# TABLE OF CONTENTS

1 Introduction ........................................................................................................................................... 1

2 Target Operational Environment ........................................................................................................ 3
   2.1 Multi-VNF Environment at the Service Provider ........................................................................... 3
   2.2 Multi-Vendor within and Across Providers .................................................................................. 6
   2.3 Target OSS/BSS Environment ...................................................................................................... 6
   2.4 An Agile Target Operational Environment ................................................................................... 7
   2.5 Controlled Agility .......................................................................................................................... 9
   2.6 Target Development Operations Environment ............................................................................. 9

3 Operational Issues and Recommendations ....................................................................................... 11
   3.1 Operator’s Operational Transformation ........................................................................................ 12
   3.2 Scale and State Management ........................................................................................................ 15
   3.4 Performance Management ............................................................................................................ 17
   3.5 VM Deployment and Migration Management ................................................................................. 19
   3.6 Security and Threat Management: Authentication, Encryption and Trust Across Virtualized Network Domains .................................................................................................................. 19
   3.7 Considerations for Using SDN within NFV ................................................................................ 21

4 Deployment Considerations ............................................................................................................... 24
   4.1 Non-intrusive and Gradual Implementation Approach to NFV in both Greenfield VNF and Combined legacy NF–VNF Environments ................................................................................... 26
      4.1.1 Greenfield Deployments ......................................................................................................... 28
      4.1.2 Offloading of Existing PNFs .................................................................................................... 28
      4.1.3 Virtualization of the Management Network .......................................................................... 29
      4.1.4 Cutover of Individual PNFs .................................................................................................. 30
   4.2 Open-Source Initiatives .................................................................................................................. 31
   4.3 DevOps – Automation in Deployment and Operationalization ....................................................... 31

Conclusions ................................................................................................................................................... 32

Acronyms ................................................................................................................................................... 33

Acknowledgements ................................................................................................................................... 33
INTRODUCTION

Future telecom networks will be designed using sets of aggregated capabilities coming from existing physical network functions and newly introduced virtualized equivalents, as well as evolving and new virtualized network functions. The goal is a virtualized and programmable network that has the ability to evolve and adapt to service providers’ emerging requirements.

To meet that goal, telecom networks are undergoing a massive transformation enabled by emerging technologies such as virtualization, Software-Defined Networking (SDN) and cloud. Several factors are driving this transformation, including the need for less complex and lower-cost network operations, shorter Time to Customer (TTC) and Time to Market (TTM) for new services, and new business opportunities built on the Anything-as-a-Service (XaaS) model.

This white paper focuses on the items that need to be addressed for the wireless industry to be comfortable with virtualizing their networks. It takes a hard look at some of the key real-world deployment and operational issues when moving toward network function virtualization (NFV). The paper will describe prerequisites to deliver predictable mission-critical applications based on NFV. In particular, it will examine key architectural aspects and functions delivered by management and orchestration. The paper also explores technologies such as SDN and their role, as well as community efforts such as Open Platform for NFV (OPNFV).

In 2014, the 4G Americas white paper “Bringing Network Function Virtualization to LTE” described NFV’s expected benefits, business case and roadmap considerations for planning and deployment of NFV-based solutions. It introduced and described the NFV architecture and ecosystem including use cases such as IMS and VoLTE and virtualized EPC.

NFV provides a new path that can increase the flexibility required by mobile service providers and network operators to adapt and accommodate a dynamic market environment with rapid capacity growth and increasing traffic diversity. NFV is a new operational approach applying well-known virtualization technologies to create a physical commercial off-the-shelf (COTS) distributed platform for the delivery of end-to-end services.

NFV’s key concept is the transformation of the network infrastructure and associated operational and business models. By applying virtualization technologies, the software of network functions can be broken apart from dedicated hardware. This separation unleashes massive flexibility in terms of how network functions (NFs) can be dynamically deployed, elastically resized, and offered on an on-demand basis. The ability to dynamically instantiate virtualized NFs (VNFs), create end-to-end services and independently scale the network infrastructure capacity to support emerging service demands, all frees service providers from the lengthy procurement and deployment timeframes associated with physical network functions.

Like NFV, SDN provides the ability to programmatically define and manage networks. This programmability empowers operators to take a holistic view of their networks so they can be dynamically configured end-to-end. Although SDN is not a mandatory technology for NFV, it is very complementary for several well-known reasons:

- SDN allows highly configurable and flexible networking to support dynamic virtual instances of network functions in an NFV system.

1 Bringing Network Function Virtualization to LTE, 4G Americas, November 2014.
- SDN brings the ability to dynamically chain together VNFs to realize a network service. Then it further is able to direct service flows of arbitrary granularity into those service chains.

- In addition to many synergies with NFV, SDN can act as an enabler for NFV. By separating out the control plane, SDN supports distributed virtualized switching functionality implemented completely in software running on top of standard hardware.

Of course, integrating SDN into NFV comes with its challenges. Starting with practical questions such as where to place the SDN controller in an NFV architecture soon leads to deeper questions, such as “in what manner.” SDN may be used for different purposes and in different ways at different layers within the NFV architecture. SDN can enable linking together multiple VNFs into a service chain, forming the network within a data center, and finally, powering the wide-area transport network between data centers and administrative domains. Operators need to decide at which levels to deploy SDN and determine the interactions between levels.

The original aim of combining NFV and SDN was to decouple services from resources. But when these two technologies come together, they provide the additional benefit of detaching lifecycle management from physical constraints. Today, it is possible to provision an SDN/NFV service instantaneously without the need to deploy new physical resources. This flexibility is the foundation of network agility. Operators need to decide upon the best possible architecture to deploy the SDN/NFV combined network function for achieving network agility.

Open-source efforts such as OpenStack, OpenDaylight and OPNFV are already working on integrating an SDN controller into the cloud infrastructure management. Specifically, these and other projects are in the midst of adding capabilities to define and manipulate service function chains and VNF-forwarding graphs. There are still challenges. One is figuring out how to harness SDN’s power across different administrative domains. Another is learning how to integrate SDN operation with management and orchestration functions. The operator's ultimate goal is to control and achieve end-to-end behaviors. However, sorting out these kinds of issues will be worth the effort. A carrier network with NFV and SDN tightly knitted together goes a long way toward yielding the benefits of service innovation/agility, operational improvements and cost reductions envisioned by the NFV industry.

The following sections will look at three main areas of concern. Section 2 will examine the target operational environment, which is expected to be a multi-vendor, multi-VNF, agile and controlled environment supporting DevOps ways of working while providing a deterministic behavior with regard to scale, performance and availability. Consequently, the importance of automation in managing this increasingly complex, versatile and agile environment will be addressed. In particular, section 2 will investigate the target OSS/BSS environment and implications of a multi-VNF and multi-vendor environment. Separate sub-sections will then look at agility and DevOps in the target environment.

Section 3 will address operational challenges when transforming operations to manage next-generation infrastructure. It takes a close look at managing VNFs’ scale and state because being able to easily scale network functions is an expected benefit but difficult to achieve initially. Tooling supporting management and operations will be equally addressed as performance and security considerations.

Section 4 discusses implementation paths for NFV in Greenfield deployments and combined legacy NF and virtualized environments highlighting a non-intrusive gradual implementation approach. This includes different approaches to moving workload from legacy to virtual and virtualization of the management
network. It closes with an outlook to automation in deployment and moving towards DevOps ways of working.

2 TARGET OPERATIONAL ENVIRONMENT

The target operational environment is expected to feature multi-vendor VNFs, VNF managers (VNFM), virtual infrastructure manager (VIM) orchestrators and NFV infrastructure (NFVI). In this multi-vendor environment where interfaces are not clearly defined, it is imperative to understand how deterministic behavior can be ensured with regards to scale, performance and availability. Consequently, the importance of automation in managing this increasingly complex, versatile and agile environment based on physical and logical (virtualized) network infrastructure needs to be understood. The following section explores this envisioned target operational environment.

2.1 MULTI-VNF ENVIRONMENT AT THE SERVICE PROVIDER

In the ETSI white paper, *Network Function Virtualisation, An Introduction* initial design objectives were developed to enable the VNFs to execute on an independently sourced COTS hardware infrastructure, as well support and enable open innovation, particularly in the development of new virtualized network functions. In this environment, service providers expect to be able to select VNFs developed by their vendor(s) of choice and execute them with the NFVI and multi-vendor management and orchestration functions deployed by the operator.

