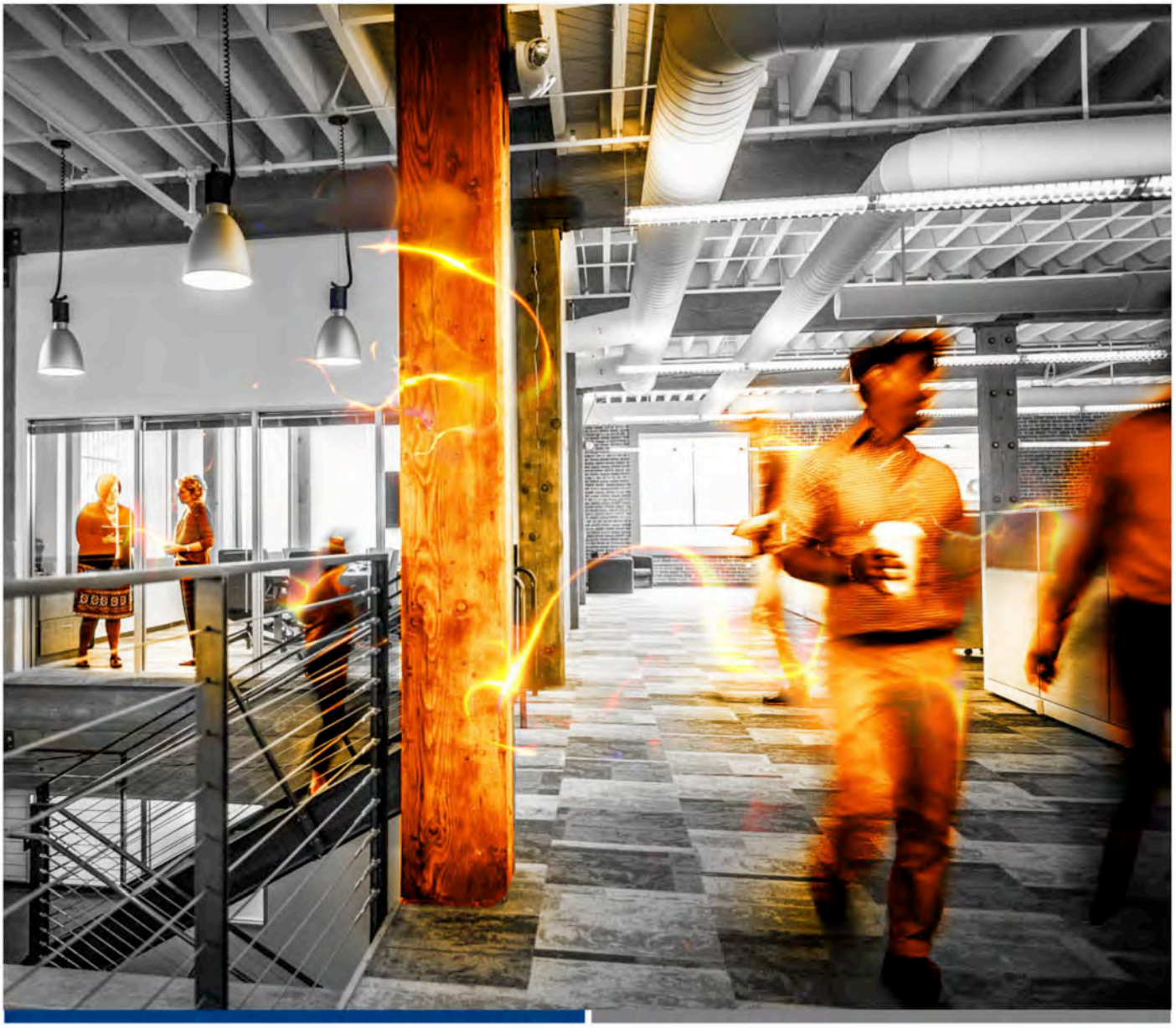

A Guide to NFV and SDN

by Martin Taylor, CTO, Metaswitch Networks

Metaswitch Networks

THE BRAINS OF THE NEW GLOBAL NETWORK

Executive Summary

The rapid evolution of standardized commodity computing infrastructure over the last few years has created an opportunity to build communications networks in totally new ways. A past that has been dominated by special-purpose network elements based on proprietary hardware will give way to a future in which network functions are implemented almost entirely in software running on shared pools of standard hardware resources, just like cloud-based IT workloads. The virtualization of network functions will enable new services to be created, deployed and managed far more quickly and at far lower cost than has been possible in the past. The programmability of the virtualized networking environment will go hand-in-hand with a new degree of programmability of the network functions themselves, enabled in part by a clean separation between control plane and data plane functions.

This new approach to building networks will provide numerous benefits to network operators, including reductions in both capital and operating costs and the promotion of rapid innovation in the services space. It should, therefore, enable network operators to compete far more successfully with over-the-top service providers, which currently represent a serious threat to the health of the networking industry.

This document is intended to help network operators understand the key technologies that support this transformation in networking, namely Network Functions Virtualization (NFV) and Software-Defined Networking (SDN); to assess how these technologies could be applied in their networks; and to identify the kinds of benefits that could flow from embracing them.

About the Author

Martin Taylor is chief technical officer of Metaswitch Networks. He joined the company in 2004, and headed up product management prior to becoming CTO. Previous roles have included founding CTO at CopperCom, a pioneer in Voice over DSL, where he led the ATM Forum standards initiative in Loop Emulation; VP of Network Architecture at Madge Networks, where he led the company's successful strategy in Token Ring switching; and business general manager at GEC-Marconi, where he introduced key innovations in Passive Optical Networking. In January 2014, Martin was recognized by *Light Reading* as one of the top five industry "movers and shakers" in Network Functions Virtualization.

