Independent market research and competitive analysis of next-generation business and technology solutions for service providers and vendors

HEAVY READING
WHITE PAPER

Fulfilling the Promise of NFV: Infrastructure & Software Implementation Best Practices

A Heavy Reading white paper produced for Napatech A/S

napatech

AUTHOR: JIM HODGES, PRINCIPAL ANALYST, HEAVY READING

INTRODUCTION

After some 49 months of intensive activity, network functions virtualization (NFV) has transitioned from a concept defined in a white paper to commercial reality. But while progress has been formidable, given the magnitude of the changes and challenges, additional steps and decisions must be taken to achieve the total promise of NFV.

One area we see as falling into this category is defining the interaction between the hardware and software realms. Conceptually, the NFV "endgame" views all applications running in a pure and independent software environment, as virtualized network functions (VNFs). Mirroring a traditional virtualized data center architecture, these VNFs run as virtual machine (VM) instances on commercial off-the-shelf (COTS) servers, anywhere in the cloud.

There is little doubt that this approach delivers considerable value, cost efficiencies and application agility. However, we believe that truly delivering the promise of NFV will require further advances in management and acceleration of applications running on cloud-based servers.

Accordingly, this white paper documents and evaluates the options associated with managing functional separation between software and hardware domains. This paper explores the commercial realities of deploying NFV infrastructure (NFVI), documents NFVI architecture deployment options and provides NFVI and virtualized software deployment best practices.

NFVI: THE COMMERCIAL REALITIES

As noted above, at the heart of NFV is the concept that software is totally liberated from any requirement to run on a specific hardware infrastructure. Yet market and technology forces are advancing so quickly that keeping pace presents several challenges that must be addressed. These challenges include implementing a common infrastructure; NFVI performance; and NFVI failover and resiliency.

Implementing a Common Infrastructure

One of the fundamental tenets of NFV is that the virtualized software instances will run on open COTS-based infrastructure. There are several value propositions associated with this approach. These range from stifling vendor lock-in, facilitating third-party software VNF development and integration, fostering open source adoption, service agility and automation, as well as optimizing infrastructure capex by utilizing common shared compute platforms.

Fundamentals aside, it is important to note that while the concept of COTS is well understood, depending on the time continuum selected, the actual platform characteristics and performance attributes can vary significantly. For example, some 15 years ago, COTS platforms typically deployed were likely aligned with CompactPCI (cPCI) design standards.

Advance the schedule to 2007, and cPCI gave way to a new COTS hardware specification – Advanced Telecommunications Computing Architecture (ATCA). While ATCA remains relevant in an NFV context, x86 rackmount servers are often identified as the COTS server of choice, since they have been deployed on a large scale in data centers. Still, given the dynamic and changing requirements of cloud-based compute platforms, it is highly likely that

additional evolutions will occur that will reshape COTS architecture requirements and software support capabilities.

Another way to look at this is that COTS will continue to develop additional software support capabilities in the future just as it has in the past, which means the very definition COTS infrastructure will remain somewhat malleable.

NFVI Performance

Although at first glance running VNFs on x86 servers seems straightforward, the unique requirements of telecom VNFs can introduce performance issues. There are a number of factors to be addressed. The first is that telecom applications and associated VNFs are extremely sensitive to latency, and x86 NFVI infrastructure was not designed with this in mind. Factor in the overhead associated with the addition of a hypervisor, and there is even greater potential for application performance degradation.

Add the impact of a MANO orchestrator to manage VNF lifecycle into the NFVI performance equation, and it's clear that telecom NFVI has very stringent and high performance levels to meet. Moreover, NFVI performance requirements are expected only to continue to tighten, given that 5G will reduce RAN latency, which must be matched by core cloud performance to enhance customer experience.

In parallel, the fixed domain is also moving to adopt an ultra-low-latency model associated with the introduction and support of the tactile Internet. In addition to these new performance requirements, the mass deployment of billions of Internet of Things (IoT) devices at the network edge will effectively mean that cloud-scale performance boundaries and thresholds will also have to be recalibrated.

