Network Function Virtualization: Challenges and Opportunities for Innovations

Bo Han, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee

ABSTRACT

Network function virtualization was recently proposed to improve the flexibility of network service provisioning and reduce the time to market of new services. By leveraging virtualization technologies and commercial off-the-shelf programmable hardware, such as general-purpose servers, storage, and switches, NFV decouples the software implementation of network functions from the underlying hardware. As an emerging technology, NFV brings several challenges to network operators, such as the guarantee of network performance for virtual appliances, their dynamic instantiation and migration, and their efficient placement. In this article, we provide a brief overview of NFV, explain its requirements and architectural framework, present several use cases, and discuss the challenges and future directions in this burgeoning research area.

INTRODUCTION

It is well known that bringing new services into today's networks is becoming increasingly difficult due to the proprietary nature of existing hardware appliances, the cost of offering the space and energy for a variety of middle-boxes, and the lack of skilled professionals to integrate and maintain these services. Network function virtualization (NFV) was recently proposed to alleviate these problems, along with other emerging technologies, such as software defined networking (SDN) and cloud computing.[1]

NFV transforms how network operators architect their infrastructure by leveraging the full-blown virtualization technology to separate software instance from hardware platform, and by decoupling functionality from location for faster networking service provisioning [3]. Essentially, NFV implements network functions through software virtualization techniques and runs them on commodity hardware (i.e., industry standard servers, storage, and switches), as shown in Fig. 1. These virtual appliances can be instantiated on demand without the installation of new equipment. For example, network operators may run an open source software-based firewall in a virtual machine (VM) on an x86 platform. Recent trials have demonstrated that it is feasible to implement network functions on general-purpose processor-based platforms, for example, for physical layer signal processing [2] and components in cellular core networks [9].

As an innovative step toward implementing a lower-cost agile network infrastructure, NFV can potentially bring several benefits to network carriers, dramatically changing the landscape of the telecommunications industry. It may reduce capital investment and energy consumption by consolidating networking appliances, decrease the time to market of a new service by changing the typical innovation cycle of network operators (e.g., through software-based service deployment), and rapidly introduce targeted and tailored services based on customer needs, just to list a few.

Along with the benefits of NFV, network operators also face several technical challenges when deploying virtual appliances. A frequently raised issue about virtualized network functions (VNFs)[2] is their network performance. Previous work has shown that virtualization may lead to abnormal latency variations and significant throughput instability even when the underlying network is only lightly utilized [14]. Therefore, ensuring that network performance remains at least as good as that of purpose-built hardware implementations will be one of the key challenges in realizing NFV. Besides the network performance issue, another major problem network carriers are confronted with is how to smoothly migrate from the existing network infrastructure to NFV-based solutions, given the former's large scale and tight coupling among its components. Moreover, the separation of functionality from location also creates the problem of how to efficiently place the virtual appliances and dynamically instantiate them on demand.

These facts all impose the need to investigate open research issues brought by NFV in order to ensure its successful adoption. However, there are very limited prior efforts in the literature to offer an overview of aspects to be considered and issues to be addressed when adopting NFV. Our goal is to bridge this gap by identifying critical research challenges involved in the evolution toward NFV.

Bo Han, Vijay Gopalakrishnan, and Lusheng Ji are with AT&T Labs Research.

Seungjoon Lee is currently with Two Sigma Investments, LLC. This work was done when he was employed at AT&T Labs Research.

[1] We discuss the relationship between NFV, SDN, and cloud computing later.

[2] A VNF is the software instance in NFV that consists of some number or portion of VMs running different processes for a network function.