Figure 2.1 illustrates a service provider acquiring VNF A2 and VNF B5 from VNF provider A and VNF provider B, respectively. The service provider is then able to design and deploy an end-to-end network service using multi-vendor VNFs. In the case where VNF A2 and VNF B5 require VNF A2 VNF manager and VNF B5 VNF manager, respectively, as part of their deployment, these two VNF managers are attached to a single NFV orchestration (NFVO) platform. ETSI NFV-MAN describes these architecture deployment options.3

---

The acquisition of the VNFs and the design of network services are asynchronous operations. End-to-end network services can be developed only from the set of VNFs that a service provider has already acquired and ingested into its management systems.

Affinity rules and High Availability (HA) zones will play an important role in VNF deployments:

- Critical VNFs will always be deployed with some level of redundancy.
- Affinity rules will ensure that VNFs are running on separate hardware so that a single hardware failure would not affect all VNFs.
- For high-performance VNFs, CPU affinity rules would be important. These types of VNFs will fully utilize all CPUs, and sharing them with other VNFs can cause unpredictable problems.
- Certain VNFs may require specific NFVI resource requirements, such as hardware accelerators. In those cases, service providers should provide a method to define such requirements at deployment time. One example is the need for network interface (NIC) cards with special features such as single root input/output virtualization (SR-IOV).

Operators and service providers should ensure that the NFV infrastructure provides all the needed resources for optimal VNF execution. The main resources are network, CPU and storage, with the network providing appropriate levels of latency, jitter and bandwidth. The computing resources provide CPU resources, while storage provides the necessary amounts of input/output operations per second (IOPS) and capacity. Additionally, some VNF capabilities, such as live migration, would strongly depend on each operator’s unique configuration.
Service providers can use a variety of commercial arrangements to acquire the use of VNFs on a permanent or temporary basis because VNFs are software and thus can be licensed. The intellectual property rights associated with a software license are typically copyrights, though some other intellectual property rights (e.g., trademarks, trade secrets, patent rights) may also be impacted. A software license would be expected to have a number of terms similar to other commercial purchase or rental contracts for physical devices, such as agreements for how to handle disputes that arise under the contract. Some of the software license terms (e.g., those related to the performance warranties or indemnities) may be different from the contract terms associated with the acquisition of physical network functions (PNFs) to reflect the fact that the execution performance of VNFs may be impacted by other factors beyond the control of the VNF provider.

The TM Forum has developed some guidance for the procurement process. Other groups have envisaged various forms of application store. One example is the EU’s T-Nova Project. The NFV Industry Specification Group (ISG) considered that VNFs may be delivered in some cases in an “as-a-service” basis. The flexibility of software licensing arrangements for VNFs enables innovation in the business models for both VNF providers and service providers.

Acquired VNFs must be validated to ensure integrity and consistency with the associated VNF package metadata. The contents of the VNF package metadata are still being defined in ETSI NFV ISG and are expected to evolve over time as new use cases drive additional requirements. The initial VNF package metadata may provide little beyond the structure of the associated VNF images and some deployment constraints (e.g., affinity/anti-affinity rules for the VNF).

Service providers will need to validate proper functionality within the range of VNF configurations permitted and characterize the performance for the VNF configuration options that they wish to make available to their network service designers. This performance characterization may consider not just throughput in “normal” operational scenarios, but also assure proper operation during various failure scenarios, such as disconnected/misconnected I/O, hardware failures, chaos monkey operations that delete VNFC instances. This validation and characterization can take place on a suitable partition of the NFVI where the execution environment is comparable to the target execution environment. The results of this ingestion testing may create additional metadata associated with the VNF in the service provider’s management systems. Some of the validation and characterization may result in the development of additional tests that can be applied to validate proper operation of a VNF while it is in operation with live traffic. In some cases, the validation and characterization may be done in third-party laboratories to minimize the ingestion time. It is desired that to develop a suitable standards for the execution environment to enable the results to be translated into the service provider’s environment.

The design of a new end-to-end network service may require custom engineering to develop the service and test that it meets the desired design objectives for performance, cost, energy efficiency, etc. The multivendor context is reflected in requirements such as GEN.4 and Port.1 of “ETSI NFV; Virtualisation Requirements.” The design of end-to-end network services is a complex task that is not expected to be completely automated. The random interconnection of VNFs is unlikely to result in a useful end-to-end network service as not all the VNFs available to the service provider’s network service designer can be interconnected. Where VNFs have existing, well-defined, matching interfaces (e.g., client/server TCP/IP ports), interconnection of the VNFs may be possible.

---

4 TMF062 Procurement Support Process and Templates R14.5.0, TM FORUM.  
5 http://www.t-nova.eu/for-software-developers/  
Some network service design tools may be able to prevent interconnection of incompatible VNF interfaces. The internal interconnections of VNFCs should have been validated within the VNF ingestion process. There are a broad range of end-to-end network services that need to be supported over virtualized infrastructure. If service provider’s infrastructure is completely virtualized, then all of its services need to be provided in such a fashion. For a service provider that is used to operating as a mobile virtual network operator (MVNO), this may not be much of a change. But for integrated service providers – and not just of mobile services – other services such as Metro Ethernet would need to be created and supported in the same service environment. The availability of such service creation and design environments enables rapid innovation in new services permitting differentiation between service providers. If the service creation environment and appropriate resource authorizations are provided in a VNPaaS offering, it enables open innovation in network services.

2.2 MULTI-VENDOR WITHIN AND ACROSS PROVIDERS

The previous subsection addressed the reality of a multi-vendor environment within a service provider. However, this heterogeneity goes beyond the single provider. Only a few players are expected to own all the assets (e.g., VNF, network services) needed to create services that are attractive to end user. Typically, assets from different vendors and providers will be dynamically combined in partner-to-partner relationships. Operators will blend their capabilities together with partner assets to expose novel services. For example, operators can offer over-the-top (OTT)-aware communication services or third-party products may offer communication-aware services.

In multi-vendor environments, the orchestration layer should be made capable of provisioning the new interfaces automatically as new NFV-based applications are added or removed or to address capacity or performance needs in the network. It would be beneficial to have monitored authorization capability in the orchestration layer for deploying services.

2.3 TARGET OSS/BSS ENVIRONMENT

In the target operational environment, the multi-vendor bare metal and the virtualized infrastructure will coexist as the industry migrates to an SDN/NFV-based network architecture. The OSS/BSS network and the orchestration function need to interwork to provide the necessary service with the optimum level of performance.

The management and orchestration (MANO) specifications developed in the first phase of the ETSI NFV ISG MANO work focused only on the novel virtualization aspects building incrementally on the classic OSS/BSS infrastructure already defined to manage and orchestrate (physical) network services.

While MANO from the functional point-of-view groups all the management capabilities, in reality those functions are scattered across different OSS/BSS layers as illustrated Table 1.

---

8 *Bringing Network Function Virtualization to LTE*, 4G Americas, November 2014.
### Table 1. Distribution of MANO capabilities.

<table>
<thead>
<tr>
<th>Layer</th>
<th>MANO Mapping</th>
<th>Scope</th>
<th>Duration</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business/order</td>
<td>OSS/BSS (Classic NF Oriented)</td>
<td>Executes across business &amp; operations functions</td>
<td>Provides exception management (work lists) capabilities</td>
<td>Manages long-running customer-facing processes</td>
</tr>
<tr>
<td>Service control</td>
<td>NFVO(NS Orch) + EMS</td>
<td>Executes within the network domain, across controllers</td>
<td>Manages model-driven automated real-time network-facing processes</td>
<td>Provides service &amp; network abstractions to business/ops functions to simplify their interactions</td>
</tr>
<tr>
<td>Network control</td>
<td>VNFM + EMS</td>
<td>Executes within a single control plane to monitor and scale within its scope</td>
<td>Manages autonomic run-time processes</td>
<td>Manages the state of the service or network</td>
</tr>
<tr>
<td>Cloud/Infrastructure control</td>
<td>NFV (Aggregated RO) + VIM</td>
<td>Executes within the cloud domain</td>
<td>Manages real-time resource allocation and management</td>
<td>Provides resource abstractions to simplify resource management</td>
</tr>
</tbody>
</table>

The table highlights the challenge of having a MANO platform that spans across different OSS/BSS layers, which are called responsibility domains at times.

Today’s operational environment does not continue to scale and support the rapid service creation and deployment environment. However, future programmable operational environments with abstraction layer of service control should provide capabilities for modeling existing and future services in a more interactive fashion.

### 2.4 AN AGILE TARGET OPERATIONAL ENVIRONMENT

The target operational environment is more dynamic than traditional physical NF deployments. Where traditional physical network elements may have had multiple versions of cards deployed and operating in the field, the ease of version control of software components enables the service provider to maintain fewer variants of virtualized network functions.

Figure 2.2 illustrates how ideally there would be three variants of VNFs supported in a service providers’ network: the current version used in services, an old version maintained to enable rollback in case of problems detected in the current version and a new version being prepped for future deployments. Some
Service providers may keep additional old versions available as checkpoints to ensure service stability for some period of time.