Table of contents

Introduction

Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) are the two hottest topics for network operators right now. These technologies promise new ways of building communications networks at lower cost and with greater scope for innovation in network services. Regardless of what kind of network you operate – fixed, mobile, metro Ethernet, long-distance interconnect – or what kinds of services you offer – data, voice, messaging, content delivery, virtual private networks – NFV and SDN offer opportunities to transform the economics of your network while at the same time accelerating your ability to design and deploy new service capabilities. In short, NFV and SDN represent the most significant pivot in the telecom industry since the transition from TDM to packet got under way a decade or more ago.

This guide to NFV and SDN sets out to provide the key facts that will help you plan how these technologies will figure into the future of your network, and to describe a vision of the future that is enabled by NFV and SDN.

We'll start by defining the two terms NFV and SDN.

What is NFV?

Network Functions Virtualization is concerned with the transition of networks from collections of proprietary boxes to collections of software components running on industry-standard hardware. In a traditional network, each distinct function was typically implemented as a specialized appliance based on proprietary hardware. Such appliances invariably include a substantial amount of software, but the software and hardware can't be separated – they are highly dependent on one another. Examples of traditional proprietary hardware-based network elements include routers of various kinds, deep packet inspection devices, content delivery network appliances, firewalls, load balancers, network address translators, session border controllers, mobile base station controllers, mobile packet gateways and so on.

NFV is based on the proposition that network functions of the kind just described can be implemented entirely in software running on "industry-standard hardware." In general, industry-standard hardware is taken to mean commercial off-the-shelf servers based on Intel's x86 architecture, together with commercial off-the-shelf Ethernet switching devices.

This implies that the network function itself -- for example, a session border controller function -- is delivered to the network operator as piece of pure software. This is then installed by the network operator on a standardized hardware infrastructure that is typical of a data center environment: rack-mounted or blade servers connected by Ethernet switching systems.

The term "Network Functions Virtualization" was coined by a group of Tier 1 network operators who published a white paper in October 2012 under this title. This white paper is essential reading for anyone who wishes to understand the topic in more depth.

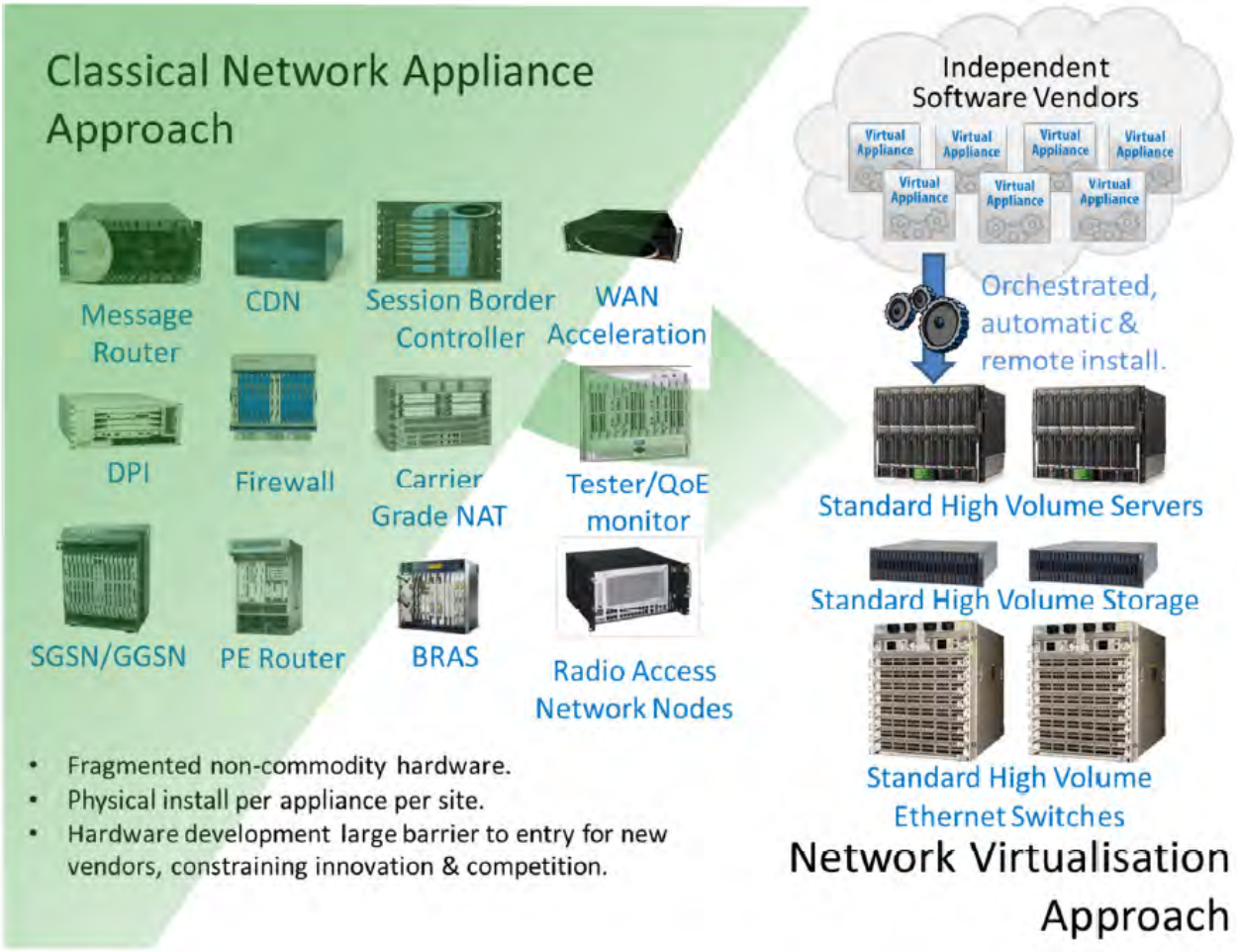

Figure 1: NFV Vision (acknowledgement: NFV White Paper published by ETSI)

What has enabled NFV?

