Network Service Virtualization Requirements

Version 1.0

A white paper from the
ONUG NSV Working Group

October, 2014

NSV WORKING GROUP
2014
Open Networking
USER GROUP

Definition of Open Networking

Open networking is a suite of interoperable software and/or hardware that delivers choice and design options to IT business leaders, service and cloud providers. At its core, open networking is the separation or decoupling of specialized network hardware and software - all in an effort to give IT architects options in the way in which they choose to design, provision, and manage their networks. These technologies must be based on industry standards. The standards can be de-facto as adopted by a large consortium of the vendor community, open in the sense that they are community based, or defined as standards by the prevailing standards bodies. Open networking hopes to deliver on two promises:

1) Decoupling of network hardware and software which mitigates vendor lock-in and shifts network architecture structure options to users
2) Significant reduction of the total cost of ownership model, especially operational expense

NSV WORKING GROUP
2014
Open Networking
USER GROUP

Executive Summary

This document defines a common set of functional solution requirements for one of the open networking use cases, network service virtualization, identified by the ONUG community. The content of this document is intended as general guidelines for IT enterprise end users to compare vendor solutions and develop formal RFI specifications, and for IT vendors to develop and align product requirements. Defining a common set of solution requirements aligns with ONUG's goal to drive the IT vendor community to deliver open and interoperable networking solutions in order to provide IT end users maximum choice and flexibility.

The expectation is that this document provides a common architecture framework, covering the majority of enterprise deployment requirements for network service virtualization solutions. The assumption is being made that this set of requirements will be completed by enterprise-specific requirements to meet specific deployment needs.

Finally, the expectation is that the scope of requirements defined in this document will evolve. Hence, the versioning of this document.

Why Network Service Virtualization?

One of the leading complaints from ONUG IT leaders is the cost and complexity of managing a huge number of Layer 4-7 network appliances from different vendors with different management tools. Layer 4-7 comprises a diverse group of network elements, including but not limited to::

- Server load balancers and application delivery controllers (e.g., F5 and Citrix),
- WAN optimization (e.g., Riverbed),
- Firewalls (e.g., Palo Alto, Checkpoint, Cisco, Juniper),
- SSL/IPSec VPNs and Intrusion Detection and Prevention (IDS/IPS).

Each of these appliances is optimized to scale vertically to provide a specific service, but they are typically provided by different suppliers, each with a unique way of provisioning, managing and operating. The result is largely a complex stack of high-priced, proprietary appliances (boxes), which are ill adapted to the rapid pace of business application changes and innovation in the data center.

The Network Service Virtualization or NSV Working Group seeks to leverage the flexibility and low costs of commodity servers to establish a scale out pooling of virtual and physical appliances, which can be put to use servicing applications. As each Layer 4-7 function is virtualized in software, it provides the following benefits:

- Lower CAPEX costs (approximately 30 percent less);
- Rapid service provisioning and ability to deploy Layer 4-7 services that follow specific virtualized sets of applications;
- Reduced risk through service distribution;
- Eased management and reduced operational costs through ability to be centrally managed by generalized IT operational teams;
- Consistent policies across different Layer 4-7 services and across data center, campus and WAN networks;
- Programmatic control and ability to offer network functions as a service to developers.

Open Networking User Group (ONUG)

ONUG is one of the largest industry user groups in the networking and storage sectors. Its board is made up exclusively of IT business leaders, with representation from Fidelity Investments, FedEx, Bank of America, UBS, Cigna, Pfizer, JPMorgan Chase, Citigroup, Credit Suisse, Gap, Inc., and Symantec. The ONUG mission is to guide and accelerate the adoption of open networking solutions that meet user requirements as defined through use cases, proof of concepts, hackathons, and deployment examples to ensure open networking promises are kept.