0163-6804/15/$25.00 © 2015 IEEE

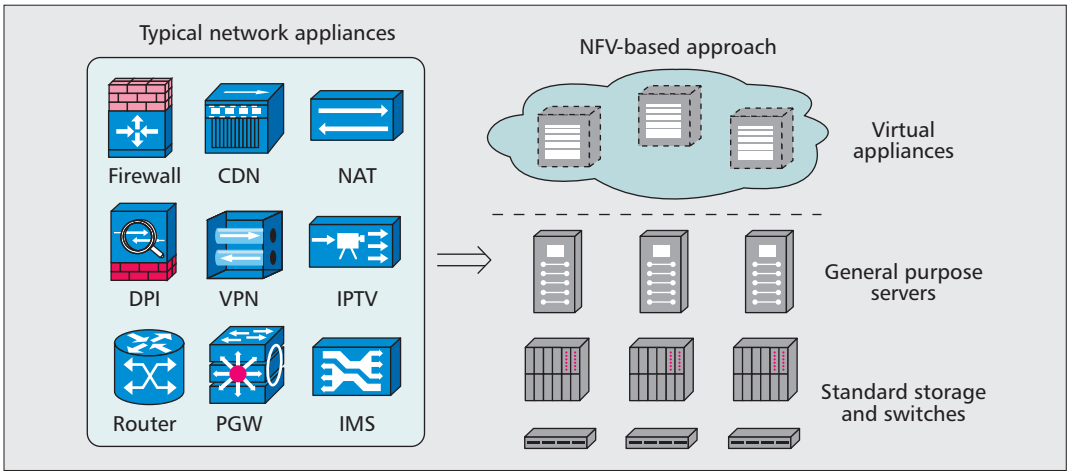

Figure 1. From dedicated hardware-based appliances for network services, such as firewalls, content delivery networks (CDNs), network address translation (NAT), deep packet inspection (DPI), virtual private networks (VPNs), IPTV, routers, packet data network gateways (PDN-GWs or PGWs), and IP multimedia subsystems (IMSs), to software-based NFV solutions.

When talking about software-based implementation of network functions through virtualization technologies on general-purpose servers, the first question we may ask is whether the performance, such as throughput and latency, will be affected.

In this article, we first present the related work and key technical requirements of NFV. We then introduce its architectural framework. We also describe several use cases of NFV, including the virtualization of the cellular core network and home network. Finally, we discuss the open research issues and point out future directions for NFV, focusing on the network performance of virtualized appliances, and their efficient instantiation, placement, and migration.

Related Work

The European Telecommunications Standards Institute (ETSI) has created an Industry Specification Group (ISG) for NFV to achieve the common architecture required to support VNFs through a consistent approach. This ISG was initiated by several leading telecommunication carriers, including AT&T, BT, China Mobile, Deutsche Telekom, Orange, Telefónica, and Verizon. It has quickly attracted broad industry support, and had over 150 members and participants by the end of 2013, ranging from network operators to equipment vendors and IT vendors.

The ETSI NFV ISG currently has four working groups: Infrastructure Architecture, Management and Orchestration, Software Architecture, and Reliability & Availability; and two expert groups: Security and Performance & Portability. Although it is not a standards development organization, it seeks to define the requirements that network operators may adopt and tailor for their commercial deployment. Part of this article (e.g., the architectural framework) is based on the NFV white paper [3] and several related specifications [12, 13] published by this ISG.

Besides ETSI, the Third Generation Partnership Project (3GPP) and Internet Engineering Task Force (IETF) have also been actively involved in NFV. The 3GPP Telecom Management working group (SA5) created a Study Item on the management of virtualized 3GPP network functions. The goal is to investigate whether the architectural framework proposed by ETSI NFV impacts the existing management reference model of 3GPP when all or some instances of 3GPP-defined network elements are virtualized. IETF has formed the Service Function Chaining (SFC) working group to study how to dynamically steer data traffic through a series of network functions, either physical or virtualized. In this article, we review some of the existing work and offer deeper insights on the research challenges of NFV. There are also several multivendor proofs of concept (PoCs) to build the confidence that NFV is a viable technology. For example, CloudNFV[3] is an open platform to implement NFV by leveraging cloud computing and SDN technologies in a multivendor environment. Services, functions, and resources in CloudNFV are represented in an "active virtualization" data model with two key components, the active contract and active resource. When it manages NFV-based services, CloudNFV integrates resource commitments in the active contract with resource state from the active resource.

Technical Requirements

In this section, we summarize the technical requirements when implementing VNFs, including their network performance, and manageability, reliability, and security.

Performance

When talking about software-based implementation of network functions through virtualization technologies on general-purpose servers, the first question we may ask is whether the performance, such as throughput and latency, will be affected. The per-instance capacity of a VNF may be less than the corresponding physical version on dedicated hardware.

Although it is hard to completely avoid performance degradation, we should keep it as small as possible while not impacting the portability of VNFs on heterogeneous hardware platforms. One possible solution is to leverage clustered VNF instances and modern software technologies, such as Linux New API (NAPI)[4] and Intel's Data Plane Development Kit