NFVI Failover & Resiliency

Moving telecom applications to the cloud does not mean the abandonment of carrier-grade five-nines performance metrics. Thus, NFVI also brings new failover and resiliency requirements to the table to meet or even exceed current telecom platform requirements. Likewise, end users expect their service provider-sourced applications to have seamless failover and resiliency, regardless of the underlying technology.

Both failover and resiliency will represent even greater challenges for NFVI as more VNFs, or even decomposed microservices, are deployed and orchestrated at the network edge.

NFVI INFRASTRUCTURE STRATEGIES: ASSESSING THE ALTERNATIVES

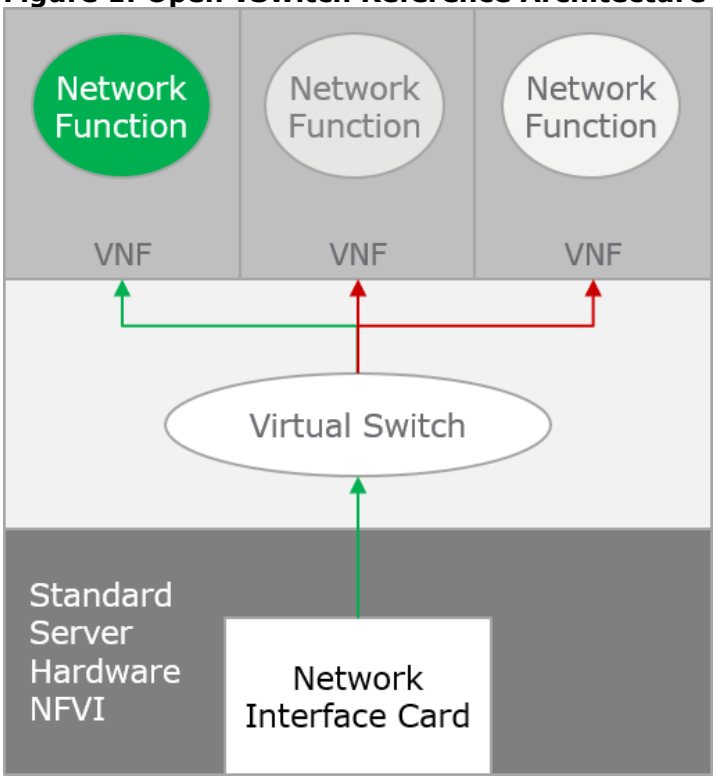
One of the key decision points that must be addressed to optimally support software-based VNFs is to define an NFVI infrastructure deployment strategy. This will require that CSPs evaluate a number of NFVI design alternatives. Essentially, there are three viable design approaches: Open Virtual Switch (OVS), Single Root Input/Output Virtualization (SR-IOV) and OVS Acceleration.

Open vSwitch (OVS)

As the name suggests, Open vSwitch is a community-led, open source-based virtual switch designed to manage VMs in a virtualized environment. Therefore, by default it is also relevant for managing telecom VNFs.

As shown in **Figure 1**, OVS manages these VNFs by managing traffic and control plane requirements between a standard network interface card (NIC) deployed on the NFVI and the VNF itself. Since OVS can run under hypervisor control, it aligns with NFV cloud reference architecture requirements.

Figure 1: Open vSwitch Reference Architecture



Source: Heavy Reading/Napatech

There are a number of benefits associated with implementing OVS. The first is that OVS has been adopted on a large scale by server, hypervisor and chipset vendors and continues to be enhanced via an ongoing release program. Since it is open source-based, it also minimizes the impact of vendor lock-in on NFVI platforms (including NICs). In addition, OVS is proven and mature, in that it has been deployed in some very large cloud deployments outside of the telecom realm.

However, in telecom clouds – in part because of the additional NFVI requirements documented earlier – OVS has shown that in some implementations, under load, due to VNF and hypervisor overhead, as illustrated in **Figure 1**, OVS can be severely capacity-constrained and unable to manage VNF/VM requirements.