The ONUG community is led by IT business leaders and aims to drive industry dialogue to set the technology direction and agenda with vendors. To that end, ONUG hosts two major conferences per year where use cases are defined and members vote to establish a prioritized list of early adopter, open networking projects that communicate propensity to buy and budget development. The vendor community stages proof of concepts based upon ONUG Use Cases, while standards and open source organizations prioritize their initiatives and investments based upon them. ONUG also hosts user summits and smaller, regional user-focused Fireside Chat Meet-Ups through the year.

ONUG defines six architectural areas that will open the networking industry and deliver choice and design options. To enable an open networking ecosystem, a common multivendor approach is necessary for the following six architecture components:

1) Device discovery, provisioning, and asset registration for physical and virtual devices
2) Automated "no hands on keyboards" configuration and change management tools that align DevOps and NetOps
3) A common controller and control protocol for both physical and virtual devices
4) A baseline policy manager that communicates to the common controller for enforcement
5) A mechanism for sharing (communicating or consuming) network state and a unified network state database that collects, at a minimum, MAC and IP address forwarding tables automatically
6) Integrated monitoring of overlays and underlays

In addition, NSV is a critical component required to enable IT business leaders to provide on-demand or self-service IT delivery to business unit managers.

A majority of Layer 4-7 networking suppliers currently offer software versions of their popular appliances. However, to enable NSV, Layer 4-7 software must be optimized to run in a virtual environment. NSV elements should be designed to provide the following in a virtual environment:

- Performance at scale (without the need for proprietary hardware appliances) with capability scale-out/in automatically based on predefined trigger policies;
- Orchestration and management support that integrates well with other open networking elements, e.g., SDN controllers, virtual overlay networks and other Layer 4-7 (virtual) applications;
- A new licensing model that enables multiple instances of a network service to be attached to dedicated applications;
- NSV instances per application to avoid application traffic bouncing between network services within the data center;
- Fully open and well-documented APIs that enable programmatic control and ease of automation and integration;
- Ability to accept policy from higher layer orchestrators plus policy managers for translation into device specific configuration;
- Central point of management and control for all instances of a given service to simplify management via a single pane of glass.

Vendors, in many cases, provide only a unique API in which to provision their network service appliance. Thus the implementer is tasked with the time consuming and daunting challenge to manage policies, changes, syncs, audits, etc. Further, the implementer then faces a challenging learning curve with each vendor's automation strategy. In addition to providing an API, vendors should provide on-boarding services and global management capabilities, which could help integrate automated ecosystems and help IT departments improve time-to-market via integration with OpenStack, Chef, Puppet, Netconf and others.

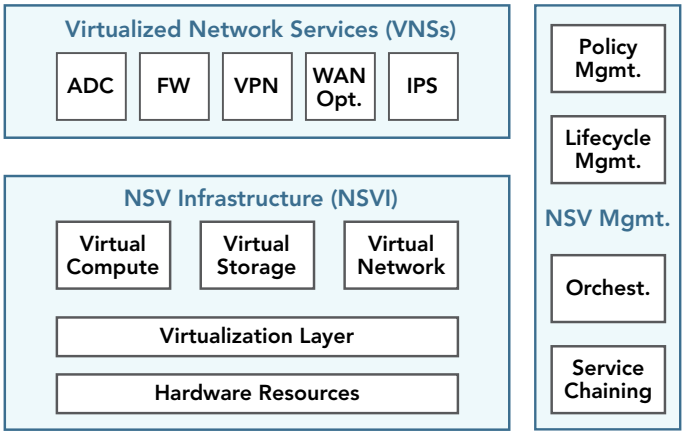
Due to the many high-priced, hardware-based Layer 4-7 network appliances paired between branch offices and data centers, ONUG members view NSV as a high priority for adoption. Key challenges for implementation include the revision of existing software licensing policies, lack of standards, and need for service orchestration between multiple types and "flavors" of Layer 4-7 appliances. The ONUG community requires that vendors subscribe to a common policy model so that, once the policy is defined, IT departments can leverage various vendor network services and be assured that policy is enforced consistently across all services.