With operational processes optimized to support continuous integration, version-upgrade-cycle times can be significantly reduced. A PNF version upgrade involves replacing cards in active devices and physical stocks, and there may be multiple versions of cards deployed in the network. Some old versions may still be serviceable for some functions.

With VNFs, it becomes an electronic deployment of a new VNF to perform an upgrade of a service deployment. This electronic deployment does not require truck rolls to deploy the new VNF version to the NFVI nodes in the field. Physical equipment upgrades are limited by the capital cost of the equipment and the human cycles for distribution of the new hardware. This results in hardware upgrade practices measured in years. If the VNFs are developed with a DevOps/continuous integration process, new versions may be available much more frequently. Quarterly release cycles are not uncommon for open-source projects using these DevOps/continuous integration methodologies. Information about proprietary development processes and cycle times is harder to obtain but likely to be comparable.

Frequent software upgrades would require more frequent software testing to validate proper functionality. That would be possible only with extensive automated testing. Additionally, it would require changes in the workflow of network operational team. The network operations center (NOC) should be able to deploy, validate and monitor new DevOps requirements. That would also require new deployment tools and new sophisticated monitoring systems.

While many VNF upgrades may be relatively minor releases incorporating bug fixes of various kinds, they still need to be accepted and ingested by the service provider. The acceptance testing needs to validate the VNF package’s integrity, the VNF’s functionality and performance and the operation of the new VNF version in the context of those network services that have been defined using this type of VNF. Much of this testing is in the nature of regression testing that can be automated and executed in a sandboxed section of the NFVI. Independent regression testing can be scaled out and performed in parallel, if necessary, resulting in a reduction in the calendar time for the ingestion processes.

The net effect of moving to VNFs should be a dramatic reduction in the time to onboard new network functionality, reducing the bottleneck of testing where serialization of hardware testing is enforced by limitations of dedicated laboratory space. New service introduction in the network could be validated with a small test environment with a few VNFs in a controlled manner in a production network to see their potential impact before they’re launched. This can be followed by enabling a much faster rate, with
perhaps tens of VNFs per week becoming available to the service provider’s network service designers. That permits a higher service velocity, with updates to existing services and introduction of new network functionality at a much faster rate.

Service providers are expected to develop similar old, current and next-generation versions for the network services that they develop based on the VNFs they’re authorized to use.

### 2.5 CONTROLLED AGILITY

Mission critical services have stringent service-level agreements (SLAs) to be met, and this is why an extensive ingestion testing and burnout phase is a must: to test the envelope conditions. Nevertheless here is where the virtualization of both NFV resources (NFVI) and networks with SDN can provide a suitable on-demand test environment to speed up the entire lifecycle.

Agility starts with the ability to design the NS mashup in a structured service-creation environment. This environment should provide integrity constraints to allow only valid VNF composition and possibly drive (MDA) a continuous integration framework for the testing and validation phase, as well as the creation of the related test NFVI and network sandbox.

### 2.6 TARGET DEVELOPMENT OPERATIONS ENVIRONMENT

DevOps refers to the orchestration of complex interdependent processes associated with software development and IT operations to speed up the production and deployment of products and services. In SDN and virtualized environments, DevOps tools are essential to create a more flexible and responsive infrastructure.⁹

According to QualiSystems research, there are several reasons why DevOps is considered key to SDN and NFV architecture evolution.¹⁰

- **Agility**: SDN and NFV promise greater agility in the service network. However, to leverage these new promising technologies, the internal operations need to evolve. Service development, quality assurance and pre-production testing need to move from waterfall timelines to agile timelines. Considering an exponential increase in deployed service chains over time, the certification process needs to adapt to be able to cope with this ever increasing scope. Network DevOps is the conceptual framework that would allow for required agility in the service development to deployment process.

- **Quality**: Another reason why DevOps is important to the success of SDN and NFV is that rapid innovation needs to be accompanied by quality. Communication services provided by service providers are expected to be higher grade in terms of reliability and quality, especially when targeted to enterprise users. DevOps creates a culture that expects requirements to be collaboratively understood and aimed at by the teams throughout the development-to-deploy process so that it is much less likely to fail upon production deployment, while maintaining an agile innovation pace.

---

⁹ [DevOps], sdxcentral.com.
¹⁰ Why Network DevOps is key to SDN and NFV, Qualisystems.com.
Programmability. SDN and NFV are predicated on the notion of programmable infrastructure. To make the service development to deploy process programmable, one requires a top-down agreement about how that continuous development, integration and delivery process must ultimately work. DevOps offers the framework for that internal agreement and also builds expectations that hand-offs between teams are based on programmable, automated ways of doing business, rather than the obfuscating, silo-reinforcing, manual ways of the past.

In short, integration of DevOps focuses on product delivery, quality testing, feature development and maintenance release to increase reliability and faster development. Table 2 depicts an example use case.

### Table 2. Example of a DevOps use case.\(^\text{11}\)

<table>
<thead>
<tr>
<th>Title</th>
<th>DevOps: Service Development and Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors</strong></td>
<td>SP DevOps team.</td>
</tr>
<tr>
<td></td>
<td>Service execution test resources.</td>
</tr>
<tr>
<td></td>
<td>Support test network resources.</td>
</tr>
<tr>
<td><strong>What is Programmable?</strong></td>
<td>A service development and test environment including the supporting network. This use case covers only testing purposes.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>A brand new service (or application), or service feature is installed in the service provider’s facilities but in a protected “sandbox” environment that is suitable for realistic testing. The sandbox consists of two major areas: the application run time environment made up of compute and storage resources, and a network that connects various run time resources together. The environment is isolated in a way that the production applications and network are not impacted regardless of the new service’s behavior. The setup is automated such that from configuration information, the sandbox environment can be instantiated on demand. Further, the execution of test cases within the environment is also automated via scripts or configuration. This use case focuses on the network portion of the sandbox environment and interface to the service execution resources.</td>
</tr>
<tr>
<td></td>
<td>In addition to testing new services or service features, an existing deployed service may also be tested with different configurations. In other words, testing scenarios may not always require software changes on the service side.</td>
</tr>
</tbody>
</table>

*\(^\text{11}\) Operational Opportunities and Challenges of SDN/NFV Programmable Infrastructure, ATIS-I-0000044, 2013.*
# DevOps: Service Development and Test

发育一个新服务”和“测试它”成为一体化过程，包括硬件、软件、测试案例、配置和部署。软件更新可能在自动化的网络中以程序化的方式进行测试。

## Network Programmability

一个程序化的接口存在于服务执行和支撑网络之间。支撑网络以以下方式实现：

- 动态网络配置，包括拓扑、地址、基本能力（L2或L3操作）、支持网络服务（命名、发现、安全）等。
- 每个网络链路的当前容量要求（如果服务所需）以QoS参数形式表示。这些容量的保留。
- 需要时的任何项目设置的动态变化。这包括拓扑和容量变化（可能在某些预先安排的限制内）。这些变化可以通过SDN安装。
- 可选地，网络对服务有一个响应，如果可用资源不足，则可能需要谈判。
- 网络在服务执行期间向服务提供一些反馈，以检测任何错误或不可预见的事件。

## Transition Challenges

网络必须能够支持服务的动态需求，并以自动化的方式提供支持。

网络必须能够以隔离的方式设置测试网络，不会影响生产网络。

服务提供商可能需要参与软件开发，以便修改或定制由供应商提供的解决方案。

## Advantages

- 实验新服务并测量其性能和对提供商网络的影响。
- 快速地对服务进行增量更改，进行服务改进的测试，而不会影响生产网络。
- 给提供商提供自定义服务的能力。

### OPERATIONAL ISSUES AND RECOMMENDATIONS

虚拟化技术的引入带来了新的运营挑战，因为现有的工作流程/方式预计会经历重大转型。这种转型被视为必要的，以实现下一代基础设施的高效管理。本节探索这些运营挑战并提供支持转型的建议。
### 3.1 OPERATOR’S OPERATIONAL TRANSFORMATION

The operational environment’s ‘softwarization’ has major implications for the operational skill mix and organizational structure. ATIS identified a number of significant changes in the operational environment and concluded that the transformed operational environment was likely to emerge from component standards around APIs rather than system level specifications. Table 3 lists these changes.