In some implementations, OVS runs in kernel mode, which presents performance challenges due to competing resources. In other implementations, OVS is run in the user space along

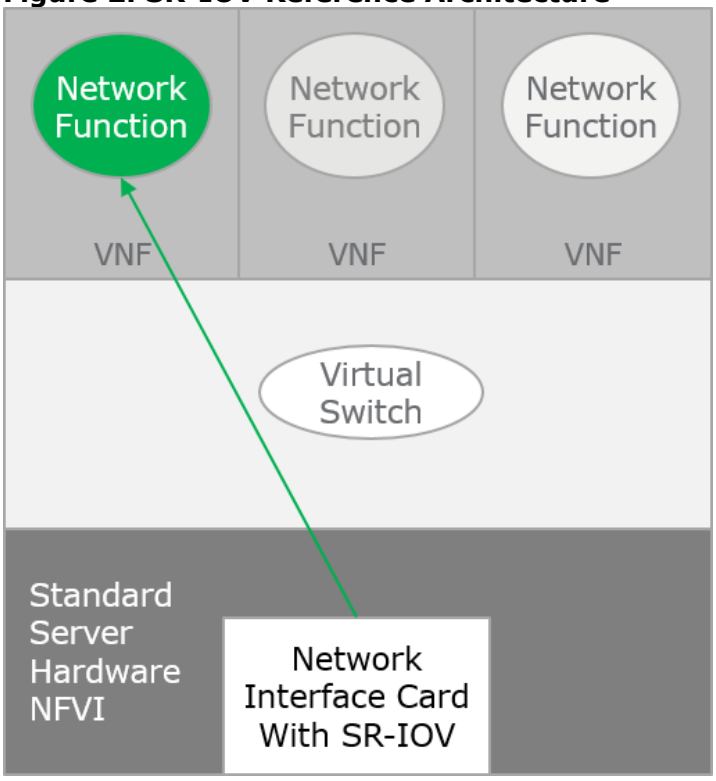
with DPDK, which improves throughput performance but requires additional CPU resources to meet high-speed applications (e.g., 100 Gbit/s). So, while OVS in conjunction with DPDK in user space provides a solution that is aligned with the needs of NFV with respect to virtual function mobility, automation and ease of management, the fact that full throughput is not achievable and that considerable CPU resources are required to implement virtual switching at high speeds is considered a hindrance to deployment.

SR-IOV

Given the limitations of OVS, an alternative approach has been proposed by some industry participants. Single Root Input/Output Virtualization (SR-IOV) is one of several methods of sharing physical input/output ports among multiple applications. Essentially, SR-IOV makes each application think that it has full control over the ports available. In this regard, the port is "virtualized" from a control perspective.

While this approach is new in a telco cloud context, SR-IOV dates to the previously noted PCI-based infrastructure era. Conceptually, as shown in **Figure 2**, SR-IOV enhances performance by bypassing the hypervisor and OVS to create a virtual connection over the server PCI bus between the VNF and the NIC. Several of these virtual connections can be made from the NIC to each VNF that needs to be supported, with each VNF believing that it has sole access to the NIC's ports.

Figure 2: SR-IOV Reference Architecture



Source: Heavy Reading/Napatech

Because OVS is not used to switch data, there is virtually no CPU load associated with delivering data to the VNF; the data is delivered over the PCI bus directly to the memory of the