Network Service Virtualization Architecture Framework

For the sake of consistency and alignment between user groups, we will try to reuse as much as possible the Architecture Framework developed by ETSI NVF ISG in ETSI GS NFV 002 (October, 2013). We will only modify it where necessary to fit enterprise use cases and requirements.

Network Service Virtualization aims to transform the way enterprises architect networks by evolving standard IT virtualization technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage, which could be located in data centers, network nodes and in the end user premises. It

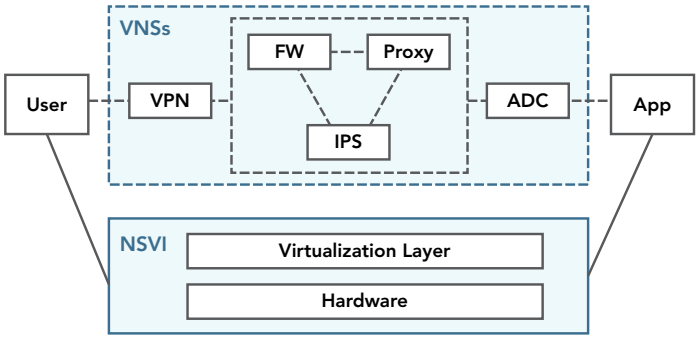
involves the implementation of network services in software that can run on a range of industry standard server hardware, and that can be moved to, or instantiated in, various locations in the network as required, without the need for installation of new equipment.



The figure above illustrates the high-level NSV framework and the three main components of NSV:

- Virtualized Network Service (VNS) is the software implementation of a network service which is capable of running over the NSVI (e.g., application delivery controller, firewall, WAN optimization controller, etc.).
- NSV Infrastructure is the platform comprised of hardware and software components where VNSs are deployed, managed and executed.
- NSV Management includes orchestration and lifecycle management of physical and virtual resources needed by NSVI, lifecycle management of VNSs, policy management for VNS-specific policies, and a common data store for VNS forwarding graphs. NSV Management focuses on all NSV-specific management tasks necessary in the NSV framework.

Network services can be chained together into forwarding graphs, defining the path between two end points, e.g., between two applications, or between an application and an end user. A forwarding graph can have VNS nodes connected by logical links that can be unidirectional, bidirectional, multicast or broadcast. All forwarding graphs are stored into a common data store that is part of the NSV Management component. An example of such a forwarding graph is shown below. The decoupling of hardware and software in NSV is accomplished through a virtualization layer abstracting the hardware resources of the NSV Infrastructure. VNS nodes run on top of the virtualization layer and are chained together into a forwarding graph. The



interfaces between VNSs and between VNSs and the NSVI must be based on open industry standards.

High-Level Requirements

For the sake of consistency and alignment between user groups, we will try to reuse as much as possible the Virtualization Requirements developed by ETSI NVF ISG in ETSI GS NFV 004 (October, 2013). We will only modify it where necessary to fit enterprise use cases. Requirements labeled as [Adopt.] are adopted without modification, ones labeled as [Mod.] are modified, and [New] are introduced here for the first time. In addition, for each requirement, a priority is assigned using the following guidelines:

Priority: < High, Medium or Low >

High: Functionality that must be supported at day one and is critical for baseline deployment.

Medium: Functionality that must be supported, but is not mandatory for initial baseline deployment.

Low: Desired functionality, which should be supported, but can be phased in as part of longer term solution evolution.

General Requirements

[Mod.] [Gen.1] The NSV framework shall be able to permit enterprises to fully virtualize the network services they deploy and operate. **Priority: < High >**

[Mod.] [Gen.2] Any impact on the performance or operation of non-virtualized services shall be manageable, predictable and within the acceptable limits. **Priority: < High >**

[Mod.] [Gen.3] There shall be manageable impact on the legacy management systems of the network services that have not been virtualized. **Priority: < High >**

[New] [Gen.4] The NSV framework shall be able to support composition of VNS forwarding graphs consisting of VNS nodes from multiple vendors. **Priority: < High >**

General Requirements

The NSV target is to achieve portability across multiple vendors, hypervisors and hardware, while continuing to meet SLA requirements.