If it's possible to build networks in this way, then you might well ask why networks have historically been based on proprietary boxes. The answer is that industry standard servers and their operating software environments have only recently become powerful and richly featured enough to compete effectively with purpose-built hardware in terms of cost, physical space and power consumption. "Recently," in this context, means over the last four to five years, during which time we have seen dramatic improvements in the bandwidth and packet processing throughput of x86 CPUs, and rapid growth in the number of CPU cores that are available per unit of rack space. We've also seen the emergence of commodity low-cost switching silicon that can match the silicon developed in-house by the leading router and switch vendors in terms of performance and features.

Most network functions include two distinct areas of activity: "control plane" and "data plane." The control plane activity is concerned with operations based on routing or signaling protocols that have the effect of maintaining routing tables, establishing and tearing down voice or video sessions, authorizing network access, etc. The data plane is concerned with the movement and processing of data or media traffic. For example, the data plane of a network address translator transfers data packets in both directions while swapping IP address and port numbers in each packet between the two addressing realms it connects. Likewise, the data plane of a session border controller relays RTP

packet streams and performs operations such as encryption and transcoding on those RTP packet streams.

In general, control plane functions are CPU-intensive but not bandwidth-intensive, while data plane functions are the opposite: bandwidth-intensive but (usually) not CPU-intensive. For example, setting up a voice call in an IMS network typically requires the exchange of 10 to 15 SIP signaling packets in the control plane, while three minutes of conversation requires the exchange of 36,000 RTP packets. The SIP signaling packets include numerous protocol elements that must be parsed, analyzed and acted upon, whereas the RTP packets may simply need to be relayed from one port to the other with no more than simple swapping of IP addresses.

Perhaps not surprisingly, control plane functions lend themselves well to general-purpose CPUs, and many proprietary network appliances include x86 or similar CPUs running software that implements the control plane. It is the data plane functions that have historically driven the need for proprietary hardware. Industry-standard server hardware wasn't originally designed to move millions of packets per second of network traffic between ports while performing routing-table or flow-table lookups, performing manipulations on packet headers and managing multiple outbound queues per port. It is in this area of that we have seen the most rapid changes in industry-standard hardware in the last few years:

- CPUs are available with more and more cores. Multiple cores in a single processor can operate on packet traffic in parallel, and that has increased the total processing power in a single host that is available to handle network packets by 30 to 40 percent each year.
- Larger and larger amounts of very fast cache memory is being built in to the latest generations of CPUs, offering the possibility of packet and flow table lookups at speeds comparable to the best dedicated hardware.
- The total bandwidth supported by the peripheral chips supporting the CPUs has increased dramatically in recent generations of the x86 architecture.

These are the evolutionary changes that have made NFV possible. But they don't explain why NFV is desirable. We'll deal with that next.

Why does it make sense to use industry-standard hardware?

Designing and building specialized hardware is difficult, expensive and time-consuming. Even the simplest and smallest-scale network appliance platform costs tens of millions of dollars to develop and takes at least two years from initial design to general availability. Network equipment vendors have to recover their investment in hardware development when they sell their products, so all of these costs have to be passed on to their customers. Specialized network equipment sells in far smaller volumes than industry-standard servers, so it needs to attract far higher margins than industry-standard servers in order to recover its development costs. Furthermore, the cost and elapsed time associated with a new hardware design are such that network equipment vendors generally can't afford to refresh their hardware platforms any more often than once every four to five years. During this time, industry-standard server hardware has typically gone through two or even three new generations of silicon, bringing price/performance and functionality improvements in each generation.

The very high costs of hardware development have also meant that only the very largest vendors -- those that can expect to win at least a 20 percent share of the global market for any given kind of network equipment -- can afford the investment. And specialized hardware development is getting *more* expensive at the same time as competition in the

network services space is driving down the capital expenditure budgets of network operators. Economy of scale means that, in a world dominated by proprietary hardware, only a very few very large network equipment vendors can survive and prosper, and the result for network operators has been a reduction in the range of choices available to them, and an excessive concentration of power in the hands of major vendors that can dictate the pace of network evolution and innovation.

By contrast, industry-standard server hardware has become a high-volume commodity benefiting from massive economies of scale and a highly competitive marketplace that keeps margins low. It's a far less expensive way to purchase the computing cycles needed to drive network services, and it continues to improve year-on-year.

What are the benefits of software-based network functions?

Software development is not nearly as capital-intensive as hardware development. Furthermore, software lends itself to a process of incremental improvement that is simply not possible with hardware. As a result, we can expect a much wider range of vendors to participate actively in the market for software-based network functions in comparison with traditional network boxes. Competition among these vendors will result in a healthy marketplace, offering network operators a far greater range of choices than before. And the pace of innovation in network functions is bound to accelerate, propelled both by increased competition and by the shorter release cycles that are typical of software-only products.

Network operators should also find it easier and less costly to approve new software-based network functions for deployment in their networks, by comparison with the approval process to which new hardware needs to be subjected.

With software-based network functions, it finally becomes feasible for network operators to drive service innovation by conducting small-scale trials of new service concepts. The capital cost exposure of conducting such trials is minimal, since they can be run on spare capacity provided by existing hardware resources, and the risk of disruption to existing services is minimized since the trials can be conducted inside what is effectively a "sandbox" that provides insulation from other established services. An experimental service that proves unsuccessful can quickly be shelved without stranding any hardware investment, while providing valuable lessons in what works and what doesn't. On the other hand, a new service concept that successfully gains market traction in early trials can be put into production and scaled up very quickly. The philosophy of prototyping new service concepts with the mantra of "fail fast, fail early and fail often" has been a key factor in the success of many over-the-top services, and software-based network functions enable network operators to embrace this philosophy for the first time.

All of these effects combined together should result in a dramatic improvement in the rate at which network operators can innovate in the service space. This is obviously hugely positive for the industry.

What's the role of virtualization?