[3] http://cloudnfv.com/

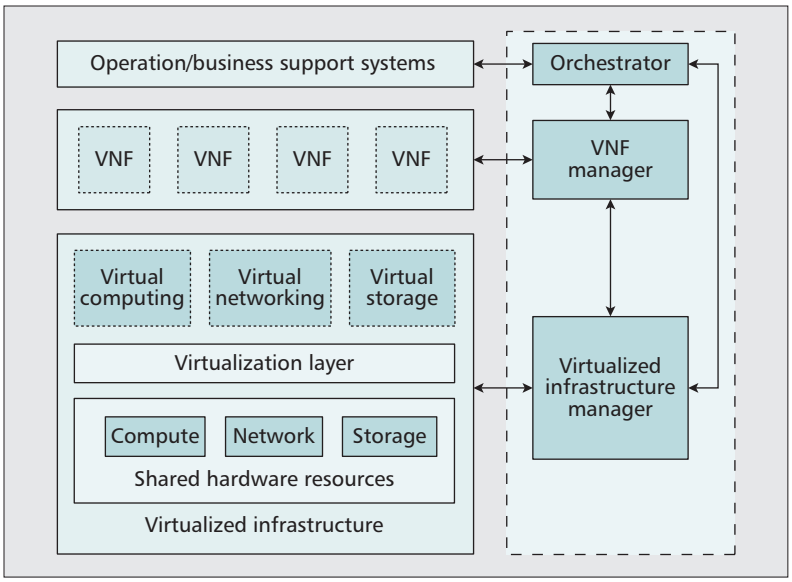

Figure 2. NFV architectural framework [12].

(DPDK).[5] When deploying VNF instances, we need to design efficient algorithms to split network load across a number of distributed and clustered VMs while keeping the latency requirement in mind. Moreover, the underlying NFV infrastructure should be able to gather network performance information at different levels (e.g., hypervisor, virtual switch, and network adapter). We discuss the research challenges related to NFV performance later.

The bottom line is that when designing NFV systems, we should understand the maximum achievable performance of the underlying programmable hardware platforms. Based on this information, we can make the proper design decisions.

Manageability

The NFV infrastructure should be able to instantiate VNFs in the right locations at the right time, dynamically allocate and scale hardware resources for them, and interconnect them to achieve service chaining.[6] This flexibility of service provisioning poses new requirements to manage both virtual and legacy appliances. The manageability in NFV is quite different from that in data center networking, where the hardware resources are almost equivalent, which makes their coordination easier. However, the cost and value of resources may vary significantly between network points of presence and customers' premises. The management functionality should take the variations into account and optimize resource usage across the wide area.

Since service unavailability is typically thought unacceptable, network carriers usually overprovision their services [5]; thus, the utilization of resources allocated to these services is normally low due to the offered redundancy for unexpected traffic increase or service element failure. If we share cloud resources across multiple services, and their failure modes are independent, we can leverage the pool of spare resources to provide the necessary redundancy across them and dynamically create VNFs to appropriately handle traffic increase or failure. In addition, NFV can potentially improve resource utilization through the elasticity feature of cloud computing, for example, by consolidating the workload on a small number of servers during overnight hours and turning the rest off (or using them for services such as online gaming). The management functionality should be able to support sharing spare resources and elastic provisioning of network services effectively.

Although NFV may make planned maintenance relatively easy [15], it presents new requirements for service quality management. Network operators should be able to obtain and process actionable information from various service impacting events, determine and correlate faults, and recover from them by monitoring compute, storage, and network resource usage during the life cycle of a VNF. Since VNFs can be dynamically created/migrated, it brings an additional dimension of complexity in terms of keeping track of where a given VNF is running. Moreover, a VNF can behave erratically even if the underlying infrastructure is running fine, which makes the detection of issues nontrivial.

Reliability and Stability

Reliability is an important requirement for network operators when offering specific services (e.g., voice call and video on demand), whether through physical or virtual network appliances. Carriers need to guarantee that service reliability and service level agreements are not affected when evolving to NFV. Purpose-built network equipment can provide the traditional five-nines reliability in the telecommunications industry. To meet the same reliability requirement, NFV needs to build resilience into software when moving to error-prone hardware platforms. Moreover, as mentioned above, the elasticity of service provisioning may require the consolidation and migration of VNFs based on traffic load and user demand. All these operations create new points of failure that should be handled automatically.

In addition, ensuring service stability poses another challenge to NFV, especially when reconfiguring or relocating a large number of software-based virtual appliances from different vendors and running on different hypervisors. Network operators should be able to move VNF components from one hardware platform onto a different platform while still satisfying the service continuity requirement. They also need to specify the values of several key performance indicators to achieve service stability and continuity, including maximum unintentional packet loss rate and call/session drop rate, maximum per-flow delay and latency variation, and maximum time to detect and recover from failures.

Security

When deploying VNFs, operators need to make sure that the security features of their network will not be affected. NFV may bring new security concerns along with its benefits. Virtual appliances may run in data centers that are not owned by network operators directly. These VNFs may even be outsourced to third parties [11]. The introduction of new elements, such as orchestra-

[4] http://www.linuxfoundation.org/collaborate/workgroups/networking/napi

[5] http://dpdk.org/

[6] Service chaining describes a method for the delivery of network services based on their function associations, and enables the ordering and topological independence of the network functions.

tors and hypervisors, may generate additional security vulnerabilities that increase the load of intrusion detection systems. The underlying shared networking and storage can also introduce new security threats, for example, when running a software router in a VM that shares the physical resources with other network appliances. Moreover, these software-based components may be offered by different vendors, potentially creating security holes due to integration complexity. All these changes require us to rethink security issues when designing and building NFV systems.