[Adopt.] [Port.1] The NSV framework shall be able to provide the capability to load, execute and move VNSs across different but standard multivendor environments. **Priority: < High >**

[Adopt.] [Port.2] The NSV framework shall support an interface to decouple VNS associated software instances from the underlying infrastructure. **Priority: < High >**

[Adopt.] [Port.3] The NSV framework shall be able to provide the capability to optimize the location, reservation and allocation of the required resources of the VNSs. **Priority: < High >**

Portability

The NSV target is to achieve portability across multiple vendors, hypervisors and hardware, while continuing to meet SLA requirements.

[Adopt.] [Port.1] The NSV framework shall be able to provide the capability to load, execute and move VNSs across different but standard multivendor environments. **Priority: < High >**

[Adopt.] [Port.2] The NSV framework shall support an interface to decouple VNS associated software instances from the underlying infrastructure. **Priority: < High >**

[Adopt.] [Port.3] The NSV framework shall be able to provide the capability to optimize the location, reservation and allocation of the required resources of the VNSs. **Priority: < High >**

Performance

[Adopt.] [Perf.1] The NSV framework shall be able to instantiate and configure any given VNS over the underlying infrastructure so that the behavior of the resulting VNS instance in terms of performance is conforming to the requirements expressed in the VNS information model provided by the VNS vendor for such a type of infrastructure. **Priority: < High >**

[Adopt.] [Perf.2] The NSV framework shall be able to describe the underlying infrastructure requirements of a VNS so that it can be sized for a given performance target while the corresponding resources are allocated and isolated or shared on the infrastructure accordingly. **Priority: < High >**

[Adopt.] [Perf.3] For any running VNS instance, the NSV framework shall be able to collect performance related information regarding the usage of compute, storage and networking resources by the VNS instance.
Priority: < Medium >

[Adopt.] [Perf.4] The NSV framework shall be able to collect performance related information concerning the resource usage at the infrastructure level (e.g., hypervisor, NIC, virtual switch).
Priority: < Medium >

Elasticity

The following requirements apply when a VNS or its components can be parallelized to realize elasticity:

[Adopt.] [Elas.1] The VNS vendor shall describe in an information model for each component capable of parallel operation the minimum and maximum range of such instances it can support as well as additional information such as the required compute, packet throughput, storage, memory and cache requirements for each component. **Priority: < Medium >**

[Adopt.] [Elas.2] The NSV framework shall be able to provide the necessary mechanisms to allow virtualized network services to be scaled with SLA requirements. Different mechanisms shall be supported, e.g., on-demand scaling, automatic scaling, etc.

On-demand scaling of a VNS instance may be initiated by the VNS instance itself, by another authorized entity (e.g., Orchestrator) or by an authorized user (e.g., NOC administrator).

Automatic scaling of a VNS instance can be initiated based on some trigger, e.g., when pre-defined criteria included in the information model describing a VNS are met.
Priority: < Medium >

[Adopt.] [Elas.3] The scaling request or automatic decision may be granted or denied depending on, for example, network-wide views, rules, policies, resource constraints or external inputs.
Priority: < Medium >

[Adopt.] [Elas.4] The VNS user, through standard information model, shall be capable of requesting for each component capable of scaling specific minimum and maximum limits within range specified by the VNS vendor to fulfill individual SLA, regulatory or licensing constraints. **Priority: < Medium >**

[Mod.][Elas.5] The NSV framework shall provide the capability to move some or all VNS components from one compute resource onto a different compute resource while meeting the service continuity requirements for the VNS components. The movement of VNS components can be within a single administrative domain or between administrative domains within a data center or between data centers. **Priority: < High >**

Resiliency

Adopt.] [Res.1] The NSV framework shall be able to provide the necessary mechanisms to allow network service to be recreated after a failure. The relevant resiliency characteristics of a VNS or a set of VNSs shall be made available to the entities handling the network service. Different mechanisms shall be supported, e.g., on-demand recreation, automatic recreation.