So far, we have discussed the advantages of building networks based on software functions running on industry-standard hardware. NFV stands for "Network Functions Virtualization." So where does virtualization come into this picture?

The current generation of industry-standard servers offers up to 16 CPU cores per host, and that number will grow in each of the next few generations of x86 processors. The only way to make full use of that kind of processing power

across a broad range of applications is to use a hypervisor to support multiple virtual machines per host, with each virtual machine running a specific software-based network function. Virtualization is standard practice in the IT world – the Gartner Group recently estimated that two-thirds of the world's x86 workloads are running in virtualized form.

Supporting a network of any significant size with software-based network functions is going to require a substantial number of physical hosts. For reasons of flexibility and ease of management, ideally we want to be able to treat a collection of host machines as a pool of processing resources on top of which we can deploy our virtualized network functions. This can be accomplished by deploying cloud management software such as OpenStack. Cloud management software enables a network operator to create virtual machines that run specific network functions, to deploy those virtual machines on a pool of server resources, and to configure the connectivity relationships between the network functions in accordance with the needs of the services that they are supporting.

What's more, cloud management software makes it possible to automate many of the functions required to deploy a network service based on NFV. This literally enables "point and click" deployment of network functions. For example, to deploy a traditional physical Session Border Controller in the network, you have to order a box from your SBC vendor, ship it to the location where it is to be installed, rack it up, and connect it to power and to the physical network infrastructure. In an NFV environment, you can create a virtualized SBC in minutes entirely through simple actions performed at a cloud management console – to instantiate a software SBC image in the NFV cloud environment, and to configure virtual network connections so as to plumb the SBC into the relevant sub-nets. The operational cost savings arising from this kind of deployment automation represent perhaps the single biggest set of benefits promised by NFV.

What is SDN?

Software-Defined Networking is a little harder to define than Network Functions Virtualization, although NFV and SDN have much in common. The general characteristics of SDN can be described as follows:

- SDN is concerned with the implementation of network transport functions, from the physical layer up to the networking layer (L1 to L3).
- In SDN, there is a clear logical separation between the data plane and the control plane. The control plane and the data plane may be implemented in physically separate systems or they may be collocated, but they are always distinct sub-systems that communicate with each other either via standardized protocols or via well-defined APIs.
- The SDN control plane is implemented as pure software that is designed to run on industry-standard hardware. The SDN data plane may be implemented in specialized hardware or in industry-standard x86 server hardware, depending on the performance and capacity needs of the SDN networking element, and depending on whether specialized hardware transport interfaces are required. A single SDN control plane element may control multiple SDN data plane elements.
- The SDN control plane is typically expected to offer a degree of programmability that goes beyond the capabilities of standard routing protocols such as OSPF or BGP. These protocols typically route all flows toward a given destination along the same route, whereas SDN solutions have the flexibility to differentiate between multiple traffic flows addressed to the same destination and route such flows via different paths through the network.

What has enabled SDN?

The catalyst for the emergence of SDN arose from research in networking at Stanford University in 2007 and 2008. Researchers interested in experimenting with novel routing protocols proposed a method for remotely programming flow tables in Ethernet switching systems via a protocol that has come to be known as OpenFlow. This protocol provides a means for a control plane element to program flow tables so as to route any arbitrary network flow, identified by source address/port, destination address/port and protocol type, via any arbitrary switch port. The wider networking community then realized that OpenFlow could provide a standardized means to separate control plane software from data plane hardware, and that this could open up some interesting new opportunities.

It is perhaps worth pointing out that methods for separating and centralizing some aspects of control plane software were in existence prior to the emergence of OpenFlow. One such method is the Path Computation Element Protocol (PCEP), which was developed to enable separate software functions to compute paths through MPLS networks. SDN embraces such methods, and it is therefore incorrect to equate SDN with OpenFlow – although undoubtedly OpenFlow is likely to be front and center in the SDN thinking of network operators.

Another important enabling factor for SDN is the growing availability of low-cost switching silicon. So-called "merchant" switching silicon is now available at very low cost and with performance levels comparable to mid-range mainframe routers, and this has enabled independent commodity hardware vendors to build switching hardware, which, when combined with suitable control plane software, is capable of rivaling purpose-built routers at a fraction of the hardware cost.

What are the benefits of SDN?

There are three types of benefits that potentially arise from the application of SDN.

The first is concerned simply with reducing the cost of networking infrastructure. Low-cost commodity switching boxes combined with control plane software running either locally in the box or remotely on industry-standard servers are likely to cost a good deal less than traditional proprietary routers that do the same job.

Secondly, the SDN control plane can expose northbound APIs that make it easy for provisioning systems to configure the network to support various kinds of services such as Virtual Private Networks, with fine-grained control over Service Level Agreements and Traffic Engineering parameters. The SDN industry is very focused at present on the kinds of northbound APIs that SDN controller should expose, and on the kinds of applications that can take advantage of those APIs.

The third area of benefit is concerned with the additional flexibility that comes from separating the control plane from the data plane and making use of a protocol such as OpenFlow between the two. This introduces two interesting possibilities, each of which can contribute to a superior solution for the networking infrastructure compared with conventional distributed routing protocols:

- The SDN control plane can be implemented in a centralized system that has a view across a large number of data plane elements and the transport links between them. The knowledge about network conditions available to such a centralized control plane element enables the implementation of new kinds of routing strategies that can potentially provide better outcomes for network traffic than those available from conventional distributed routing protocols such as OSPF or BGP.

- The SDN control plane may be able to form a holistic view of network conditions not just across a number of interconnected data plane elements, but also down through the layers of the network (e.g. IP, MPLS, Ethernet, optical transport, lambdas). With such a comprehensive view of network traffic at all the different layers, the SDN control plane may be able to optimize the network configuration to achieve far greater efficiency than is possible with current methods of provisioning each layer independently.

Realizing these kinds of optimized routing schemes that are enabled by SDN is going to take quite some time – much of the thinking required is still at the research stage. So for the time being, network operators should focus on the nearer-term benefits of SDN, namely reduced equipment costs and more dynamic service provisioning enabled by SDN APIs.