**Table 3. Major areas of operational transformation.**

<table>
<thead>
<tr>
<th>Operations Characteristic</th>
<th>Traditional</th>
<th>Transformed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services ordering and provisioning</td>
<td>Manually</td>
<td>Portal or cloud-triggered services, automatically instantiated</td>
</tr>
<tr>
<td>Services lifetime</td>
<td>Static configuration, long</td>
<td>Dynamic configuration/reconfiguration, long and short lifetime services</td>
</tr>
<tr>
<td>Coupling</td>
<td>Network largely independent of applications, except specific Telco services</td>
<td>Network openly programmable by applications, including 3rd party applications</td>
</tr>
<tr>
<td>Transaction volume</td>
<td>Low-moderate</td>
<td>Very high, driven by dynamic cloud apps and virtual network functions (VNFs)</td>
</tr>
<tr>
<td>Network abstraction, exposure</td>
<td>By layer/domain, exposed for managing network</td>
<td>Integrated multi-layer(L3-L0)/multi-domain (access, metro, core) network abstraction, exposed via APIs to applications</td>
</tr>
<tr>
<td>Control and resource management</td>
<td>usually fragmented domain-by-domain and often even vendor-by-vendor</td>
<td>Logically Centralized, multi-layer/multi-domain control and resource management</td>
</tr>
<tr>
<td>Policy</td>
<td>Policy-based access control and QoS</td>
<td>Policy-based end-to-end networking (connectivity, virtualization, multiple flow control points, etc.), in addition to access/QoS)</td>
</tr>
<tr>
<td>Control plane</td>
<td>Distributed</td>
<td>Mix of centralized and distributed control planes</td>
</tr>
<tr>
<td>Controls</td>
<td>Flow-level at selected policy enforcement points</td>
<td>Fine-grained flow-level controls at multiple points across the network</td>
</tr>
<tr>
<td>Entities managed</td>
<td>Hardware / firmware centric devices</td>
<td>Software centric network abstractions</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Operations Characteristic</th>
<th>Traditional</th>
<th>Transformed</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT/Network Integration</td>
<td>Separate IT/Data Center &amp; Network CO with hardware equipment from different vendors</td>
<td>Common technology &amp; technical plant with predominantly virtualized software</td>
</tr>
<tr>
<td>Operations upgrade cycle times</td>
<td>Periodic (typically quarterly) software releases</td>
<td>Continuous software process - “sandbox”; good integration with DevOps</td>
</tr>
<tr>
<td>Deployment</td>
<td>Geographically fixed, single purpose equipment</td>
<td>Highly dynamic &amp; configurable topology &amp; roles</td>
</tr>
<tr>
<td>Coupling</td>
<td>Tight coupled Network Element (NE) instance, generic EMS &amp; NMS/OSS</td>
<td>Separation of physical and logical components</td>
</tr>
<tr>
<td>Service vs. support</td>
<td>Separation of service elements &amp; support systems</td>
<td>Integrated orchestration, automation &amp; virtualization</td>
</tr>
<tr>
<td>Fault impacts</td>
<td>Faults as service failures; fault detection is pseudo real-time</td>
<td>Faults as capacity reduction events; real-time fault detection</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Hardware based; need custom hardware</td>
<td>Software and probe based monitoring; no need for custom hardware</td>
</tr>
<tr>
<td>Reusability</td>
<td>Service specific resource combinations</td>
<td>Profiles, templates &amp; reusable resource combinations</td>
</tr>
<tr>
<td>Processes</td>
<td>Special design and provisioning processes</td>
<td>Configurable catalog/ rule driven delivery frameworks</td>
</tr>
<tr>
<td>Optimization focus</td>
<td>Optimized around provider networks &amp; ops process</td>
<td>Optimized for customer experience</td>
</tr>
</tbody>
</table>

Where functionality implemented in hardware has rigid capacity and deployment constraints, functionality implemented in software is much more flexible and fungible across different services. The traditional siloed operations teams dedicated to support are not sufficiently flexible to operate this environment. The lifecycle processes timeframes, and availability management mechanisms for the infrastructure are quite different from those of VNFs. Rather than service-specific operations teams managing the whole silo of infrastructure, they manage the service in terms of SLAs associated with the underlying service components. Figure 3.1 illustrates these changes.
With the introduction of virtualization, an operator's operational procedures are expected to go through a significant transformation:

- Any unplanned operational and maintenance (OA&M) activity can be reduced in a virtualized environment due to scalability and self-healing functionality available in the infrastructure.

- Cloud elasticity features enable simplification of network monitoring, and support could be reduced from 24x7 to a lesser amount. Also, more activities could be done during the daytime, effectively reducing the maintenance window for all kinds of activities.

- Due to elasticity and high availability design, there might be no need to replace hardware immediately. Also, if an operator has fewer hardware vendors, that will enable more streamlined and efficient processes for repairing hardware.

- Software release upgrades can become more simplified. Orchestration and SDN function increases the efficiency in both integration process of the network elements and acceptance. In cloud environments, more servers can also be upgraded simultaneously due to the added flexibility of reallocation of traffic to different VMs. Work can also be done during the daytime with no downtime.

- There would be significant reduction in data center-related operational activities. Cloud design also eliminates the need for IP infrastructure changes due to minor data center configuration changes.

- Performance monitoring for different applications is envisioned to be highly automated for maintaining the network 24x7.
• With the introduction of NFV along with the SDN component, new services could be launched in a very controlled manner, possibly with no customer impact. Monitoring rollout of new software upgrades should be done through automation of key performance indicators (KPIs) reporting with a feedback mechanism inbuilt for taking necessary actions.

3.2 SCALE AND STATE MANAGEMENT

Scaling is one of the best benefits of the cloud infrastructure. While automatic scaling may be a characteristic desired in target state VNFs, many of the VNFs available initially may not have such capabilities. The time and effort required to redesign some VNFs for automatic scaling is part of the rational for positioning this as a terminal state rather than an initial feature in the evolution of the NFV ecosystem capabilities. Cloud enables dynamically adapting the network according to the performance demand and business policies. The main challenge for scaling is synchronization of topology and configuration changes in an NFV infrastructure. There are three important points about scaling.

1. Scaling in/out and scaling up/down: It is very important to differentiate scaling and the cloud and application level. Scaling in/out mostly works at the application level with adding/removing VMs. Scaling up/down works at the infrastructure level with increasing/decreasing memory, CPUs, etc. Cloud infrastructure has good scaling capabilities, which should be understood and utilized. For example:

- **Scaling up/down:** In this case, the VNFM might detect a lack of resources – such as vRAM, vCPU, vstorage or vpower – for a specific VM and add them dynamically. Scaling up/down could require modification to the existing hardware infrastructure for additional capacity to be introduced in the network. Advances in computing power have vastly increased server resources in each new design. Today, it is possible to replace an aging server with a model having far more processing, memory and I/O capability than previous models, yet occupying the same physical footprint — such as a 1U or 2U rack chassis — and often consuming less energy. This approach is called “scale up” because the physical box can handle more or larger workloads. In a virtualized network, this would possibly be part of a preplanned activity. A scale-up/down server approach might be right for a major virtual server consolidation initiative, where more workloads must reside on the fewer physical servers and vice versa. This would be more targeted towards reusing the same base hardware for a longer period by keeping the building blocks updated.

- **Scaling in/out:** In this case, the application manager detects a lack of resources at the application level. A scale-out approach could be the right answer when a large number of smaller nodes are needed, perhaps for a server cluster where physically redundant hosts are required. For example, if the mobility management entity (MME) doesn’t have enough control plane capacity, then the application manager might decide to add a new VM.

Scaling in and scaling out operations could be fully automated or configured for semi-automated or manual modes. This functionality demonstrates the system’s capability to adjust its resource use to the actual traffic needs based on monitoring of operations for the load on the network elements. When the system becomes overloaded, the system could initiate a scale out operation by triggering to deploy additional scaling units of the network elements. During scale in, when the system loading drops, the system could start a process to remove scaling units from the load-sharing groups so the load balancers do not direct new traffic to those units. The system then
waits until the ongoing calls/sessions are finished. Once the units are empty, it stops the application and the VM, and releases the resources back to the cloud for other applications.

2. **Scaling can be applied for stateful and stateless VNF:** Stateful and stateless VNF should be differentiated. Setting up a new stateless VNF is a relatively simple task because states are not transferred or synchronized. During the stateless scaling, NFV infrastructure should be updated with the address of new network element. The network should be aware of the new network functions and start using them. The simplest solution is to update the load balancer, which will add network elements to the database and start using them. It’s recommended to use any open standard interface for the load balancer configuration. For example, adding a new home subscriber system (HSS) would require updating the DRA load-balancer configuration.

Scaling of a stateful VNF is more challenging. During stateful scaling, NFV infrastructure should be updated with addresses of new network elements. Additionally, state databases should be transferred and/or synchronized to the new element. That would require software modification of existing network functions. For example, adding a new control plane processing unit to the MME, a process that would require updating the MME load balancer and probably transferring states to the new unit. The MME software should be modified to work with the VNFM and provide the ability to add/delete VNF instances dynamically.