On-demand recreation of a VNS instance may be initiated by the VNS instance itself, by another authorized entity (e.g., Orchestrator) or by an authorized user (e.g., NOC administrator).

Automatic recreation of a VNS instance can be initiated based on some trigger, e.g., when pre-defined criteria included in the information model describing a VNS are met.
Priority: < High >

[Adopt.] [Res.2] The NSV framework shall be able to provide a means to classify (sets of) VNSs that have similar reliability/availability requirements into resiliency categories.
Priority: < Low >

[Adopt.] [Res.3] The NSV framework shall be able to support standard-based replication of state data (synchronous and asynchronous) and preservation of data integrity with the necessary performance to fulfill the SLAs. **Priority: < High >**

[Adopt.] [Res.4] The NSV framework (including the orchestration and other functions necessary for service continuity) shall facilitate resiliency schemes in both the control plane and the data plane in order to secure service availability and continuity. Orchestration functionalities and other functions necessary for managing service continuity shall not become a single point of failure. **Priority: < High >**

[Adopt.] [Res.5] The SLA shall specify the "metrics" to define the value and variability of "stability." **Priority: < Medium >**

[Adopt.] [Res.6] In order to enable network stability, the NSV framework shall support mechanisms to measure the following metrics and ensure that they are met per SLA:

- Maximum non-intentional packet loss rate (e.g., packets lost due to oversubscription of the service network interconnects, not due to policies or filters).
- Maximum rate of non-intentional drops of stable calls or sessions (depending on the service).
- Maximum latency and delay variation on a per-flow basis.
- Maximum time to detect and recover from faults aligned with the service continuity requirements (zero impact or some measurable impact).
- Maximum failure rate of transactions that are valid and not made invalid by other transactions.

Additional metrics necessary for defining network stability can be addressed at a later point. **Priority: < Medium >**

Security

[Adopt.] [Sec.1] The NSV framework shall implement appropriate security countermeasures to address:

- Security vulnerabilities introduced by the virtualization layer;
- Protection of data stored on shared storage resources or transmitted via shared network resources;
- Protection of new interfaces exposed by the interconnectivity among NSV end-to-end architectural components, e.g., hardware resources, VNSs, management systems;
- Resource isolation of distinct VNS sets executing over the NVSI to ensure security and separation between these VNS sets;
- Secure management of VNS sets by other third-party entities (e.g., vendors, outsource contractors, etc.).
  Priority: < High >

[Adopt.] [Sec.2] The NSV framework shall be able to provide mechanisms for the network operator to control and verify the configuration of the VNSs and the elements that virtualize the hardware resource. **Priority: < High >**

[Adopt.] [Sec.3] Management and orchestration functionalities shall be able to use standard security mechanisms wherever applicable for authentication, authorization, encryption and validation. **Priority: < High >**

[Adopt.] [Sec.4] NSV Infrastructure shall be able to use standard security mechanisms wherever applicable for authentication, authorization, encryption and validation. **Priority: < High >**

[Adopt.] [Sec.5] The NSV framework shall be able to provide role-based information access and rights management. Each actor based on its associated role definition will have access to a subset of the VNS instances and a subset of the VNS instance management functions (e.g., creation, modification, activation). A special role will be the administrator role that is able to manage roles and rights. **Priority: < High >**

[Adopt.] [Sec.6] Access to NSV functions via NSV exposed APIs at all layers shall be protected using standard security mechanisms appropriate for that layer wherever applicable for authentication, authorization, data encryption, data confidentiality and data integrity. **Priority: < High >**

[Adopt.] [Sec.7] The management and orchestration functionality shall provide at least two levels of privileges to API users (e.g., root privilege and user privilege; in this case, the root privilege is a higher level of privilege that the user one). Each privilege gives access to a range of differentiated APIs. **Priority: < Medium >**

[Mod.][Sec.8] The NSV exposed APIs should be divided into multiple subsets of APIs so that users with different levels of privilege will only be able to use certain subsets of API functionality based on the users' levels of privilege. A special case is that the management and orchestration functionality allow using all APIs for the highest privilege only. **Priority: < Low >**

[Mod.][Sec.9] The management and orchestration functionality shall be able to authorize users' privilege for using APIs based on administrator-defined criteria. **Priority: < Medium >**

Service Continuity

The NSV framework shall provide the following capabilities:

[Adopt.] [Cont.1] The SLA shall describe the level of service continuity required (e.g., seamless, non-seamless according to the definitions) and required attributes.