How do NFV and SDN relate to each other?

The relationship between NFV and SDN is multifaceted:

- SDN techniques are often used to program the cloud networking infrastructure to interconnect Virtualized Network Functions (VNFs) when deploying an NFV-based service. For example, when working with OpenStack as the cloud management software, the OpenStack component known as Neutron (formerly Quantum) may include an SDN controller plug-in that uses OpenFlow to program the physical switching infrastructure and the virtual switches associated with hypervisors so as to create the subnets and routing rules required to deploy the VNFs.
- SDN control plane elements may be deployed as VNFs in an NFV infrastructure.
- In some circumstances, for example when small scale routing or switching functions are required, both control plane and data plane elements may be implemented in software and deployed together as VNFs in an NFV infrastructure, so as to perform the routing or switching task entirely in software.

At a very high level, NFV and SDN have much in common. The central idea is the separation of hardware and software in the network, and the possibility to leverage low-cost industry-standard commodity hardware with independently developed software. Both NFV and SDN also envisage a high degree of automation in the deployment and management of network services, made possible by running NFV and SDN software in virtualized or cloud environments. SDN goes beyond NFV in that it introduces the possibility of breaking out of the limitations imposed by traditional routing protocols, enabling routing and traffic optimization to be performed in novel ways. But realizing the benefits of this aspect of SDN will take time, and for the moment, most network operators are focused on the shorter-term benefits that are expected to come from NFV, and from more tactical aspects of SDN.

What does an NFV environment look like in practice?

The starting point for NFV is a cloud environment of the kind that is widely used today to support IT workloads. This includes three main ingredients:

- Commercial off-the-shelf servers
- A hypervisor such as KVM or ESXi
- A cloud management solution such as OpenStack or VMware vSphere

It is entirely possible to start out with NFV using existing products in these three categories, deploying Virtualized Network Functions just as if they were IT workloads. However, current cloud environments have significant

shortcomings in two areas, which need to be addressed in some degree in order to realize the full benefits of NFV – namely data plane throughput and operations automation or "orchestration."

What data plane throughput issues need to be addressed?

Most IT workloads such as Web serving, transaction processing and big data analytics are not particularly input/output intensive. The capacity of most virtualized IT applications is determined primarily by CPU utilization rather than network I/O.

In the NFV domain, there are two distinct kinds of workloads: control plane workloads and data plane workloads. Control plane workloads are concerned with processing signaling and control plane protocols such as BGP, OSPF and SIP, and in general these workloads are CPU-intensive and not limited by the I/O capabilities of the virtualization environment. Data plane workloads, by contrast, are concerned with the routing, switching, relaying or processing of network traffic payloads. These kinds of workloads are I/O intensive, often requiring some combination of total I/O bandwidth and packets-per-second throughput that is orders of magnitude greater than is typical of common IT workloads. Currently, all of the commercially available cloud computing environments impose significant limitations on the throughput of data plane workloads.

These limitations arise because virtualization introduces a layer of software between virtualized guest applications and host networking hardware, and because this layer of software has not (yet) been optimized for high bandwidth or high packet rate applications. This software effectively provides switching between the virtual network interface cards (NICs) that guest applications connect to and the physical NICs that connect the host system to the physical network, and is known as the vSwitch.

The vSwitch performance issue is very widely recognized by NFV industry players, and is being addressed by multiple vendors. There are two contrasting approaches to solving the problem, and we are likely to see both approaches succeeding in the market.

The first approach is essentially to take the vSwitch software out of the path between virtualized guest applications and the physical networking hardware. One way to do this is to use NICs that present multiple virtualized hardware interfaces toward guest applications, so that the guest application binds directly to a hardware interface on the NIC rather than to a software-based virtual NIC. There's a standard for this known as Single Root I/O Virtualization (SR-IOV), and it's supported on many current Ethernet NICs. However, SR-IOV is a relatively new technology, and is currently not well supported by cloud management software solutions. This means that although you can usually directly configure a hypervisor such as KVM or ESXi to permit guest applications to take advantage of SR-IOV, you can't necessarily perform all of the low-level configuration necessary to achieve maximum data plane throughput via cloud stacks such as OpenStack or vSphere. But it's only a matter of time for cloud stacks to catch up and offer full support for SR-IOV.

The second approach is to optimize the performance of the software vSwitch for high bandwidth and high packet rate applications. In general, this involves re-architecting the vSwitch software so as to reduce or even eliminate packet copying operations in the data path through the vSwitch between the physical NIC and the virtual NIC that supports network connectivity at the guest application. Since cloud stacks already deal with the configuration of the software vSwitch, this approach to data plane performance improvement doesn't require anything new of the cloud stack.

Both of these approaches can deliver levels of data plane performance that make NFV an economically sound proposition for the great majority of virtualized network functions. SR-IOV is known to deliver data plane performance that is very close to that of bare metal, and while the outlook for vSwitch software acceleration suggests substantially less impressive performance gains, it may offer some compensating benefits in terms of improved cloud flexibility.

What's the role of operations automation in NFV?

Existing cloud management tools provide quite sophisticated operations management support in an IT environment, but arguably they lack certain functions that may be valuable in a carrier networking NFV environment, particularly in the area known as "orchestration."

Orchestration is concerned with automation of the life-cycle management operations for Virtualized Network Functions, and includes the following:

- Service instantiation – deployment of the VNF software components and configuration of the virtual network infrastructure so as to create an instance of a network service.
- Service component health monitoring and repair – monitoring of the virtual machines that are running VNF software components, reporting on errors and failures, and performing repair operations when a component failure is detected.
- Elastic scaling – continuous monitoring of the load on VNF software components that support a given service, and dynamically increasing or reducing the population of VNF component instances in response to changing load conditions so as to maximize the usage efficiency of the NFV hardware.
- Virtual machine migration – managing the movement of VNF software instances off a given host in order to enable that host to be taken out of service, for example for maintenance.
- Software upgrade – managing the in-service upgrade of VNF software that supports a given service.
- License management – tracking the usage of VNF software for vendor revenue assurance purposes.
- Service termination – gracefully shutting down a service when it is no longer needed.