Design and Architectural Framework

Virtualization provides us the opportunity for a flexible software design. Existing networking services are supported by diverse network functions that are connected in a static way. NFV enables additional *dynamic* schemes to create and manage network functions. Its key concept is the VNF forwarding graph, which simplifies the service chain provisioning by quickly and inexpensively creating, modifying, and removing service chains. On one hand, we can compose several VNFs together to reduce management complexity, for instance, by merging the serving gateway (SGW) and PGW of a 4G core network into a single box. On the other hand, we can decompose a VNF into smaller functional blocks for reusability and faster response time. However, we note that the actual carrier-grade deployment of VNF instances should be transparent to end-to-end services.

Compared to current practice, NFV introduces the following three major differences [12]:

- *Separation of software from hardware*: This separation enables the software to evolve independent from the hardware, and vice versa.
- *Flexible deployment of network functions*: NFV can automatically deploy network-function software on a pool of hardware resources that may run different functions at different times in different data centers.
- *Dynamic service provisioning*: Network operators can scale the NFV performance dynamically and on a grow-as-you-need basis with fine granularity control based on the current network conditions.

We illustrate the high-level architectural framework of NFV in Fig. 2. Its four major functional blocks are the orchestrator, VNF manager, virtualization layer, and virtualized infrastructure manager. The *orchestrator* is responsible for the management and orchestration of software resources and the virtualized hardware infrastructure to realize networking services. The *VNF manager* is in charge of the instantiation, scaling, termination, and update events during the life cycle of a VNF, and supports zero-touch automation. The *virtualization layer* abstracts the physical resources and anchors the VNFs to the virtualized infrastructure. It ensures that the VNF life cycle is independent of the underlying hardware platforms by offering standardized interfaces. This type of functionality is typically provided in the form of virtual machines (VMs) and their hypervisors. The *virtualized infrastructure manager* is used to virtualize and manage the configurable compute, network, and storage resources, and control their interaction with VNFs. It allocates VMs onto hypervisors and manages their network connectivity. It also analyzes the root cause of performance issues and collects information about infrastructure fault for capacity planning and optimization.

As we can see from this architectural framework, the two major enablers of NFV are industry-standard servers and technologies developed for cloud computing. A common feature of industry-standard servers is that their high volume makes it easy to find interchangeable components inside them at a competitive price, compared to network appliances based on bespoke application-specific integrated circuits (ASICs). Using these general-purpose servers can also reduce the number of different hardware architectures in operators' networks and prolong the life cycle of hardware when technologies evolve (e.g., running different software versions on the same platform). Recent developments of cloud computing, such as various hypervisors, OpenStack, and Open vSwitch, also make NFV achievable in reality. For example, the cloud management and orchestration schemes enable the automatic instantiation and migration of VMs running specific network services.

NFV is closely related to other emerging technologies, such as SDN. SDN is a networking technology that decouples the control plane from the underlying data plane and consolidates the control functions into a logically centralized controller. NFV and SDN are mutually beneficial, highly complementary to each other, and share the same feature of promoting innovation, creativity, openness, and competitiveness. These two solutions can be combined to create greater value. For example, SDN can support NFV to enhance its performance, facilitate its operation, and simplify the compatibility with legacy deployments. However, we emphasize that the virtualization and deployment of network functions do not rely on SDN technologies, and vice versa.

NFV enables additional dynamic schemes to create and manage network functions. Its key concept is the VNF forwarding graph, which simplifies the service chain provisioning by quickly and inexpensively creating, modifying, and removing service chains.

Use Cases

In this section, we describe two use cases of NFV, the virtualization of a mobile core network and a home network. We focus on the problems of existing architecture and the benefits of NFV-based solutions. NFV is applicable to both data plane processing and control plane function. We refer interested readers to the specification of ETSI [13] for more use cases, such as the virtualization of the content delivery network (CDN) and fixed access network.