3. **Scaling can be applied at the VNF instance and VNF level:** There are cases when new VNFs (e.g., MME, SGW/PGW, PCRF) should be added to the network. This is the most complicated case because it would require reconfiguration of the other VNF. For example, adding a new MME would minimally require updates to the DNS records and eNodeB (eNB) and HSS configurations. Adding new serving/packet gateways (SGW/PGW) would require updates to the DNS records and IP network configuration. It’s extremely important to follow DevOps processes to avoid failure due to configuration.

### 3.3 MANAGEMENT AND MONITORING TOOLS - DIAGNOSING AND ANALYZING PERFORMANCE ISSUES, BOTH NETWORK AND APPLICATION - VPROBES

Traditional telecommunications networks have excellent management and monitoring tools. Measurements and logs are collected from every NF and analyzed using sophisticated software. The telecom industry recognized the importance of KPI analysis and alarms investigation. Every NF has hundreds of counters that are used to observe its health and performance. History proves that KPIs are extremely important to measure network performance, troubleshoot issues and plan network changes.

Achieving the same level of monitoring and measurement in a cloud environment is more challenging and more difficult than in a traditional telecom environment. Recommendation ETSI GS NFV-REL 001 Topic “9.3 Failure Prediction” has an excellent explanation of this increased complexity and the importance of monitoring performance and operational issues. Here is short list of the reasons:

- Ever increasing systems complexity with the addition of virtualization.
- The complexity of integrating the third-party, open-source software and COTS hardware and software components into a system.
- The interoperability across hardware and software components provided by different parties.
Dynamicity (migration, resource elasticity, frequent configurations, reconfigurations, updates, and upgrades) of the system.

Growing connectivity with complex virtual internetworking.

User inexperience in operating the virtual environment.

Software bugs in a new system (e.g., VNF, NFVO).

NFV environment monitoring can be separated into an application level and an NFVI level:

- **Application level**: There are almost no changes with the current practice of monitoring bare-metal applications. The same KPIs that are available in the bare-metal version should be made available for the virtualized application for monitoring purposes.

- **NFV infrastructure KPI**: The main challenge is how to integrate these KPIs with the existing management tools and develop practices to collect and analyze them. A successful approach is to monitor all KPIs that might affect NF performance. Some examples of monitoring cloud infrastructure are:
  
  - Resource utilization of storage, compute (RAM, CPU) and network resources.
  - Monitoring virtual switches and NICs.
  - Monitoring of storage protocols counters.
  - Any other network functions that should be monitored.

It’s important to differentiate active and passive monitoring. Passive is the standard way of collecting and analyzing counters and logs from NFs. In contrast, active monitoring sends special packets that are called probes to measure quality of service (QoS) and availability of a peer. For the monitoring of critical NFVI, processes and application active monitoring should be implemented. Doing so would provide a significant advantage in detecting failures and would decrease restoration time.

Due to the NFV environment’s high complexity, issues might be unpredictable. Sometimes problems appear under specific circumstances that are impossible to reproduce and test in the lab. Unfortunately, standard operational and troubleshooting tools have very limited capabilities for dealing with these types of issues. As a result, virtual probes (vProbes) become an important troubleshooting tool. vProbes collect specific logs based on specific events and are extremely useful for collecting logs when a problem is unpredictable. In some cases, it’s the only one possible way to collect proper logs for the troubleshooting purposes. It’s worth mentioning that vProbes are very different from standard logs collection because vProbes are activated only when an unpredictable problem occurs.

### 3.4 PERFORMANCE MANAGEMENT

Performance management is a highly complex task and is critical for successful NFV deployments and operations. Note that scaling and state management are highly correlated with performance. Performance management can be separated into the following tasks:
• **NFV infrastructure management, including storage, network and CPU resources:** Networks are highly coupled and might be seriously affected by improperly functioning network elements. Resource allocation and resource utilization are extremely important for this type of network. A VM that doesn’t have enough resources due to improper settings might seriously affect the whole network. Any VM that has external connectivity to the other network elements might seriously affect the whole network. For example, a MME load balancer that doesn’t have enough resources might cause flapping HSS links, which might lead to the HSS failure.

Following are guidelines for infrastructure resource management:

- Reservation of the VM resources for 100 percent to ensure the system has enough resources for proper functionality and elasticity.
- Resource sharing should be avoided and considered only as workaround.
- Affinity rules must be considered to allow critical VNFs to run on different blades.
- Resource limits should be considered only as workaround or security measure. Using limits might lead to unpredicted consequences in VM functionality.
- A high availability domain should be available for manually restarting VMs in case of blade failure. But it’s crucial to understand that in most cases, high availability is the method of manual starting VMs in case of hardware failure. State database and signaling would be lost in that case.
- Any method to launch application in sequence (vAPP).

The OpenStack Mistral service provides a sequencing engine that supports the ability to execute complex steps with interdependencies in a specific order. For example, any VM should be able to get all the CPU and memory resources it requires. There should not be any limitation of usage already assigned and provisioned resources during operations. Alarms should be triggered for the resource utilization if a threshold value is achieved. Operational teams would handle this case the same way they did for bare-metal applications.

- In addition to ensuring proper allocation of resources to VNFs, service availability and KPI specifications and monitoring for the infrastructure domain ensure that overall system performance KPIs are met. Examples of measurement KPIs include automated lifecycle management (VM provisioning reliability, DoA ratios, placement compliance, VM failures), virtual machine metrics (VM stall, scheduling latency, clocking errors), networking (packet latency, jitter and loss, throughput), fault detection and recovery as performed by the infrastructure (detection and recovery latencies). These SLAs and KPIs also help perform fault isolation to the VNF applications or infrastructure domains when issues do occur whether at the functional or performance quality dimensions.

• **VNF instance performance management:** VNF instance performance management is very vendor specific because it requires an understanding of internal counters and KPIs. In most cases, only that vendor can ensure proper application scalability. To support scalability, the
vendor-provided performance manager should monitor internal counters and propose scaling. Here are some guidelines:

- The interface from the application to the performance manager should have enough bandwidth.
- There should be integration with the orchestration and alarm systems. Scaling might require assigning different resources that the performance manager doesn’t have access to.

### 3.5 VM DEPLOYMENT AND MIGRATION MANAGEMENT

In the cloud environment with proper configuration, VMs can generally be deployed in any location. There are location transparency exceptions for some applications such as radio access network (RAN) VNFs, which must be situated near the network edge in order to meet protocol latency requirements between the base station, RF antennas and mobile devices. Another example is providing geographical separation for redundant VNFs that provide critical services. In these cases, the orchestrator must incorporate also the VNF deployment specifications.

Every VM has requirements for resource allocation and utilization such as CPU, storage, memory and network. Networking resources are critical because each VM should get its required QoS in terms of bandwidth, jitter and latency. The orchestration system should take to account these requirements and place each VM in a location where it can get enough resources. In theory, it can place VNFI in different data centers if there are enough resources. In practice, the VNFI of one VNF should be located closest to each other.

Live migration is a powerful tool for administrators based on the software-hardware separation paradigm of cloud computing. If a server blade(s) need to be removed from service for maintenance purposes the software applications can be live migrated to another server. Two metrics are important: total migration time (the time taken from the start of the source VM’s migration to the end of the destination VM’s migration) and downtime (the period in which the VM is suspended during the migration).

The downtime impact is particularly important regarding telecom VNFs. When a VM is migrated, it might be unavailable for a few seconds. During this period, all messages to the VM application would be lost. That might be extremely critical for applications that are time sensitive and don’t have any retransmission in case of packet loss. One example is inter-VM traffic inside one network element that is usually UDP based. For these types of applications, it’s highly recommended to move VMs only during maintenance windows. Reducing downtime periods is an important requirement to provide flexibility in maintenance operations and leveraging cloud computing advantages. This is an ongoing area of research.

### 3.6 SECURITY AND THREAT MANAGEMENT: AUTHENTICATION, ENCRYPTION AND TRUST ACROSS VIRTUALIZED NETWORK DOMAINS

In a virtualized environment, security is critical. At a minimum, current threats to, and best practices for securing cloud, network and application environments, all must be taken to the consideration.

Security threats are analyzed in the ETSI recommendation ETSI GS NFV-SEC. Following are some of its important recommendations, along with additional clarification.

- **Loss of availability:**
Flooding an EPC interface/element: Attackers flood an interface/network element, resulting in a denial of service (DoS) condition in the signaling and data planes (e.g., multiple authentication failure on s6a, DNS lookup, malware).

Crashing EPC network elements: Attackers crash a network element by sending malformed packets, creating a buffer overflow.

**Loss of confidentiality:**

- Eavesdropping: Attackers eavesdrop on sensitive data on the control and bearer planes.
- Data Leakage: Unauthorized access to sensitive data on the server, such as the HSS profile.