All of these operations could, in principle, be performed manually via a cloud management console. However, automation of frequently performed life-cycle management operations will substantially reduce operational costs and is also likely to reduce the likelihood of human error during the performance of operations.

The value of automating any given aspect of NFV operations management depends to a considerable degree on the nature of the service being deployed on NFV. For example, a large-scale multi-tenanted service such as Voice over LTE supported by IMS is instantiated only very infrequently, so automating service instantiation in this case is likely to be a low priority. Furthermore, the load on such a service is likely to be highly predictable, so automatic elastic scaling may not be a high priority. On the other hand, monitoring the health of the service and automatically repairing failed VNF component instances is likely to be very important. By contrast, some services such as enterprise VPNs are single-tenanted, so a service instantiation operation is required for each new customer of the service. There is obvious value in automating that operation, and possibly even enabling self-service instantiation by the customer via a Web portal.

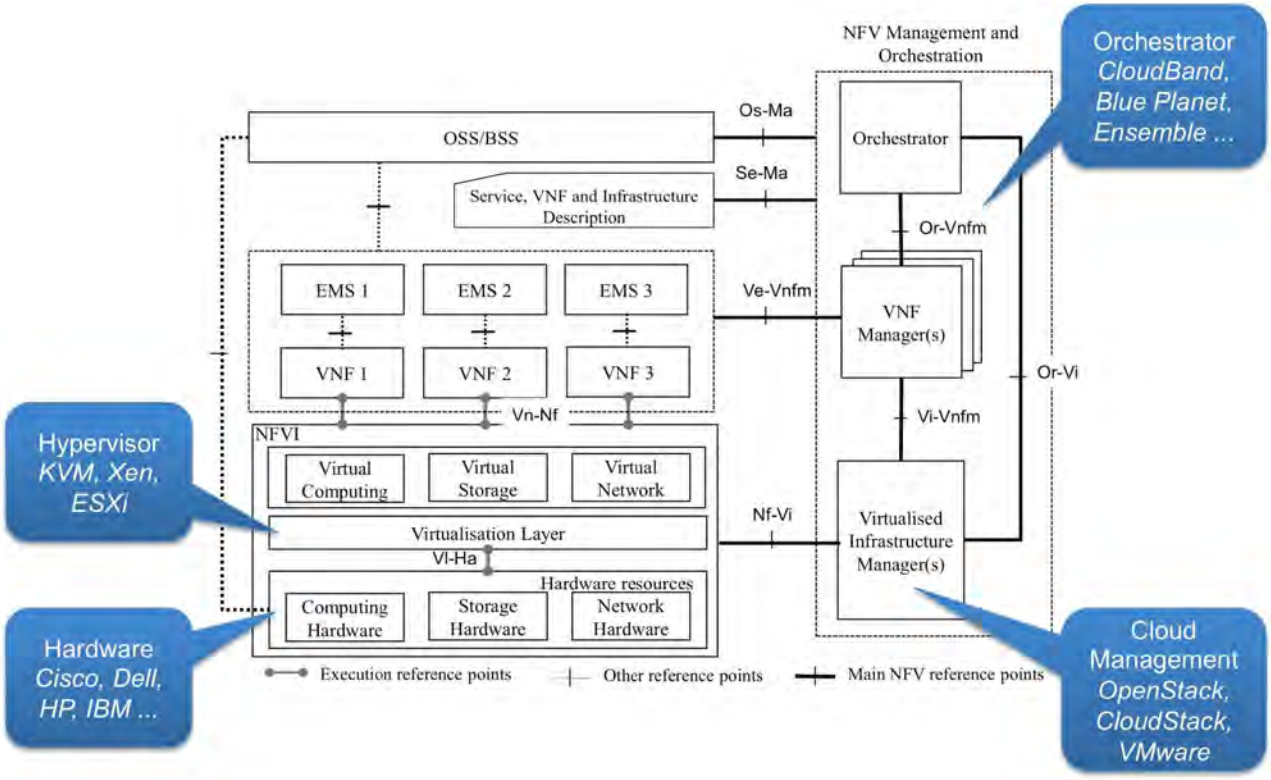

Figure 2: NFV Architecture and Key Elements

The concept of orchestration exists in IT clouds as well as in the world of NFV, but NFV demands rather more from orchestration. In the IT world, the focus is mainly on service instantiation, with some limited capabilities for monitoring and repair. The emerging NFV industry is starting to demand enhanced orchestration solutions that address additional aspects of VNF life-cycle management such as elastic scaling, optimized placement of VNFs relative to the underlying switching fabric and more fine-grained control of the switching fabric via SDN controllers.

When setting priorities for which network functions to deploy first in virtualized form in an NFV environment, network operators would do well to assess the kinds of orchestration that may be required in order successfully to manage those virtualized network functions. Given the relative immaturity of NFV orchestration solutions, it will make sense to focus on those VNFs that are least demanding in terms of management automation.

What impact will NFV have on OSS/BSS?

NFV will clearly have a major impact on the way in which network services are managed. Traditional OSS solutions are architected around the concept that network services are built on a set of appliances in which software and hardware are tightly integrated. In the NFV world, software and hardware need to be managed separately. Hardware comprises a more or less homogeneous pool of processing, switching and storage resources, while services are created entirely by deploying and configuring virtualized software elements. An NFV environment looks very different from a traditional network environment from the point of view of the OSS.

NFV also introduces a new set of operations management requirements in the area of orchestration. Concepts such as point-and-click service instantiation, elastic scaling and automatic recovery following hardware failure just don't exist in the traditional network environment. However, it's obvious that the OSS is going to have to deal with these concepts.

