. We compute the revenue by pegging each service at 30–40 percent higher price than before. For example, a 12 Mb/s HD video pipe was priced at US$20 per month with no revenue sharing and hence no NV support. The same pipe with guaranteed bandwidth (no packet loss) is priced at US$26, while it is priced at US$30 with bounded latency and 50 ms restoration of service in case of fiber cut/equipment failure.

Figure 5 studies the impact of VNEP on throughput in the network as a function of the ratio (defined as Omega) of multipoint-to-multipoint (MP2MP) traffic to point-to-point traffic. The graph is generated for the case of MPLS. As Omega increases, the throughput without VNEP decreases rapidly, while that with VNEP decreases gradually. This is a critical result showing the maximum benefit of the use of VNEP, which is modeled as implemented in FPGA+ASICs. The graph shows how VNEPs can impact new service support such as multicast services that are poorly handled at higher loads.

CONCLUSION

We present an approach to integrate OTT application providers with service providers using network virtualization in hardware. Our work is inspired by [9, 12]. We propose the concept of virtual network equipment partitions that enable an NE to be partitioned as per service requirement, thereby benefiting from programmability of the control plane. Policies to partition an NE are discussed. Results from a simulation study show the benefit for ASPs in a provider network using NV-compliant hardware.

REFERENCES

[1] V. Mishra, "Routing Money, Not Packets" *Commun. ACM*, vol. 58 no. 6, pp. 24–27.

[2] FCC Report: Protecting and Promoting the Open Internet, FCC 15-24, GN Docket No 14-28.

[3] K. C. Webb, A. C. Snoeren, and K. Yocum, "Topology Switching for Data Center Networks," *Proc. Hot-ICE*, 2011.

[4] J. Mogul and L. Popa, "What We Talk about When We Talk about Cloud Network Performance," *ACM Proc. Sigcomm 2013*, Chicago, IL.

[5] A. Gumaste and S. Akhtar, "Evolution of Packet-Optical Integration in Backbone and Metropolitan High-Speed Networks: A Standards Perspective" *IEEE Commun. Mag.*, vol. 51, no. 11, Nov. 2013, pp. 105–11.

[6] Infonetics Research, "Data Center and Enterprise SDN Hardware and Software Report," 2013.

[7] S. Bidkar *et al.*, "On the Design, Implementation, Analysis, and Prototyping of a 1-ms, Energy-Efficient, Carrier-Class Optical-Ethernet Switch Router" *IEEE/OSA J. Lightwave Tech.*, vol. 32, no 17, pp 3043–60.

[8] S. Das, G. Parulkar, and N. McKeown, "Rethinking IP Core Networks," *IEEE J. Optical Commun. Networking*, Dec. 2013.

[9] P. Bossharty, "Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN," *ACM Proc. Sigcomm 2013*, Hong Kong, China.

[10] Per Harald Knudsen-Baas, *OTN Switching*, Master's thesis, Dept. of Telematics, Norwegian Univ. of Science and Tech-

nology, June 2011.
[11] A. Mathew *et al.*, "Multi-Layer High-Speed Network Design in Mobile Backhaul Using Robust Optimization" *IEEE/OSA J. Opt. Commun. and Networking*, vol. 7. no. 4, Apr. 2015. pp 352–67.
[12] E. Haleplidis *et al.*, "Network Programmability with ForCES," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 3, 2015.

BIOGRAPHIES

ASHWIN GUMASTE is currently an associate professor in the Department of Computer Science and Engineering at the Indian Institute of Technology (ITT) Bombay, Mumbai. He was the Institute Chair Associate Professor (2012–2015) and the JR Isaac Chair (2008–2011), and from 2008 to 2010 he was a visiting scientist at the Massachusetts Institute of Technology, Cambridge. He has held positions with Fujitsu Laboratories (USA) Inc and Cisco Systems, and has been a consultant to Nokia Siemens Networks. He has also held short-term positions at Comcast, Lawrence Berkeley National Labs, and Iowa State University. His work on light-trails has been widely referred to, deployed, and recognized by both industry and academia. His recent work on omnipresent Ethernet has been adopted by tier 1 service providers and also resulted in the largest ever acquisition between any IIT and the industry. This has led to a family of transport products under the premise of carrier Ethernet switch routers. He has 23 granted U.S. patents and has published about 150 papers in referred conferences and journals. He has also authored three books on broadband networks. For his contributions he was awarded the Government of India's DST Swaranajayanti Fellowship in 2013, the DAE-SRC Outstanding Research Investigator Award in 2010, the Vikram Sarabhai research award in 2012, the IBM Faculty award in 2012, the NASI-Reliance Industries Platinum Jubilee award in 2016, as well as the Indian National Academy of Engineeringís (INAE) Young Engineer Award in 2010.

TAMAL DAS is a research scientist at IIT Bombay. Prior to this, he was a postdoctoral researcher at the Technical University of Braunschweig, Germany. He received his Ph.D. from IIT Bombay, and B.Tech.+M.Tech. from IIT Delhi. His research interests are in stochastic analysis, telecommunication networks, and network algorithms. He has authored over 25 high-quality scientific publications, and was a recipient of the IEEE ANTS 2010 Best Paper Award.

KANDARP KHANDWALA is currently a Master's student at the University of California San Diego, specializing in human-computer interaction. He is the co-inventor of two U.S. patents applied for in the areas of online shopping cart abandonment and (predicting) permission email unsubscription. He received his Bachelor's degree in computer science and engineering from IIT Bombay in 2015, where he conducted research under Prof. Ashwin Gumaste. He was also a finalist for the Aditya Birla Scholarships Programme, 2011.

INDER MONGA serves as the division director of the Scientific Networking Division, Lawrence Berkeley National Lab, and executive director of the Energy Sciences Network, a high-performance network interconnecting the National Laboratory System in the United States. In addition to managing the organization, his research interests include developing and deploying advanced networking services for collaborative and distributed big data science. Recently, his focus is on the broad adoption of SDN in the wide area network including recent work on operationalizing SDN, multi-layer SDN, and software-defined exchanges. He actively contributes to the Open Networking Foundation (ONF), the standards organization for SDN, as Chair of the ONF Research Associates and a member of the Open Source Software Leadership Council. He currently holds 20 patents and has over 20 years of industry and research experience in telecommunications and data networking.