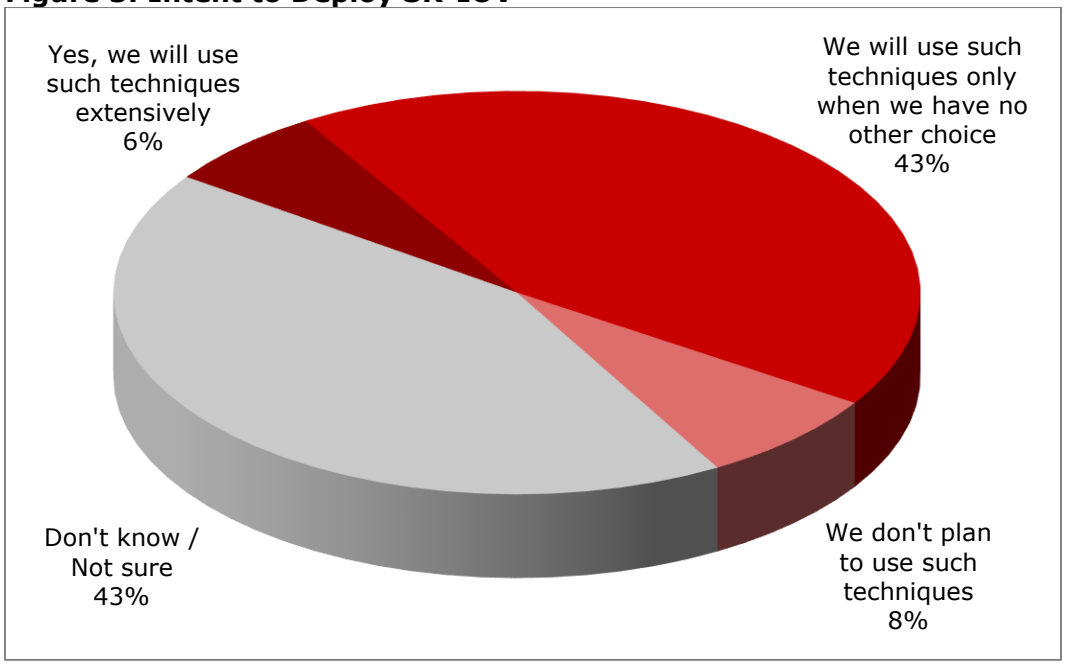
VNF. With this approach, it is possible to improve performance and reduce CPU load, which were the main concerns of OVS implementations.

Although SR-IOV does deliver enhanced performance, there are trade-offs as well. These stem from the fact that the hypervisor is not utilized to manage the spinning up of VNFs, which makes MANO orchestration and even failover recovery more complex. It is effectively bypassed, as data is delivered directly from the NIC to the VNF. This also means that the "virtualization" layer has now been bypassed, and the state of communication between the NIC and VNF needs to be managed by the NIC, which makes migration of VNFs from one server to another difficult.

Similarly, this means the concept of VNF portability and resource sharing on a pool of NFVI is compromised, since there is no longer a single logical virtual switch that all server resources can utilize. The final consideration is that SR-IOV support must be implemented on the NIC card, which can vary from one vendor to another.

Given its complexity and limited flexibility, service providers have been reluctant to adopt SR-IOV. As shown in **Figure 3**, this was first noted in a **Heavy Reading 2Q15 NFV Multi-Client Study**, in which 43 percent of survey respondents stated they would only deploy SR-IOV when no other options were available. One of the concerns raised by several respondents was the additional complexity that SR-IOV introduced.