**Loss of integrity:**

- Traffic modification: In this variant of man-in-the-middle attack, hackers modify information during transit, such as DNS redirection.
- Data modification: An attacker captures admin credentials to facilitate unauthorized access to EPC network elements and install malware.
- Attackers modify data on the network element to change its configurations.

**Loss of control:**

- Control the network: Attackers control the network via protocol or implementation flaws.
- Compromise of network element: Attackers compromise network elements via their OA&M interface.

**Insider attacks:** Insiders modify data on network elements to, for example, make unauthorized configuration changes.

**Theft of service:** Attackers exploit a flaw to use services without being charged. For example, an attacker exploits a flaw in the HSS/PCRF/PCEF to use services without being charged.

The main rule of current security practices is security in depth. Following are some best practices for the network and cloud threat mitigation:

**Initial deployment:** Initial connectivity, certificate distribution and revocation of certificates.

**Hypervisor:** Exploiting vulnerabilities (e.g., buffer overflow), network attack (e.g., DoS), loss of sensitive data and secure boot.

**Networking attack from a VM:** Address resolution protocol (ARP) poisoning, dynamic host configuration protocol (DHCP) attacks, man in the middle, maximum media access control (MAC)
address limitation, DOS (TCP, UDP), confidentiality and integrity of traffic, authentication and Authorization (including initial deployment).

- **Orchestration**: Authentication and authorization of API, misconfiguration.

- **VM**: Stealing a VM, patching, application layer attack, authorization and authentication of configuration.

- **Storage access**: Unauthorized access.

- **Threat mitigation**:
  - Security zone concept: Separating a VM into different trust zones. Concept is very similar to traditional firewall zones.
  - Hypervisor protection: Hardening (e.g., disable unused ports, access control list), monitoring of integrity critical files.
  - VM protection: VM image encryption, sensitive information encryption, intrusion prevention system (IPS) for application traffic.
  - Networking protection: ARP inspection, DHCP snooping, MAC address limitation, control of multicast and unicast traffic, secure communication: (IPSEC, TLS, SSH), PKI solution with certificates, hardware firewall for initial traffic filtering, traffic separation with virtual routing and forwarding (VRF).

Taking into account performance considerations, the security zone concept is extremely important. Each VM should be separated by functionality and level of trust. For example, MMEs that are handling sensitive customer data should be separated from untrusted web servers. That's because the MME is considered a trusted element with strict requirement for performance. Therefore, some security rules, such as IPS or a firewall for each unit, aren't appropriate. In contrast, any web server that's handling external customer requests should be secured as much as possible.

### 3.7 CONSIDERATIONS FOR USING SDN WITHIN NFV

SDN can be used to support and optimize the operation of VNFs, wide area network (WAN) links and ultimately the network services that use them. But there are a number of things operators should consider to successfully integrate SDN into their NFV platforms. These considerations range from straightforward aspects such as deploying virtual or physical switches, to the more challenging ones of dealing with component failure and recovery. There's also the tricky art of conjoining different vendors' solutions into a seamless service, where SDN serves as the glue.

In its most basic form, SDN enables the separation of control and data plane for forwarding packets. The decision on the type of data plane, virtual or physical, rests mostly on the level of performance that is required, although other aspects, such as use of existing transport facilities, might also be a factor. If high packet forwarding throughput is required, then the operator must use a data plane based on hardware/physical network elements. The use of physical elements, however, means that its configuration and layout are fairly static. The SDN forwarding rules used must be suited to a hardware-
centric implementation. Routers in the core network are typically optimized for high performance in all possible ways, for example.

In another case, the traditional cloud has a hardware firewall on the edge of the network that might be stateless or stateful. Whereas, if flexibility is more valued that sheer forwarding speed, a virtualized data plane of virtual switches can be used. Here, software-centric implementations of the data plane allow more dynamic configuration changes and the application of per packet processing such as complex policy rules. Usually virtual switches are used at the access layer to ensure network policy settings and dynamic configuration. Further, connectivity of components within a single VNF or between VNFs lends itself to virtualized data planes that may be tailored to a particular VNF or set of VNFs. One of the advantages of SDN is that a combination of data plane types may be used at the same time. For instance, control plane elements such as MMEs can use virtual switching, while SGWs/PGWs might have problems because of high throughput requirements. In that case, network policy settings should be enforced in the hardware switches at the access layer.

A centralized SDN controller can be a devastating single point of failure. For operators, maintaining high availability of controllers and accomplishing swift failovers are of upmost importance. There are two broad aspects of the problem: responding to hardware/software failures and network partitioning. Some of the hardware failure aspects are very familiar, and operators have deep experience using these redundancy techniques for many years. Traditional methods include the use of HA platforms for SDN controllers, for example, using hot standby components with active state synchronization in between them.

Active and standby controllers have high bandwidth interconnection links. The active controller is monitoring the link status to the agent. It might be simple monitoring of echo request/reply or more complicated with measuring delay, jitter and application KPI. When the active controller detects a link failure or the link quality is worse than that of standby, it will make switchover. This is a highly reliable approach that works in the production IP networks. Key to success is proper tuning of the failover algorithm: The link between controllers must be extremely reliable to avoid an active-active case. Active-standby controllers also help when there is a software failure in the controller as SDN applications can resend requests to the second controller.

Redundant platforms can also be geographically separated to spread out risks. Some SDN protocols support the existence of multiple controllers and rapid switching from a failed controller to an available one. These SDN controllers are geographically separated in different data centers.

Network partitioning helps when the controller itself is up but only some switches have lost connectivity. This provides time-tested techniques such as partitioning the port space across controllers to spread out risks, which prevents the whole switch from going down, or pre-provisioning backup paths to controller(s). Network partitioning approaches rely on the switch to keep operating even in the (limited) absence of a controller. For SDN, the forwarding table in a switch may be kept frozen so existing flows are maintained even if new flows are not admitted. But that would be extremely problematic for switches with external connectivity. They would not be able to respond to any external topology change.

Another technique is to install a backup path ahead of time. That would be easy for the edge routers because they would have a default route to the upper routers. It would be difficult for the intermediate routers. The controller must calculate an end-to-end path to avoid loops.

Finally, hybrid switches might be deployed, that combine SDN control with a traditional control plane (using traditional peer to peer routing protocols), that continue to operate during the time the controller is not available. The switch can run some critical control plane protocols such as ARP, spanning tree
protocol (STP) and link aggregation control protocol (LACP). In case of a controller failure, some basic functionality would work.

Availability zones are a concept that can be used in the data center. The idea is to have at least two totally independent IP networks, which are called availability zones. If one zone is unavailable, the second would be used. For example, Amazon uses this concept.

In addition to these control and data plane approaches, the SDN controller can be viewed as another software component of NFV. As a result, the same approaches of an agile operating environment and continuous integration can be applied to SDN controller instances. This means use of the DevOps approach of regression testing and devoting sufficient time and resources to the ingestion cycle of controllers. For SDN controllers that operate at the VNFC or VNF level, this is natural.

We have seen that SDN can exist in many layers within the NFVI to support and optimize the operation of VNFCs, VNFs, WAN links and ultimately the network services that make use of them. One of the more challenging tasks for the operator is to coordinate these layers and have them interact (in varying degrees) in order to achieve end-to-end service goals. What makes this more difficult is that standards are still forming and there are many implementation possibilities. SDN controllers might reside within the NFVI, potentially as an independent entity. Or perhaps the VIM might expose some control interface to cloud consumers to configure the SDN networks. Additionally, an SDN controller might exist as part of a VNF implementation, in which case, outside the local NFVI, SDN controllers might be operating WAN links to other NFVI sites and perhaps across administrative boundaries.

So far, the industry has not determined the necessary interfaces and information elements that must be exchanged in order for this to be accomplished. The methods for installing a network service that maps to configuring individual VNF SDN networks, as well as the SDN controller in the NFVI, have to be determined. This complex task needs to be automated and derived from the network service description. Without this capability, each SDN layer could be pre-provisioned to operate in isolation with no interaction from other layers. While this is possible, it is closer to traditional networking and does not realize the full potential of an NFV/SDN network. Thus, coordination among various SDN layers is an area that requires more study.

If we turn our attention to a single network service, we see that it may be composed of a set of different VNFs (also known as service functions) strung together in particular order for a particular set of traffic: a service function chain. Here, SDN provides the dynamic networking capabilities to connect these service functions. But the operation of the network service requires that individual VNFs or service functions communicate with the service chain itself, and the SDN controller in order to make adjustments and changes. These can be triggered by service logic within any service function, as well as shifting loads on individual services function instances requiring rebalancing, and re-classification of a flow that requires different treatment. How communication occurs between individual service functions and the SDN controller is currently left as a proprietary deployment exercise. Because network services can vary so greatly, the nature of the information that needs to be exchanged in all cases is not yet clear. So interaction between service functions and the SDN controller for the service chain to meet network service requirements is an area that would benefit from study and standardization.