Virtualization of Mobile Core Network

Today's mobile core networks suffer from a huge variety of expensive and proprietary equipment, as well as inflexible hard-state signaling protocols [9]. When a specific function is not available, cellular operators have to replace existing equipment even if it is still sufficient for most purposes, which reveals the difficulty of

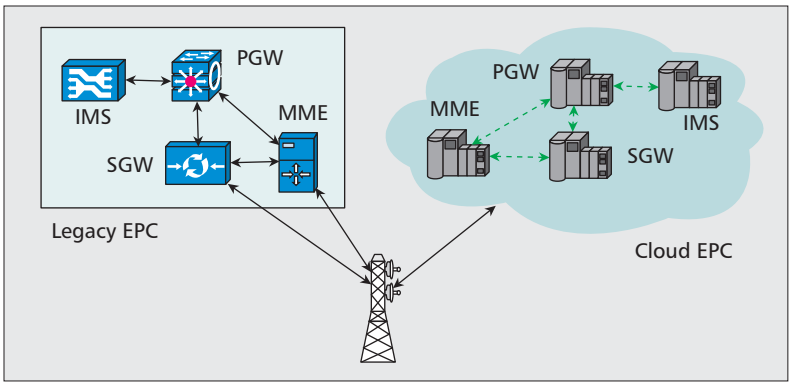

Figure 3. Virtualization of EPC and its coexistence with legacy EPC.

scaling offered services up and down rapidly as required. Moreover, the mobile core network leverages the tunneling mechanism over lower-layer transport protocols to and from a few centralized gateways (PGWs in case of 4G Evolved Packet Core, EPC) for the delivery of user data traffic. These long-distance permanent tunnels are very expensive to control and maintain for cellular operators.

Cloud EPC can potentially address these problems by virtualizing the mobile core network to meet changing market requirements. The virtualization targets of EPC include the mobility management entity (MME), home subscriber server (HSS), SGW, PGW, and policy and charging rules function (PCRF). We illustrate the virtualization of EPC for 4G LTE networks and its coexistence with the legacy EPC in Fig. 3. The coexistence is made possible through technologies such as MME pooling. We note that it is possible to virtualize only part of the mobile core network, such as SGW and PGW, and use physical appliances for other components.

Benefits: By virtualizing the aforementioned network functions, Cloud EPC allows us to move toward a more intelligent, resilient, and scalable core architecture. It enables flexible distribution of hardware resources to eliminate performance bottlenecks and rapid launch of innovative services to generate new revenue sources (e.g., machine-to-machine, M2M, communications). The virtualization of EPC frees distributed network resources from their geographic limitations to ensure service reliability and stability in the event of local resource failure, and reduce the total cost of ownership (TCO). It also makes the flexible deployment of SGW and PGW possible, for example, co-locating them with an eNodeB[7] and thus eliminating long-distance tunnels. With Cloud EPC, cellular carriers can not only expand their current horizontal market business, but also capitalize on previously untouched vertical markets.

Challenges: One of the challenging issues of Cloud EPC is that carriers need to dynamically redirect user traffic when scaling offered services. Early work has shown that SDN could enable the service chaining of various components in cellular core networks [9]. However, as SDN has primarily focused on data center networking in the past, it is still not clear how existing SDN controllers perform in the wide area in terms of scalability and manageability, especially for cellular networks, which have strict latency requirements. Another interesting topic for Cloud EPC is the support of M2M and Internet of Things (IoT) applications where there are a huge number of devices carrying very limited traffic but consuming the bearer resources in the core network.

Virtualization of Home Network

Network service providers offer home services through dedicated customer premises equipment (CPE) supported by network-located back-end systems. Typical CPE devices include residential gateways (RGs) for Internet access and set-top boxes (STBs) for multimedia services. Under this architecture, the delivery of time-shifted IPTV services is known to be complicated due to the interactive stream control functions (e.g., rewind and fast forward) [1]. The emerging NFV technology, with the availability of high-throughput last-mile access, facilitates the virtualization of the home network and brings down the complexity of IPTV services.

We depict the architecture of virtualized home networks in Fig. 4. The virtualization targets are STBs and a range of components of RGs, such as firewall, DHCP server, VPN gateway, and NAT router. By moving them to data centers, network and service operators need to provide only low-cost devices to customers for physical connectivity with low maintenance requirements, demonstrated by the three gray boxes at the bottom left corner of Fig. 4. These devices need to provide only layer 2 functionality for Internet access, as the layer 3 and above functions of RGs are moved into the operators' network. We note that with this virtual architecture, it is possible to share some functionalities of RGs and STBs among customers.

Benefits: This virtualized architecture presents numerous advantages to network operators and end users. First, it reduces the operating expense by avoiding the constant maintenance and updating of the CPE devices, and alleviating the call center and product return burdens. Second, it improves the quality of experience by offering near unlimited storage capacity and enabling access to all services and shared content from different locations and multiple devices, such as smartphones and tablets. Third, it allows dynamic service quality management and controlled sharing among user application streams which helps content providers programmatically provision capacity to end users via open APIs. Finally, it introduces new services more smoothly and less cumbersome by minimizing the dependency on the CPE functions.