Figure 3: Intent to Deploy SR-IOV



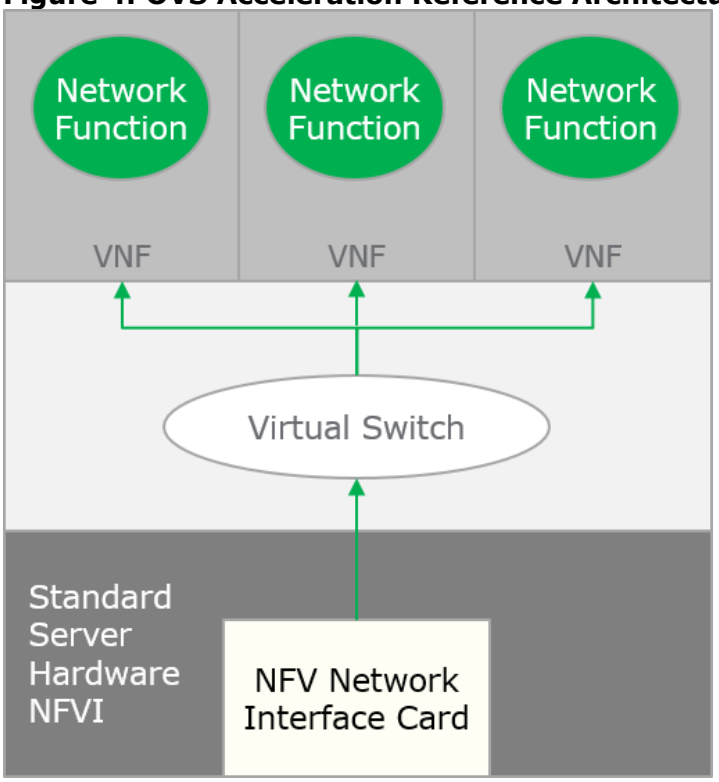
Source; Heavy Reading NFV Multi-Client Study, 2Q15
Does your company plan to use techniques such as SR-IOV to improve forwarding performance, even though such techniques can "violate" cloud principles of application portability? (n=93)

So, while SR-IOV does address the immediate concern of OVS performance and CPU resource usage, it does so in a way that can limit virtualization flexibility, while also making MANO and automation in general more complex.

OVS Acceleration

The third option that is now emerging is referred to as OVS Acceleration. Practically, this approach is designed to address the capability concerns inherent with OVS, while avoiding the complexity and architecture limitations associated with SR-IOV. As shown in **Figure 4**, this is accomplished by maintaining the basic OVS architecture and enhancing it with an NFV NIC, which accelerates OVS performance.

Figure 4: OVS Acceleration Reference Architecture



Source: Heavy Reading/Napatech

Principally, the NFV NIC card batches together IP packets as a single entity, which greatly reduces the overhead and ultimately the number of CPU cores required for OVS to switch and deliver the packets to VNFs. In turn, this allows OVS to manage throughput, which means more VNFs can be supported. Like SR-IOV, this approach also requires the deployment of a specialized NIC card, at least for now. Since the NFV NIC is FPGA-based, it is also software-reconfigurable, making it well-suited for evolving application acceleration requirements as NFV matures.

As part of the solution, certain functions from OVS are implemented on the NFV NIC to enable the destination of packets to be determined upon reception, so that contiguous packet batches can be built for each VNF. These are then delivered either on a latency-threshold basis, or when the batch reaches the equivalent size of an Ethernet jumbo frame. When the batch is presented to OVS, the destination is already specified and OVS can deliver the entire batch as if it was a very large Ethernet frame or IP packet.

This approach supported by vendors such as Napatech greatly reduces the switching CPU resource usage without compromising latency requirements. Therefore, in many respects,

OVS acceleration addresses the limitations of both the OVS and SR-IOV approaches, while also providing high performance, low latency and full VNF mobility.

While the existence of three distinct implementation approaches may be viewed as adding additional complexity to NFVI deployment, it is important to note that these options can be deployed independently, and even at various places in the network, based on throughput and application performance requirements. This means that service operators are not bound by a single architecture option and can test and deploy the option that best meets the performance, failover and resiliency requirements discussed in the previous section. The other value of this approach is that having several options affords the greatest level of platform flexibility to meet future evolutionary, yet-to-be-defined NFVI requirements.

NFVI & SOFTWARE DEPLOYMENT BEST PRACTICES

As documented, there are several independent architecture paths and options for deploying NFVI to support VNF/VM-based applications. This section documents six best practices that are designed to assist service providers in making the correct software and infrastructure decisions for deploying optimized NFVl platforms.

Best Practice 1: Define Infrastructure Performance Requirements in a Capex Context

While defining NFVI performance requirements represents a logical starting point, Heavy Reading believes this practice must be extended to define performance to tangible and achievable cost-saving metrics. Rather than simply relying on the elasticity of the cloud, we believe there must be an approach to map the various levels of elasticity associated with the NFVI options to tangible savings targets.