There are two cases of the impact on service continuity in the events of intervening exceptions or anomalies: zero impact (seamless service continuity) and measurable impact (non-seamless service continuity).

In the case of seamless service continuity in response of some anomaly (e.g., detected failure, commanded movement and migration), there will be a means specified such that no observable state loss, no observable transmit queue packet loss and no observable transmit queue storage loss occurs, and any impact on latency and delay variations will be within the SLA specification for the service.

In the case of non-seamless service continuity, some level of measurable service impact can be perceived by the end user. When there is measurable service continuity impact on the function, then any impact will be described in terms of the SLA specification to include at least a maximum value of outage duration, packet loss, latency and delay variation. **Priority: < High >**

[Adopt.] [Cont.2] In the event of an anomaly that causes hardware failure or resource shortage or outage, the NSV framework shall be able to provide mechanisms such that the functionality of impacted VNS instances shall be restored within the service continuity SLA requirements for the impacted VNS instances. **Priority: < High >**

[Adopt.] [Cont.3] In the event that a VNS instance or a subset needs to be migrated, the NSV framework shall be able to consider the impact on the service continuity during the VNS instance migration process and such impact shall be measurable and within the limits described in the SLA. **Priority: < High >**

[Adopt.] [Cont.4] When a VNS instance subset is migrated, the communication between the migrated VNS instance and other entities (e.g., VNS instance subset or physical network element) shall be maintained regardless of its location and awareness of migration. **Priority: < High >**

Service Assurance

[Adopt.] [SeA.1] The NSV framework shall provide mechanisms for time-stamping by hardware (e.g., NICs, switches, packet brokers that sit beneath virtualization infrastructure). The minimum support from hardware shall be to:

- Copy packets or frames.
- Accurately time-stamp the copies, using a clock synchronized to a source of appropriate precision.
- Forward the time-stamped copies to a configured destination.

Once the precise time-stamps have been added in hardware, all other instrumentation and diagnosis functions can then proceed as virtualized functions without strict time constraints, e.g., filtering headers, removing payloads, local analysis, forwarding for remote analysis, logging, storage, etc. **Priority: < Medium >**

[Adopt.] [SeA.2] It should be possible to interrogate whether particular network interface hardware provides hardware time-stamping facilities. **Priority: < Low >**

[Adopt.] [SeA.3] A (set of) VNS instance(s) and/or management system shall be able to detect the failure of such VNS instance(s) and/or network reachability to that (set of) VNS instance(s) and take action in a way that meets the fault detection and remediation time objective of that VNS resiliency category. **Priority: < High >**

[Adopt.] [SeA.4] A VNS shall be able to publish means by which other entities (e.g., another VNS, the orchestration functionality and/or management system) can determine whether the VNS is operating properly. **Priority: < High >**

Management and Operations

[Adopt.] [MaO.1] The NSV framework shall incorporate mechanisms for automation of operational and management functions, e.g., creation, scaling and healing of VNS instance based on pre-defined criteria described in the VNS information model, network capacity adaptation to load, software upgrades and new features/nodes introduction , service configuration and relocation, and intervention on detected failures. The above are examples and not an exhaustive list. **Priority: < High >**

[Adopt.] [MaO.2] The NFV framework shall be able to provide a management and orchestration functionality that shall be responsible for the VNS and VNS instances lifecycle management: instantiation, allocation and relocation of resource, scaling and termination. **Priority: < High >**