Challenges: A fundamental issue in this area is the performance of virtualized packet processing on standard high volume servers. To achieve the same or comparable performance in a virtual environment as in bare metal, we need to carefully design the software architecture and configure the system parameters correctly, as indicated in the testing of virtualized Broadband Remote Access Server (BRAS) PoC from Intel.[8] Moreover, security and privacy issue will be another research challenge when sharing virtualized resource among customers to minimize operating cost.

[7] Although similar schemes have been proposed in the context of mobility management in cellular networks, NFV enables the global orchestration of these entities and their flexible migration, which may further improve mobility management.

[8] Available at http://networkbuilders.intel.com/

Discussion

In various areas where NFV is expected to deliver benefits, different network functions may generate different value and face different challenges and difficulties when moving towards virtualization. For example, based on a recent analysis from Ericsson,[9] there may be higher value and less of a challenge to virtualize home network and media distribution network, but lower value and more of a challenge to virtualize access network and core routers. However, we note that the trade-off between value and challenge may change as the underlying technologies evolve.

Research Challenges and Future Directions

In this section, we discuss some of the research challenges and future directions for NFV, including the network performance of virtualization, the placement, instantiation and migration of virtual appliances, and the outsourcing of VNFs.

Network Performance of VNF

The recent effort from the telecommunications industry has been centered on the software virtualization framework (e.g,. management and orchestration). However, it is challenging to offer guaranteed network performance for virtual appliances. Wang and Ng [14] measured the end-to-end networking performance of the Amazon EC2 cloud service. They found that the sharing of processors may lead to very unstable TCP/UDP throughput, fluctuating between zero and 1 Gb/s at the tens of milliseconds time granularity, and the delay variations among Amazon EC2 instances can be 100 times larger than most propagation delays, which are smaller than 0.2 ms, even when the network is not heavily loaded. The unstable networking characteristics caused by virtualization can obviously affect the performance and deployment of virtual appliances.

As mentioned earlier, it may be possible to leverage Linux NAPI and Intel's DPDK to improve the network performance of VNFs. NAPI is a modification of the packet processing framework in Linux device drivers, aiming to improve the performance of high-speed networking. It achieves this goal by disabling some interrupts when the network traffic load is high and switching to polling the devices instead, and thus avoids frequent interruptions sharing the same message that there are lots of packets to process. Another advantage of this polling-based approach is that when the kernel is overwhelmed, the packets that cannot be handled in time are simply dropped in the device queues (i.e., overwritten in the incoming buffer). Intel's DPDK is another software-based acceleration for high-speed networking applications that also uses polling to avoid the overhead of interrupt processing. Recent work by Hwang *et al.* [6] extends the DPDK libraries to provide low latency and high throughput networking in virtualized environments.

Placement of Virtual Appliances

Ideally, network operators should place VNFs where they will be used most effectively and least expensively. Although the virtualization of certain network functions is straightforward, there are a number of network functions that have strict delay requirements. For example, network functions offered by middle-boxes usually depend on the network topology, and these boxes are placed on the direct path between two endpoints. When virtualizing these functions and moving their software implementations into data centers, data traffic may go through indirect paths, causing a potential delay of packets. Therefore, the placement of VMs that carry VNFs is crucial to the performance of offered services. For these services, it would be advantageous and efficient to run some network functions at the edge of the network [7].

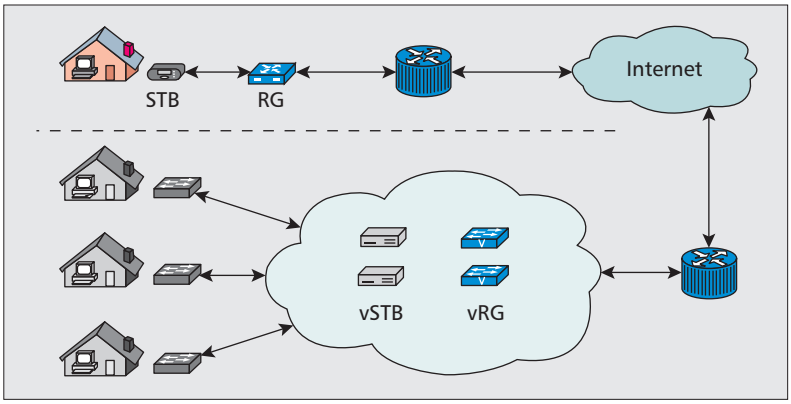

Figure 4. Virtualization of home network.