The SDN-enabled, NFV-based network will support many traditional and new applications. In today's world, networks often have to be customized to support the needs of particular applications. And expressing these needs is relatively crude (e.g., QoS requirements). The NFV-based network has the ability to adapt itself to support application specific requirements and increase efficiency.
What is missing is a way for the network to communicate with applications. There are, however, some emerging approaches to enable this. For example, the application layer traffic optimization (ALTO) protocol allows the network to express topology, link availability and the current routing costs to points in the network.\textsuperscript{13} With ALTO, applications can decide how to access different resources in the network efficiently as the dynamic multi-tenant conditions of the network change. In fact, the OpenDaylight open-source project is implementing an ALTO interface on the SDN controller in order to expose such information to applications.\textsuperscript{14} This provides operators one tool to improve network performance for applications. Over time, it would be beneficial for other technologies to also enable communication in the other direction: application to the network. In this way, the SDN layers within the NFV network and applications can work closely together to improve costs and performance.

### 4 DEPLOYMENT CONSIDERATIONS

Each network operator’s infrastructure typically contains a different mix of equipment from multiple suppliers and delivers multiple services. NFV’s introduction requires consideration of potential migration strategies for various participants in the NFV ecosystem, including VNF providers, NFVI operators and service providers.

A VNF provider that today provides integrated PNFs or network elements has some work to do to transform that functionality into a VNF. Porting or adapting software to execute in the NFVI’s cloud-like execution environment consideration of the effort required and the value achieved in leveraging NFVI capabilities. Various industry approaches are available for evaluating the awareness or maturity of applications in their use of cloud infrastructure capabilities. See, for example, the Application Cloud Maturity Model from ODCA, which defines maturity levels for applications that have been virtualized, loosely coupled, abstracted from the infrastructure and are scale-adaptive.\textsuperscript{15} A previous 4G Americas whitepaper, \textit{Bringing Network Function Virtualization to LTE}, identified auto-scaling as a significant driver for VNF, but this may occur in the later stages of cloud maturity (scale adaptive) rather than being an initial step.

The current state of VNF software may be considered “hardware entangled” because the VNF software has been developed in a fashion that is closely integrated with specific proprietary hardware. The VNF provider also has to consider the range of potential management environments in which the VNF is likely to be deployed. The VNF management arrangements of different service providers may vary and evolve over time.

An NFVI operator that today operates infrastructure based on integrated PNFs has a challenge to develop a deployment plan for the additional infrastructure components required for NFVI. Many of the existing PNFs may be adapted into the NFVI through the use of SDN, but for many NFVI operators, the cloud computing infrastructure components may be relatively new. An NFVI operator starting with a base of cloud infrastructure may need to acquire and deploy additional facilities and PNFs or SDN controllers to build out its NFVI. NFVI operators have to maintain existing services over their infrastructure, and so the deployment and operationalization of additional NFVI nodes needs to be carefully planned to avoid any issues with maintaining existing services to customers.

\textsuperscript{13} \textit{Application-Layer Traffic Optimization (ALTO) Protocol}, RFC7285, IETF, 2014.
\textsuperscript{14} \textsc{ALTO:Main}, Open Daylight.
\textsuperscript{15} \textit{Best Practices: Architecting Cloud-Aware Applications Rev1.0}, Open Data Center Alliance, 2014.
A service provider that today provides end-end services based on operation of some of its own PNFs, as well as interconnection agreements, may find changes in the management of services in a virtualized environment. Where there is a one-to-one correspondence between VNFs and PNFs, in terms of network functionality, the VNFs are virtualized and do not provide management perspectives of the underlying NFVI. Where the underlying NFVI is provided by an independent NFVI operator (whether a separate commercial entity or an internal organizational structure), the dependencies of the end-end service on that NFVI need to be managed through the SLAs with the NFVI operator.

The NFV objectives for service providers include increased service velocity, not just in deployment, but also in service design. NFV’s overlay architecture enables rapid service deployment across multiple NFVI operators where appropriate commercial agreements are in place. Service design velocity is expected to lead to rapid innovation in services, requiring commensurate rapid innovation in the management systems that support those services. In general, this results in increased sophistication of management approaches. There are a variety of such approaches that have been developed in the IT industry and are being discussed in the communications industry under rubrics such as policy and autonomies. Autonomics is generally concerned with properties such as:

- Self-configuration: Automatic configuration of components.
- Self-healing: Automatic discovery and correction of faults.
- Self-optimization: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements.
- Self-protection: Proactive identification and protection from arbitrary attacks.

When IBM introduced notions of autonomic computing, it also considered an autonomic computing deployment model,\(^{16}\) where applications would evolve through five levels, starting from basic, through managed, predictive, adaptive and finally to autonomic.

- The basic level represents the starting point where a significant number of VNFs are today, with each VNFC managed independently by systems administrators who set it up, monitor it and enhance it as needed.
- At the managed level, systems management technologies are used to collect information from disparate VNFCs into one, consolidated view, reducing the time it takes for the administrator to collect and synthesize information.
- At the predictive level, new technologies are introduced that provide correlation among several of the VNFCs. The VNF itself can begin to recognize patterns, predict the optimal configuration and provide advice on what course of action the administrator should take. As these technologies improve, service providers will become more comfortable with the advice and predictive power of the VNFs.

• The adaptive level is reached when VNFs can not only provide advice on actions, but also automatically take the right actions based on the information that is available to them in the context of what is happening in the system.

• The full autonomic level would be attained when the VNF operation is by users interacting with the VNF to monitor the business processes, and/or alter the objectives.

Figure 4.1 illustrates the variety of VNF evolution options that VNF providers face as they consider the evolution of their products in both cloud awareness/maturity and management sophistication dimensions. There is a tension between adopting a design style that integrates many functions into a large VNF, but which may have limited applicability, versus a microservices design style, which focuses on components with a wider overall applicability.
implementation approach to deployments of VNFs is required. The scale of existing infrastructures precludes wholesale replacement from both operational and economic perspectives. Incremental deployments of VNFs will occur in a variety of different scenarios. Some Greenfield VNF deployments are possible, but in many cases the VNFs will need to be deployed considering existing deployments. Where VNF are introduced into an existing service, they need to interwork with existing deployments. This is most likely through the use of existing standard interfaces. Even with standard interfaces, existing network services may require some reconfiguration, such as to redirect traffic to the VNFs.

Virtualization and cloud technology brings significant benefits for the cloud elements. They might potentially impact the architecture of network elements: single-session database, single load balancing for all network elements and so on. But for the first implementation, it’s recommended to use virtualized application software with the same functionality. VNFs must comply with the following requirements:

- Full transparency to the end user in terms of availability, accessibility and retainability. User experience should not be affected by network functions virtualization.

- Full interoperability with legacy network elements. Virtualized network elements should support all current interfaces (e.g., S1, S11, Gx, Gy).

The following additional topics should be considered:

- **Protection of IP network and network elements against VM misbehavior**: For example, a virtual MME can affect an HSS on the application level, while the vPGW might affect the PCRF. Misconfiguration of any VM might cause all possible misbehavior: broadcast or multicast storm, flapping interfaces, delay and high jitter. That might seriously affect peers and the IP network in general.

- **Graceful shutdown of virtualized elements**: For example, in the case of the vMME, it’s highly recommended to add it to the existing pool. That would give possibility to gracefully shutdown MME and statefully move subscribers to the other MME.

- **Resource allocation and utilization**: In the traditional cloud environment, the concept of overprovisioning is very common. That can be achieved with advanced tools such as resource reservation, shares and limits. But LTE networks are very highly coupled, and a VM that doesn’t have enough resources might have a significant negative impact on the network.

Based on these recommendations, the following migration strategy can be considered:

1. Identify network functions that can be virtualized. Pure control plane network functions can be virtualized faster than high-performance data plane network elements. Typically, it can be MMEs, route reflectors and IMS elements. In contrast, virtualization of data plane elements is more difficult because of requirements to high performance of packet processing.

2. Performed lab testing of the VNFs. The main task during this testing is verifying the virtualized environment but not the functionality of applications. For example, testing of a vMME would focus on understanding how virtualization impacts the network and how to operate NFVI. The vMME would support the same functionality and the same call signaling flow but in the virtualized environment. That would help to test software with absolutely the same functions and features but in the virtualized environment.
Current recommendations are to limit usage of cloud benefits like automation. Automation requires very careful analysis and development of all test cases. Due the NFV environment’s high complexity, it would be very difficult to accomplish at this stage.