It's also worth pointing out that one of the key benefits of NFV to network operators is the ability to deploy new services more quickly by eliminating the need to qualify and approve new hardware. Traditionally, the time taken to integrate new network equipment into the OSS was a major factor in the timescales for deploying a new service. It should be much quicker to integrate new Virtualized Network Functions into the OSS because there is no need for the OSS to deal with any hardware management functions associated with the VNF, but some integration effort will be required to enable the OSS to configure the new VNF and handle fault reports and performance statistics. Ideally, the OSS should be designed to make it quick and easy to integrate new VNFs.

It's clear that an OSS solution that addresses the new realities of NFV is going to look very different from a traditional OSS. Indeed, the degree of change is so significant that network operators should seriously consider whether NFV justifies the introduction of a whole new generation of OSS. With many of the OSS solutions currently in use having their roots in network practice dating back 20 years or more, NFV might be the triggering event for an in-depth review of future OSS strategy.

What kinds of network functions does it make sense to virtualize?

Any network function that is capable of being deployed in the network over generic Ethernet interfaces can potentially be virtualized and deployed in an NFV environment. This obviously rules out network functions that depend on specialized physical interfaces such as optical transport devices, but it leaves a very long list of possibilities. In a white paper on NFV published in October 2012, network operators identified the following examples:

- Message Router (e.g. SMSC)
- Content Delivery Networks
- Session Border Controllers
- WAN Accelerators
- Deep Packet Inspection
- Firewall
- Carrier Grade NAT
- Tester / QoE Monitor
- SGSN / GGSN
- Provider Edge Router
- BRAS
- Radio Access Network Nodes

In general, it doesn't make financial sense to use NFV to replace existing physical network functions with virtualized equivalents, unless, for example, those physical network functions are at or near end-of-life. First steps in NFV are likely to be around areas of the network where new build-outs are planned. This is going to vary from one network operator to another, but some good examples of new network build-outs where NFV can make sense are:

- Evolved Packet Core (EPC) – mobile broadband access IP infrastructure for LTE
- IP Multimedia Subsystem (IMS) – SIP infrastructure for Voice over LTE
- Session Border Controllers (SBC) – for IP-based voice network interconnect and business voice access via SIP Trunking or Hosted PBX

Where is SDN likely to play first in the network?

Software-Defined Networking is already an established technology in many data centers. It provides the control and programmability of the virtualized networking environment, which is essential for rapid and secure deployment of virtualized IT workloads and Virtualized Network Functions alike. In this context, SDN is simply a tool that is leveraged by the cloud stack, and this usage of SDN to support NFV can be taken for granted.

In the wide area network, SDN promises to make it far easier to provision and configure network services such as VPNs. The general concept here is that the SDN controller element, which talks OpenFlow to dumb physical switching elements, will expose northbound APIs that the OSS/BSS can leverage for rapid service provisioning. The problem is that, in the real world, a VPN service for a given enterprise is likely to need to transit through existing network elements such as access and core routers that have a traditional architecture. This need for coexistence between SDN-based switches and routers and traditional devices may make it hard to realize the promised benefits of SDN. Meanwhile, traditional router and switch vendors are exposing SDN-like northbound APIs, so the promise of quick, automated service provisioning can perhaps be achieved by leveraging these APIs, and without needing to invest in

SDN as such. These kinds of arguments are making it increasingly difficult to see how SDN is going to find its way into existing wide area networks.

Meanwhile, a growing number of commodity "white box" L2/L3 switching appliances are starting to come onto the market. These need software to provide the control plane functions such as the support of routing protocols like BGP and IS-IS, and this software can be deployed either locally on the appliance itself (typically in a Linux-based operating environment) or remotely, using OpenFlow as the control mechanism. White-box switching solutions are typically not making any new claims about network programmability, but are simply offering a substitute for a router or switch product from one of the traditional vendors at a fraction of the price. While we might view this as a degenerate form of SDN that offers no advances in functional terms, it is nevertheless attractive precisely because it can come in alongside traditional network elements without any change in architecture or design practice, and deliver very substantial CapEx savings.

These CapEx savings may make for a very attractive business case for this type of SDN. But network operators should pay close attention to software quality in the control plane implementation. Popular open source router control plane software is not as mature as that embedded in router products from leading vendors, and it will take time before it offers comparable software quality. Thorough testing is therefore essential before systems based on open source control plane software are deployed in production networks.

What standards activities apply to NFV and SDN?

Innovation in the networking space is often associated with new activity in the area of standards. This is certainly true of SDN. OpenFlow is a new protocol associated with SDN, and while not all SDN implementations are based on OpenFlow, it is widely regarded as a key building block for SDN.

OpenFlow specifications are controlled by the Open Networking Foundation (ONF). The ONF is not one of the traditional standards bodies; it's an industry consortium formed in 2011 specifically to work on OpenFlow and related specifications in the SDN space.

The first OpenFlow specifications were published by the ONF in 2011, but the standard continues to evolve as implementers gain experience with it and as different SDN use cases exercise different aspects of the protocol. As of January 2014, a total of six versions of the OpenFlow spec have been published, and further enhancements are in the pipeline.

In the NFV space, it is not actually obvious that any new standards are required. NFV is about applying technologies from the IT world to support virtualized software-based network functions, and those technologies are, by and large, quite well established. However, the network operators that came together to publish the original white paper on NFV were anxious to promote the NFV movement and build momentum for it with other carriers and with network equipment vendors – so they created an industry consortium, the NFV Industry Specification Group, and chose ETSI to host it.

The NFV ISG has explicitly stated that developing new standards is not an objective – at least initially. The aim of the group has been to educate the industry about NFV and to provide a forum in which vendors and network operators can come together to agree on the key principles of NFV. To date, the NFV ISG has published documents on Virtualization Requirements, NFV Terminology, NFV Architectural Framework, NFV Use Cases and NFV Proofs of Concept Framework.