[New] [MaO.3] The NFV framework shall be able to provide a management and orchestration functionality that shall be responsible for the logical functions provided by the VNSs, including but not limited to the operational, provisioning and management aspects of the logical functions of the VNS. **Priority: < High >**

[Adopt.] [MaO.4] As part of the VNS lifecycle management, monitoring and collection of information related to usage, the management and orchestration functionality that shall be able to interact with other operation systems managing the VNSs and/or the NSV infrastructure comprised of compute and storage machines, network software and hardware, and configurations and software on these devices. **Priority: < Medium >**

[Mod.] [MaO.5] The management and orchestration functionality shall be able to use standard information models that describe how to manage the VNS lifecycle. Information models provide a structure of operational attributes of VNSs as well as characteristics of the network service in terms of capacity, performance, resiliency, constraints and security, such as (not an exhaustive list):

- Deployment attributes and environment of a VNS, e.g., VM image, required computational and storage resources and network reachability.
- Operational attributes of a VNS, e.g., VNS topology as the links between the different network services, operations (e.g. initiation/tear-down), functional scripts, operational policies.
- Migration attributes of a VNS, e.g., limitations for maximum acceptable propagation delay, scaling and resiliency methods defined by the SLA.

  Priority: < High >

[Adopt.][MaO.6] The management and orchestration functionality shall be able to manage the lifecycle of VNSs and VNS instances using the information models in combination with run-time information accompanying scheduled or on-demand requests regarding VNS instances and run-time policies/constraints. **Priority: < High >**

[New][MaO.7] The management and orchestration functionality shall be able to manage the logical functions of VNSs and VNS instances using the information models in combination with run-time information accompanying scheduled or on-demand requests regarding VNS instances and run-time policies/constraints. **Priority: < High >**

[Adopt.] [MaO.8] The management and orchestration functionality shall be able to manage the NSV infrastructure in coordination with other applicable management systems and orchestrate the allocation of resources needed by the VNS instances. **Priority: < High >**

[Adopt.] [MaO.9] The management and orchestration functionality shall be able to maintain the integrity of each VNS instance with respect to its allocation NSV infrastructure resources. **Priority: < High >**

[Adopt.] [MaO.10] The management and orchestration functionality shall be able to monitor and collect NSV infrastructure resource usage and map such usage against the corresponding particular VNS instances. **Priority: < High >**

[Adopt.] [MaO.11] The management and orchestration functionality shall be able to monitor resources used on a per-VNS basis and shall be made aware of receiving event that reflect NSV infrastructure faults, correlate such event with other VNS related information, and act accordingly on the NSV infrastructure that supports the VNS. **Priority: < High >**

[Adopt.] [MaO.12] The management and orchestration functionality shall support standard APIs for all applicable functions (e.g., VNS instantiation, VNS instance allocation/release of NSV infrastructure resources, VNS instances scaling, VNS instances termination and policy management) that it provides to other authorized entities (e.g., CMS, VNS instances, 3rd parties, etc.). **Priority: < High >**

[Mod.][MaO.13] The management and orchestration functionality shall be able to manage policies and constraints (e.g., regarding placement of VMs) including policies for the logical functions of the VNS. **Priority: < High >**

[Adopt.] [MaO.14] The management and orchestration functionality shall enforce policies and constraints when allocating and/or resolving conflicts regarding NSV infrastructure resources for VNS instances. **Priority: < High >**

[Adopt.] [MaO.15] The NSV framework shall be able to manage the assignment of NSVI resources to a VNS in a way that resources (compute hardware, storage, network) can be shared between VNSs. **Priority: < High >**

Energy Efficiency

It is expected that the NSV framework can exploit the benefits of virtualization technologies to significantly reduce the energy consumption of large-scale network infrastructures.

[Mod.][EE.1] The NSV frameworks shall support the capability to place only VNS subset that can be moved or placed in a sleep state on a particular resource (compute, storage) so that resource can be placed into a power conserving state.