3. Develop a strategy and methods to monitor the whole NFVI. This is a very challenging task because of this environment’s high complexity.

4. Develop a strategy to protect the network against misbehavior of VNF instances and gracefully shut down in those situations.

### 4.1.1 GREENFIELD DEPLOYMENTS

Greenfield opportunities are those where the VNF deployments provide new services that were not previously deployed. Thus, there is no legacy infrastructure to be replaced. Figure 4.2 illustrates a VNF being deployed in a Greenfield situation. In this case, the Greenfield is an enterprise point of presence (PoP) where the service provider did not have existing equipment. A new NFVI node is deployed in that PoP and has the infrastructure software installed on it. The VNFs are then deployed and operated by the service provider.

The infrastructure software could be packaged by an independent provider from an open-source community effort such as OPNFV. The VNFs and NFVI node hardware could be obtained from independent commercial entities. A variety of VNF functions may be desirable in such a deployment, including firewalls, routers and NATs.

![Diagram of a VNF Greenfield deployment](image)

**Figure 4.2. Example of a VNF Greenfield deployment.**

### 4.1.2 OFFLOADING OF EXISTING PNFS

Some existing PNFs (e.g., routers) may provide a variety of network services, such as DNS, time and route reflectors. In many cases, these services provide additional load on the processors embedded in
the PNF. By offloading some of these functions to VNFs executing on a NFVI node, the PNF’s processing capacity may be freed up, enabling the useful service life of the PNF to be extended. Figure 4.3 illustrates this general case. Note that systems external to the existing PNF may need to be reconfigured to redirect traffic along a new path to where the network service is now being provided.

![Figure 4.3. Expanding a PNF’s capacity and extending its service life.](image)

### 4.1.3 VIRTUALIZATION OF THE MANAGEMENT NETWORK

4G Americas previously identified EMS, OSS and BSS as being low-risk/high-value network functions to virtualize.\(^{17}\) When selecting network functions to virtualize, service providers should give serious consideration to the virtualization of the management network. Most of the network management functions are relatively easy to virtualize because they are typically transactional in nature. The management network becomes more dynamic as new VNFs are deployed. The deployment loads and monitoring loads during diagnostics can require significantly more bandwidth on the management network than traditional network management operations. The ability to scale up or schedule the management bandwidth adds significant operational flexibility. In addition, the virtualization and disaggregation of the management functions permits service providers to implement the operational process benefits identified previously. These operational process improvements are central to the OPEX savings and service velocity objectives for NFV.

---

\(^{17}\) *Bringing Network Function Virtualization to LTE*, 4G Americas, November 2014.
Figure 4.4 illustrates a traditional management network on the left (a). On the right (b), the management network functions have been virtualized, and the virtualized management network is used to manage the existing PNFs as before.

![Figure 4.4. Comparison of management scenarios.](image)

### 4.1.4 CUTOVER OF INDIVIDUAL PNFs

Some existing PNFs (e.g., routers) may be reaching the end of their service life and need to be replaced. In this case, the services provided by the PNF need to be replaced by a VNF. Figure 4.5 illustrates this general case. Note that systems external to the existing PNF may need to be reconfigured to redirect traffic along a new data path to where the network service is now being provided.

![Figure 4.5. Replacing a PNF with a VNF.](image)
4.2 OPEN-SOURCE INITIATIVES

Deploying open source in critical infrastructure is an important topic to discuss in the context of NFV. While open-source software source code is available for modification or enhancement by anyone, making the code usable for critical infrastructure components would need some certification, which possibly can be done.

OpenStack is open-source software, which means that anyone who chooses to, can access the source code, make any changes or modifications they need and freely share these changes with the community at large. It also means that OpenStack has the benefit of thousands of developers all over the world working in tandem to develop the strongest, most robust and most secure product that they can. The key architectural attributes of OpenStack are as follows:

- Loosely coupled versus tightly coupled.
- API driven.
- Multiple hypervisors (KVM, Xen, vSphere, Hyper-V).
- Multiple disk formats (QCOW, RAW, VMDK.).
- Integrated networking.

Along with the above OpenStack attributes, improved and customized versions of OpenStack are also available today.

4.3 DEVOps – AUTOMATION IN DEPLOYMENT AND OPERATIONALIZATION

SDN/NFV promotes software development of network functionality that would give clear benefits with implementing complex logic for comprehensive business cases. These cases might include precise utilization and assigning network resources against some business rules and policy.

SDN is the process of software development of network applications. Traditional network hardware such as routers and switches have very complex software for the packet processing. It might include software for routing and switching protocols, ACL, NAT, packet inspection and so on. Networking companies use special processes to develop software to ensure each network element’s proper functionality.

To successfully utilize SDN/NFV, service providers should adapt a software development approach to develop, test, deploy and monitor software. One of the best methods currently used is DevOps, which focuses on gradual, incremental development of software. DevOps encourages excellent communication and collaboration inside teams, and it makes software development processes more reliable and faster. The following principles can be used:

- **Develop and deploy with incremental and gradual changes**: Code will be developed with the small interactions and gradual changes. Code might be tested with a predefined amount of test cases that are focused on the specific functionality and features. The development environment should have the same deployment process as in the production lab. Here are few important guidelines for the deployment process:
“Commit verify” and “commit rollback.” The configuration can be loaded to the network element and verified on errors with commit verify. If configuration is successful, then load it but roll back in case of no acknowledgement in the specified amount of time.

Separated logins for write and read operations. An account with write privilege should be used only in when it’s really needed.

Deploy software for the automatic tracking of configuration and comparing it with the planned configuration.

- **Validate in the lab:** Functional testing can be performed in the lab, which replicates the production environment, right down to the same network topology, software and firmware version, configuration and all other settings. The validation process should be automated as much as possible.

Lab validation would be important for development of operation documents for operational and deployment procedures. These documents are often called MOP. Verification of software in the lab would highly decrease risk of a configuration error. It also would help detect more software bugs and avoid crashing software in the production network. This is extremely important because if software has a bug then standard redundancy solutions such as HA and geo-redundancy would not help. They might crash all at the same time in the specific scenario.

- **Constantly monitor and verify.** Monitoring and verification is extremely important in the complex, multivendor environment. It’s recommended to use independent sources of information. For example, compare KPIs from network elements with network analyzers, which capture traffic from interfaces. That would help to find any discrepancies and improperly functioning counters.

In a virtual environment, it’s essential to monitor different levels of operational environment and correlate information from different sources. Examples include the hypervisor, utilization of virtual resources (e.g., vCPU, vMemory), application counters and counters from network analyzers.

- **Apply changes.** Modern networks are very dynamic and can require different resources based on policy rules. Orchestration systems should have a scope of predefined cases for applying changes in the network. Policy rules might include specific scenarios on resource allocation based on network utilization, date and time, location and other cases. It’s critical that most of the cases for applying changes in the network should be predefined and well tested in the lab environment.

**CONCLUSIONS**

NFV and SDN promise enormous opportunities for operators to provide innovative services at a much faster pace than traditional cycles. However, to realize the full potential of this new agile architecture, operators have some work to do with regard to their internal operations and deployment functions. Adopting this operational paradigm shift, coupled with well-planned introduction and deployment of NFV in the existing networks, would ensure non-intrusive gradual transformation to a much more efficient programmable network with capabilities to offer a multitude of innovative services rapidly to the market.
ACKNOWLEDGEMENTS

The mission of 4G Americas is to advocate for and foster the advancement and full capabilities of the 3GPP family of mobile broadband technologies, including LTE-Advanced and beyond to 5G, throughout the ecosystem's networks, services, applications and wirelessly connected devices in the Americas. 4G Americas' Board of Governors members include Alcatel-Lucent, América Móvil, AT&T, Cable & Wireless,
Cisco, CommScope, Entel, Ericsson, HP, Intel, Mitel, Nokia, Qualcomm, Sprint, T-Mobile US, Inc. and Telefónica.

4G Americas would like to recognize the significant project leadership and important contributions of Asim Butt of Ericsson and Ravi Danda of Mitel, as well as representatives from member companies on 4G Americas’ Board of Governors who participated in the development of this white paper.

The contents of this document reflect the research, analysis, and conclusions of 4G Americas and may not necessarily represent the comprehensive opinions and individual viewpoints of each particular 4G Americas member company.

4G Americas provides this document and the information contained herein to you for informational purposes only, for use at your sole risk. 4G Americas assumes no responsibility for errors or omissions in this document. This document is subject to revision or removal at any time without notice.

No representations or warranties (whether expressed or implied) are made by 4G Americas and 4G Americas is not liable for and hereby disclaims any direct, indirect, punitive, special, incidental, consequential, or exemplary damages arising out of or in connection with the use of this document and any information contained in this document.

© Copyright 2015 4G Americas