Specifically, what needs to be done is to model the various NFVI options and create a profile of what reduced performance means from a server perspective, compared to the positive impact of exponential performance. While these benchmarks will vary depending on traffic flow patterns and cloud configuration, savings can be considerable. For example, a study conducted by Napatech found that opex and capex savings for a 10,000-server data center can reach as much as $8 million over a four-year period using the OVS acceleration approach.

Adopting this best practice will also be vital for justifying an IoT deployment business case by assessing the cost of transporting IoT data to a metric such as "revenue per bit." As it currently stands, there is little disagreement within the industry that IoT will ramp up network traffic levels in the cloud and open new market segments for service providers. But there is still very little data regarding what the breakeven point will be to process this traffic. Of course, it depends on the traffic levels and application characteristics, but these factors need to be modeled to validate success vs. failure thresholds.

Best Practice 2: Create a Holistic Acceleration Plan

As documented, there are three valid options for managing and accelerating virtualized applications. Therefore, it is vital that service providers develop and implement a holistic acceleration plan that considers all acceleration requirements, based on where the application is running in the network (e.g., edge vs. centralized cloud).

The key to this best practice is to ensure that NFVI and software resources are positioned in the cloud so that they are optimized to meet transport and traffic demands using a mix of hardware or software acceleration. Stated another way, the creation of an effective acceleration plan requires that the impact of cloud-shared resources be considered holistically, rather than simply making a decision to only implement one form of acceleration (e.g., hardware only vs. software).

Most importantly, the choice of acceleration solution should not compromise the overall benefits of implementing a virtualized infrastructure. Inherently, it is better to process in software and server CPUs as much as possible, as this provides the greatest flexibility and ease of use. However, as CPU resources should primarily be used to host virtual functions rather than delivering and pre-processing data, hardware acceleration via programmable FPGAs can deliver considerable value, since it can be implemented utilizing a software-based approach that does not compromise VNF mobility or flexibility.

Best Practice 3: Define Processing Requirements

In addition to defining performance and acceleration requirements, we believe it is also important to define processing requirements. While processing requirements do not at first stand out, early trials and service deployments have confirmed that the telco cloud traits translate into unique processing requirements. There are several factors at play here. First, there are the latency, failover and resiliency requirements previously noted, as well as some emerging considerations.

We include the adoption of microservices in this latter category. Since microservices are "stateless" VM-based applications and can be deployed anywhere in the network, they present unique processing requirements. Thus, additional levels of processing capacity are needed to facilitate data manipulation at a bit level, based on VM performance requirements.

Although this requirement is still being defined, initial experience has highlighted the value proposition of utilizing integrated FGPA-based designs into the reference architecture. Another way to look at the impact is that microservices will likely be one of the triggers that will fuel the next phase of NFVI evolution.