The NFV ISG has also established a number of working groups to create more detailed documents on Infrastructure, Management and Orchestration, Performance, Reliability, Security, and Software Architecture. As of January 2014, all of these working group documents are in the drafting stage.

The NFV Architecture document published by the ISG includes descriptions of the key functional elements of the NFV environment, and identifies reference points between these elements. These reference points might be viewed as candidates for new standards. But early real-world implementations of NFV make use of some combination of APIs exposed by existing software elements (such as OpenStack, which seems to be emerging as a de facto standard), together with existing standard protocols such as SNMP, to implement these reference points. It's certainly possible that ISG comes to the conclusion that some new standards are actually needed in order to achieve the full potential of NFV, but there's no real evidence yet that this is likely to happen.

How should network operators proceed in relation to NFV and SDN?

Network operators should identify the areas of the network in which significant new investment is likely to be made in the next two years, and investigate whether NFV could be a feasible alternative approach to a conventional network build-out. The following questions should be asked:

- Is it possible to source good quality VNF software that implements the network functions that are needed?
- What is the minimum set of operational automation / orchestration functions that is realistically necessary to manage the deployment of the relevant VNFs?
- Is it possible to source or build software to perform the required orchestration functions in a sensible timescale and at a sensible cost?
- Is it going to be possible to integrate the VNFs with the existing OSS/BSS, or will it be necessary to invest in some new management solution?
- How does the total cost of an NFV-based deployment compare with a traditional deployment?

When making the business case for NFV, it's worth remembering the future savings that will arise when additional network functions are deployed on an existing and established NFV infrastructure. If the entire cost of putting this infrastructure in place is assigned to the very first network function to be virtualized, the business case for NFV may not compare very favorably with a conventional network deployment.

Regarding SDN, network operators should keep a close eye on the evolving market for very low-cost commodity switching hardware that can be combined with open source or commercial control plane software to provide alternatives to traditional switching and routing products, and identify possible CapEx savings arising from the deployment of such solutions in growth areas of the network. Network operators should also explore their options for automating the provisioning of network services such as VPNs, comparing and contrasting SDN-based solutions with alternative approaches that leverage the APIs exposed by their existing switches and routers.

What impact will NFV and SDN have over the next five years?

The advent of NFV and SDN represents perhaps the single most important technology event in the telecom industry since the arrival of digital switching. Network operators who choose fully to embrace NFV and SDN have the opportunity to radically transform their businesses both to reduce their cost base, and to become far more agile in their ability to introduce new services. In short, to become "software telcos."

In five years' time, we can expect to see those network operators enjoying the benefits of NFV and SDN as follows:

Almost all new network build-outs are based on highly commoditized hardware, combined with best-of-breed software – often sourced from specialist vendors. This will be true not just of network functions that can be deployed on standard x86 server hardware, but also switching and routing elements (based on white box switch hardware) and even optical transport devices. Major reductions in network CapEx will result from this.

Almost all new services are created by deploying new software elements in the network, or by combining existing software elements in new ways. It will be possible to prototype new services in a matter of days, and bring new services to market in weeks. It will be possible to respond to feedback about new services far more quickly than it is currently. New services that are successful can be scaled very rapidly by leveraging existing commoditized hardware, while new services that fail to gain traction can be stood down without the loss of major investments.

Automation of service deployment reduces lead times for delivering services by orders of magnitude. Putting in place an enterprise VPN service for a given customer, for example, may take weeks. New business process applications are able to leverage the programmability of the network infrastructure using the new APIs exposed by it, shortening that provisioning time to hours, and even permitting a degree of customer self-service provisioning.

The flexibility of a virtualized infrastructure enables network operators to offer entirely new types of services. For example, a network operator can address the needs of an enterprise customer that wants some specialized communications service by creating a virtualized service instance specifically for that customer, and customizing it accordingly. This can be accomplished far more quickly, easily and safely than attempting to satisfy specialized needs by making incremental software enhancements to some common, shared service platform based on proprietary hardware.

Standardization of hardware and automation of operations management dramatically reduces operational costs. For example, in a conventional network, the failure of a piece of hardware is typically not service-affecting, but it requires urgent action to replace the failed hardware in order to restore the proper level of fault tolerance. With NFV, the failure of a piece of hardware has no more impact than a temporary reduction in the maximum capacity of a given service, which is accommodated within planned hardware capacity headroom. There is no urgency to replace the failed hardware, and the procedure for doing so is extremely simple and completely standardized.

This vision of the future will only be achieved by network operators who embrace NFV and SDN fully and are prepared to implement fundamental organizational transformation to make the most of what they enable. This will not be easy: decades of procuring and building networks in the traditional way have created a culture inside most network operators that may put up strong resistance to the kinds of changes that are needed. Resistance may take many forms, including persistent disbelief that cloud-based software can ever deliver carrier-class five-nines service availability, inability to trust any vendor other than the traditional suppliers to deliver telco-grade solutions, and reluctance to acquire the new skills necessary to build and manage services in cloud-based software environment. Resistance will

also arise from the simple observation that, if NFV and SDN are to deliver major savings in operational costs, that means a lot fewer heads involved in operations.

Network operators that successfully overcome these challenges and make the most of NFV and SDN will find themselves in a far better position than they are in today to compete with over-the-top services and with aggressive new entrants into the network business. We are at the dawn of an era of dramatic change in our industry. Never before have we seen such strong pressure to evolve, and to evolve rapidly. NFV and SDN are the tools we need to enable that evolution, and we ignore them at our peril.

What other sources of information are there on NFV and SDN?

The following links may be helpful in providing further information on these topics.

ETSI NFV ISG http://www.etsi.org/technologies-clusters/technologies/nfv

Open Networking Foundation https://www.opennetworking.org/

OpenStack https://www.openstack.org/

CloudNFV http://www.cloudnfv.com/

SDN Central http://www.sdncentral.com/