Workload consolidation can be achieved by scaling facilities so that traffic load is concentrated on a smaller number of servers outside business hours so that all the other servers can be switched off or put into energy saving mode. **Priority: < Low >**

[Mod.][EE.2] The NSV frameworks shall be able to provide mechanisms to enable an authorized entity to control and optimize energy consumption on demand by, for example, scaling scheduling and placing VNS instances on specific resources, including hardware and/or hypervisors, placing unused resources in energy saving mode and managing power states as needed. Energy efficiency mechanisms should consider maintaining service continuity requirements and network stability requirements. **Priority: < Low >**

[Adopt.][EE.3] The NSV frameworks shall provide an information model that includes attributes defining the timeframe required for a compute resource, hypervisor and/or VNS (e.g., VM) to return to a normal operating mode after leaving a specific power-saving mode. This is necessary to determine when to power on resources and software sufficiently in advance of the time when such assets would be needed to meet expected future workloads. **Priority: < Medium >**

Coexistence with and Transition from Existing Networks

[Mod.][Mig.1] The NSV framework shall co-exist with legacy network equipment and shall be able to work in a hybrid network composed of classical physical network services and VNSs. **Priority: < High >**

[Mod.] [Mig.2] The NSV framework shall support a transition path from today's physical network services to a more open standards based virtual network services. **Priority: < High >**

[Adopt.] [Mig.3] The NSV framework in conjunction with legacy management system shall support the same service capability and acceptable performance impact within service SLA when transitioning from physical network services to VNSs. **Priority: < High >**

[Adopt.] [Mig.4] The NSV framework shall be able to interwork with legacy management systems with minimal impact on existing nodes and interfaces. **Priority: < High >**

[Adopt.] [Mig.5] During the transition from physical to virtual, the NSV framework shall be able to ensure security of VNS instances from various security threats without disrupting or negatively impacting existent physical network services and associated network elements and interfaces. **Priority: < High >**

Recommendations

For completeness, the specific areas in network service virtualization solutions where there is a need for open well-defined and vendor agreed on technology standards can be summarized as follows:

VNS-to-NSVI interface: open south-bound interface for configuration of virtual network services and communication of hardware independent lifecycle, performance and portability requirements of the VNS.

VNS-to-NSVM interface: open north-bound interface for management and orchestration systems to communicate policy information to the VNSs and instantiate forwarding paths composing a forwarding graph and receive topology and state information from the VNSs.

NSVM-to-NSVI interface: open south-bound interface for management and orchestration systems to perform orchestration and lifecycle management of physical and virtual resources needed by NSVI and orchestration and lifecycle management of VNSs.

References

ETSI GS NFV 001 Network Function Virtualisation (NFV): Use Cases, October 2013

ETSI GS NFV 002 Network Function Virtualisation (NFV): Architectural Framework, October 2013

ETSI GS NFV 003 Network Function Virtualisation (NFV): Terminology for Main Concepts in NFV, October 2013

ETSI GS NFV 004 Network Function Virtualisation (NFV): Virtualisation Requirements, October 2013

Network Functions Virtualisation – Introductory White Paper, October 2012

Network Functions Virtualisation – Update White Paper, October 2013

Open Data Center Alliance Master Usage Model: Software-Defined Networking Rev. 2.0, 2014

ATIS-I-0000044 Operational Opportunities and Challenges of SDN/NFV Programmable Infrastructure, October 2013

ONUG Open Networking Challenges and Opportunities: A White Paper from the ONUG Board of Directors, July 2014

ONUG NSV Working Group

Vesko Pehlivanov, Chairman — Credit Suisse

Margaret Chiosi — at&t

Travis Griffin — FedEx

Gary Hemminger — vArmour

Kevin Irwin — Fidelity Investments

Bireshwar Karmakar — bmc

Apurva Mehta — Versa Networks

Jenny Oshima — NEC

Michael Quirk — United

Kishora Vekaria — SecureKeys Consulting

Sean Wang — UBC The University of British Columbia

Steven Wright — at&t

James Younan — UBS

NSV WORKING GROUP 2014
Open Networking USER GROUP