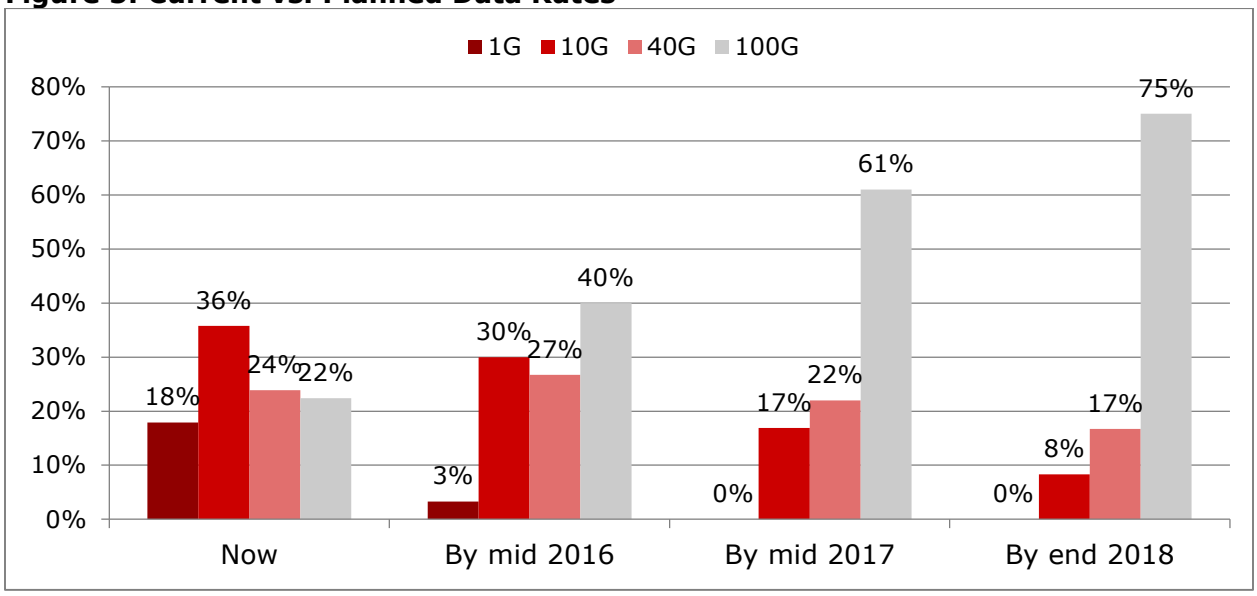
Best Practice 4: Assume High-Rate Line Speeds

The best practice of assuming high-rate line speeds may seem overly simplistic, but in practice, defining how fast is fast enough can be a challenge. A case in point in the adoption of 100G technology in the core, metro and access networks.

In order to gauge this shift, Heavy Reading conducted a custom research project in 2Q15 on behalf of Napatech. As shown in **Figure 5**, the survey identified strong support for and commitment to deploy 100G rather than 40G in the core.

Some 20 months ago, this rapid adoption scenario seemed initially unlikely. But upon further investigation, it was revealed that transport speeds, like infrastructure, were entering a new evolution phase that would result in a sharp decline in deployments of 10G and 40G in the core by 2018. While this steep adoption curve makes sense, given the other technology trends noted as well as vendor commitment to 100G, the key takeaway is that the considerable pent-up demand for high-speed access must be factored into the cloud migration and preferred NFVI reference architecture(s).

Figure 5: Current vs. Planned Data Rates



Source: Heavy Reading Napatech Custom Research Survey Q215.
What are your company's most common current and planned data rates for its core transport network? N=60-64

Best Practice 5: Exploit the Value of Automation Testing

Another best practice that we believe must be considered is to exploit the value of automation testing. In an NFVI context, automation refers to the ability to automate provisioning and testing of virtualized applications. The value of this is inherently clear, since it drastically reduces the time required for creating and introducing virtualized applications.

However, the value also can be extended to leverage automated testing to validate overall cloud performance metrics, ability to meet load and manage VNF failover. Given the concept of building networks to the high-hour, high-busy-day requirements is less valid in a cloud environment, automation testing tools must be more dynamic and configurable.