Using a mobile core network as an example, we could place a PGW, which currently sits in the cellular core network, right next to an eNodeB, and forward user traffic to the Internet as early as possible. However, the co-location of the PGW and eNodeB will make mobility management difficult, as neighboring eNodeBs will no longer share the same PGW as the anchor point. A possible solution would be to install virtualized PGWs that handle traffic for a small geographical area at the mobile telephone switching office (MTSO) or some other network points of presence in the metro area. Future work regarding low-latency operation should be based on investigation of the redirection architecture and the carrier's data center footprint. Placement problems usually involve optimization through linear programming, integer programming, or a mix, which works on a snapshot of the network and may take a long time to solve an instance. Thus, the online approximation algorithms for these optimization problems are challenging, given the dynamic nature of user traffic.

Instantiation and Migration of Virtual Appliances

Network infrastructure will become more fluid when deploying VNFs. To consolidate VNFs running in VMs based on traffic demand, network operators need to instantiate and migrate virtual appliances dynamically and efficiently. The native solution of running VNFs in Linux or other commodity OS VMs has a slow instantiation time (several seconds) and a relatively large memory footprint. The carrier-grade deployment of VNFs requires a lightweight VM implementa-

[9] http://www-ipv6.ericsson.com/ericsson/industryanalysts/telebriefings/

It is envisioned that NFV, along with cloud computing and SDN, will become a critical enabling technology to radically revolutionize the way network operators architect and monetize their infrastructure. NFV is prospectively the unifying revolution among the three, offering more revenue opportunities in the services value chain.

tion. For instance, Martins *et al.* [8] recently proposed ClickOS, a tiny Xen-based VM to facilitate NFV. ClickOS can be instantiated within around 30 ms and requires about 5 MB memory when running. However, optimizing the performance of this type of lightweight simplified VM, especially during wide-area migration, is still an open research issue.

Take virtual routers as an example, by enabling their free movement, carriers can separate the logical configurations (e.g., packet forwarding functions) from physical routers, and simplify management tasks such as planned maintenance [15]. However, it is challenging to keep the packet forwarding uninterrupted, and the migration disruptions and operating expenses minimized, while at the same time guaranteeing the stringent throughput and latency requirements and other service level agreements. To solve this problem, FreeFlow [10] has been proposed to offer efficient, transparent, and balanced elasticity for virtual middle-boxes, building on top of a state-centric system-level abstraction of network functions. OpenNF [4] is a control plane framework that provides coordinated control of network forwarding state and internal state of network functions through a set of APIs to export and import the middle-box state. A common requirement of these approaches is that they all need to modify the middlebox implementations to achieve efficient migration of virtual appliances. Hence, they cannot be applied to existing implementations as is.

VNF Outsourcing

The end-to-end principle of initial Internet architecture that does not modify packets on the fly is no longer valid in current networks with the deployment of a variety of middle-boxes. Based on a study of 57 enterprise networks with different sizes, ranging from fewer than 1000 hosts to more than 100,000 hosts, Sherry *et al.* [11] found that the number of middle-boxes in a typical enterprise is comparable to its number of hosted routers. In the last five years, surveyed large networks have paid more than US$ 1 million for their middle-box equipment. Moreover, a network with about 100 middle-boxes may need a management team of 100–500 personnel for tasks such as configuration, upgrades, monitoring, diagnostics, training, and vendor interaction [11].

By advocating the split of network functions and their locations, NFV makes the outsourcing of middle-boxes to a third party [11] easier, which may release network carriers from some of the cumbersome operation and maintenance tasks. With the help of *VNF service providers* (e.g., cloud service providers or their partners), end users and small businesses may also be able to enjoy more diverse networking services previously not affordable due to their associated complexity and costs. However, the charging rules and policy interactions between carrier network infrastructure and outsourced VNFs need to be carefully investigated before taking actual actions. Another open question along this direction is to identify what types of VNFs can be outsourced to third parties and how to do it efficiently.

There are also several other open research issues for NFV. For example, using dedicated hardware appliances, it is relatively easy to identify which component is malfunctioning and isolate it when a failure occurs. When deploying network functions in software at different locations, *troubleshooting* and *fault isolation* become harder. Moreover, as the creation of VMs is easy, when the number of VNFs increases, so-called VM sprawl could happen. There may be a large amount of VNFs sprawled across the network even if they are seldom used. As a result, the same management inefficiency problem that NFV was proposed to solve may recur. The efficient *management* and *orchestration* of VNFs, especially in the wide area, is another challenging issue.

Conclusion

In this article, we present an overview of the emerging network function virtualization technology, illustrate its architectural framework, summarize several use cases, and discuss some interesting future research directions. NFV extracts the functionality in specialized appliances and replicates it in virtual form. It is envisioned that NFV, along with cloud computing and SDN, will become a critical enabling technology to radically revolutionize the way network operators architect and monetize their infrastructure. NFV is prospectively the unifying revolution among the three, offering more revenue opportunities in the services value chain. We are looking forward to more initiatives from the networking research community to tackle various challenging issues introduced by NFV and its widespread and successful adoption.