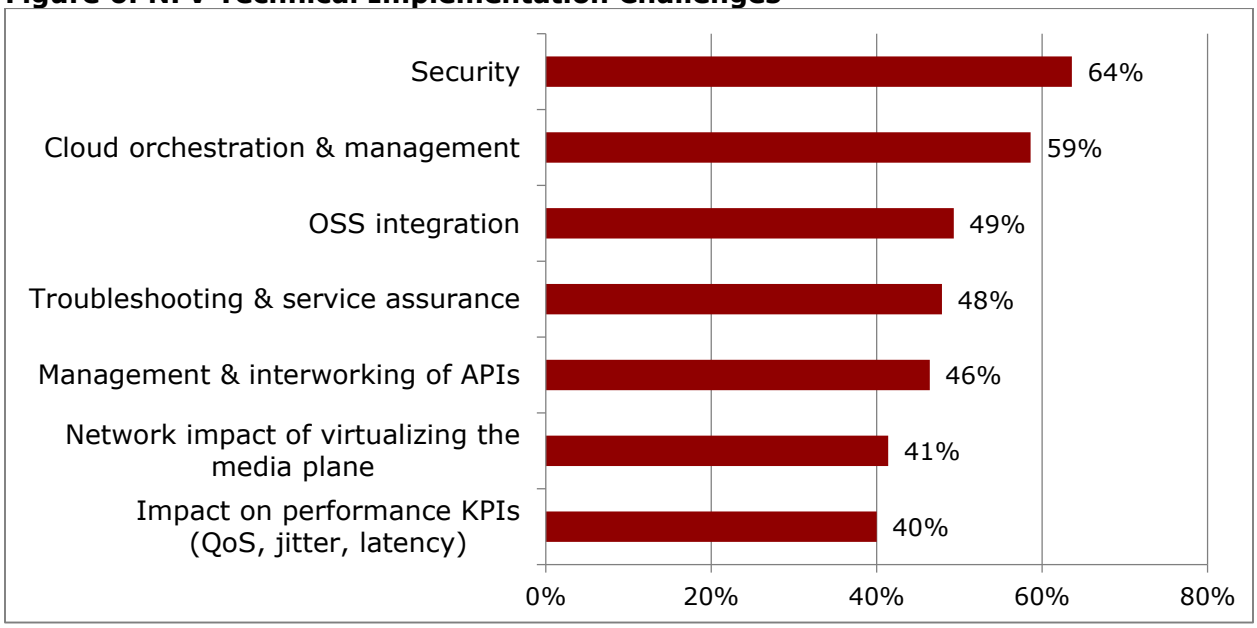
In turn, this means the NFVI architectures must be sufficiently configurable to enable running of new tools and third-party software. Once again, we see integrating support of FPGAs into NFVI as delivering the requisite levels of flexibility without taxing CPU throughput.

Best Practice 6: Be Prepared for Evolving Security Requirements

Service providers are extremely concerned about security and changing security requirements. As shown in **Figure 6**, security topped the list when service providers were asked about the biggest technical challenges related to NFV that must still be resolved.

Perhaps the greatest challenge regarding security is that all the requirements have yet to be completely internalized and defined. Moreover, these requirements are not static but extremely dynamic, and will unquestionably continue to evolve as new threats emerge. These factors in turn have a major impact on NFVI reference architecture requirements, as well. Therefore, we believe security requirements will also drive another NFVI evolutionary cycle in the very near future.

Figure 6: NFV Technical Implementation Challenges



What are the biggest technical challenges related to NFV that still must be resolved? (choose all that apply) N=141
Source: Heavy Reading NFV Tracker, September 2016

While it is still early days, so far we have learned that NFV security enforcement can be CPU-heavy from a packet processing and encryption perspective. Therefore, the selection of an NFVI architecture strategy must be focused on ways to offload CPU security requirements in software or on advanced NIC cards to meet immediate and fluid future security requirements.

How well current NFVI-based servers can meet the challenge remains to be fully assessed, but initial indications are that x86 servers are not the optimal platform to meet these requirements for massive cloud deployments, and are therefore subject to the impact of platform evolution. In tests conducted with clients, Napatech found that software-based encryption typically consumes one CPU core per 1 Gbit/s of traffic to be encrypted or decrypted. At 40 Gbit/s, this is basically an entire server, which represents a heavy overhead cost. In contrast, Napatech's testing utilizing FPGAs in the same baseline identified that performance and throughput could be improved up to 30 or 40 times.

CONCLUSION

While NFV continues to move ahead into the commercial readiness phase, key technology decisions and challenges must be addressed. As this white paper has documented, one key consideration is the implementation of a flexible, programmable and scalable NFVI architecture. Fortunately, there are several viable alternatives that can be vetted through the application of best practices to provide service providers of all sizes with the implementation programmability flexibility they require to meet the continually evolving and fluid infrastructure requirements of the cloud.