Acknowledgment

We thank the Guest Editors and anonymous reviewers for their insightful comments and suggestions, which improved the overall quality of this article. We thank Jennifer Yates, Changbin Liu, and Feng Qian for discussions and suggestions.

References

[1] V. Aggarwal *et al.*, "Optimizing Cloud Resources for Delivering IPTV Services Through Virtualization," *IEEE Trans. Multimedia*, vol. 15, no. 4, June 2013, pp. 789–801.

[2] China Mobile Research Institute, "C-RAN The Road towards Green RAN," China Mobile White Paper, Oct. 2011.

[3] M. Chiosi *et al.*, "Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action," ETSI White Paper, Oct. 2012.

[4] A. Gember-Jacobson *et al.*, "OpenNF: Enabling Innovation in Network Function Control," *Proc. SIGCOMM 2014*, Aug. 2014, pp. 163–74.

[5] A. Greenberg *et al.*, "The Cost of a Cloud: Research Problems in Data Center Networks," *ACM SIGCOMM Computer Commun. Review*, vol. 39, no. 1, Jan. 2009, pp. 68–73.

[6] J. Hwang, K. K. Ramakrishnan, and T. Wood, "NetVM: High Performance and Flexible Networking Using Virtualization on Commodity Platforms," *Proc. NSDI '14*, Apr. 2014, pp. 445–58.

[7] A. Manzalini *et al.*, "Clouds of Virtual Machines in Edge Networks," *IEEE Commun. Mag.*, vol. 51, no. 7, July 2013, pp. 63–70.

[8] J. Martins *et al.*, "ClickOS and the Art of Network Function Virtualization," *Proc. NSDI 2014*, Apr. 2014, pp. 459–73.

[9] K. Pentikousis, Y. Wang, and W. Hu, "MobileFlow: Toward Software-Defined Mobile Networks," *IEEE Commun. Mag.*, vol. 51, no. 7, July 2013, pp. 44–53.

[10] S. Rajagopalan *et al.*, "Split/Merge: System Support for Elastic Execution in Virtual Middleboxes," *Proc. NSDI '13*, Apr. 2013, pp. 227–40.
[11] J. Sherry *et al.*, "Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service," *Proc. SIGCOMM '12*, Aug. 2012, pp. 13–24.
[12] ETSI, "Network Functions Virtualisation (NFV); Architectural Framework," GS NFV 002 (v. 1.1.1), Oct. 2013.
[13] ETSI, "Network Functions Virtualisation (NFV); Use Cases," GS NFV 001 (v. 1.1.1), Oct. 2013.
[14] G. Wang and T. S. E. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," *Proc. INFOCOM '10*, Mar. 2010, pp. 1163–71.
[15] Y. Wang *et al.*, "Virtual Routers on the Move: Live Router Migration as a Network- Management Primitive," *Proc. SIGCOMM '08*, Aug. 2008, pp. 231–42.

Biographies

Bo Han received his Bachelor's degree in computer science and technology from Tsinghua University in 2000, his M.Phil. degree in computer science from City University of Hong Kong in 2006, and his Ph.D. degree in computer science from the University of Maryland in 2012. He is currently a senior inventive scientist at AT&T Labs Research. His research interests are in the areas of wireless networking, mobile computing, software defined networking, and network functions virtualization, with a focus on developing simple yet efficient and elegant solutions for real-world networking and systems problems.

Vijay Gopalakrishnan [M] is a director in the Network Evolution Department in AT&T Labs–Research, leading a small team focused on systems challenges in the architecture, protocols, and management of networks. His research interests lie broadly in the area of networked systems where he has worked on topics of network management, content delivery, and the mobile web. Prior to joining AT&T, he got his Master's and Ph.D. degrees in computer science from the University of Maryland, College Park in 2003 and 2006, respectively. He is a member of ACM.

Lusheng Ji received his Ph.D. degree in computer science from the University of Maryland, College Park, in 2001. He is a Principal Member of Technical Staff, Research with the AT&T Shannon Laboratory, Bedminster, New Jersey. His research interests include wireless networking, mobile computing, wireless sensor networks, and networking security.

Seungjoon Lee is currently with Two Sigma Investments, LLC. Before joining Two Sigma in 2014, he was with AT&T Research, New Jersey, for more than eight years. He received his Ph.D in computer science from the University of Maryland, College Park in 2006. He also received Bachelor's and Master's degrees in computer science from Seoul National University, Korea, in 1996 and 2000. His research interests include large-scale systems, network management, content distribution, cloud computing, and mobile computing.