ETSI GS NFV-REL 002 V1.1.1 (2015-09)

GROUP SPECIFICATION

Network Functions Virtualisation (NFV); Reliability; Report on Scalable Architectures for Reliability Management

Disclaimer

This document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGS/NFV-REL002

Keywords

architecture, NFV, reliability

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document describes a study of how today's Cloud/Data Centre techniques can be adapted to achieve scalability, efficiency, and reliability in NFV environments. These techniques are designed for managing shared processing state with low-latency and high-availability requirements. They are shown to be application-independent that can be applied generally, rather than have each VNF use its own idiosyncratic method for meeting these goals. Although an individual VNF could manage its own scale and replication, the techniques described here require a single coherent manager, such as an orchestrator, to manage the scale and capacity of many disparate VNFs. Today's IT/Cloud Data Centres exhibit very high availability levels by limiting the amount of unique state in a single element and creating a virtual network function from a number of small replicated components whose functional capacity can be scaled in and out by adjusting the running number of components. Reliability and availability for these type of VNFs is provided by a number of small replicated components. When an individual component fails, little state is lost and the overall VNF experiences minimal change in functional capacity. Capacity failures can be recovered by instantiating additional components. The present document considers a variety of use cases, involving differing levels of shared state and different reliability requirements; each case is explored for application-independent ways to manage state, react to failures, and respond to increased load. The intent of the present document is to demonstrate the feasibility of these techniques for achieving high availability for VNFs and provide guidance on Best Practices for scale out system architectures for the management of reliability. As such, the architectures described in the present document are strictly illustrative in nature.

Accordingly, the scope of the present document is stated as follows:

- Provide an overview of how such architectures are currently deployed in Cloud/Data Centres.
- Describe various categories of state and how scaling state can be managed.
- Describe scale-out techniques for instantiating new VNFs in a single location where failures have occurred or unexpected traffic surges have been experienced. Scale-out may be done over multiple servers within a location or in a server in the same rack or cluster within any given location. Scaling out over servers in multiple locations can be investigated in follow-up studies.
- Develop guidelines for monitoring state such that suitable requirements for controlling elements (e.g. orchestrator) can be formalized in follow-up studies.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] R. Strom and S. Yemini: "Optimistic Recovery in Distributed Systems", ACM Transactions on Computer Systems, 3(3):204-226, August 1985.

[i.2] Sangjin Han, Keon Jang, Dongsu Han and Sylvia Ratnasamy: "A Software NIC to Augment Hardware", in Submission to 25th ACM Symposium on Operating Systems Principles (2015).

[i.3] E.N. Elnozahy, Lorenzo Alvisi, Yi-Min Wang, David Johnson: "A Survey of Rollback-Recovery Protocols in Message-Passing Systems", ACM Computing Surveys, Vol. 34, Issue 3, September 2002, pages 375-408.

[i.4] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson and A. Warfield: "Remus: High Availability via Asynchronous Virtual Machine Replication". In Proceedings USENIX NSDI, 2008.

[i.5] Kemari Project.

NOTE: Available at http://www.osrg.net/kemari/.

[i.6] J. Sherry, P. Gao, S. Basu, A. Panda, A. Krishnamurthy, C. Macciocco, M. Manesh, J. Martins, S. Ratnasamy, L. Rizzo and S. Shenker: "Rollback Recovery for Middleboxes", Proceedings of the ACM, SIGCOMM, 2015.

[i.7] ETSI NFV Reliability Working Group Work Item DGS/NFV-REL004 (V0.0.5), June 2015: "Report on active Monitoring and Failure Detection in NFV Environments".

[i.8] OPNFV Wiki: "Project: Fault Management (Doctor)".

NOTE: Available at https://wiki.opnfv.org/doctor.

[i.9] E. Kohler et al.: "Click Modular Router", ACM Transactions on Computer Systems, August 2000.

[i.10] "Riverbed Completes Acquisition of Mazu Networks".

NOTE: Available at: http://www.riverbed.com/about/news-articles/press-releases/riverbed-completes-acquisition-of-mazu-networks.html.

[i.11] Digital Corpora: "2009-M57-Patents packet trace".

[i.12] S. Rajagopalan et al.: "Pico Replication: A High Availability Framework for Middleboxes", Proceedings of ACM SoCC, 2013.

[i.13] Remus PV domU Requirements.

NOTE: Available at http://wiki.xen.org/wiki/Remus_PV_domU_requirements.

[i.14] B. Cully et al.: "Remus: High Availability via Asynchronous Virtual Machine Replication", Proceedings USENIX NSDI, 2008.

[i.15] Lee D. and Brownlee N.: "Passive Measurement of One-way and Two-way Flow Lifetimes", ACM SIGCOMM Computer Communications Review 37, 3 (November 2007).

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

affinity: for the purposes of the present document, property whereby a flow is always directed to the VNF instance that maintains the state needed to process that flow

checkpoint: snapshot consisting of all state belonging to a VNF; required to make an identical "copy" of the running VNF on another system

NOTE: One way to generate a checkpoint is by using memory snapshotting built in to the hypervisor.

core: independent processing unit within a CPU which executes program instructions

correct recovery: A system recovers correctly if its internal state after a failure is consistent with the observable behaviour of the system before the failure.

NOTE: See [i.1] for further details.

flow: sequence of packets that share the same 5-tuple: source port and IP address, destination port and IP address, and protocol

non-determinism: A program is non-deterministic if two executions of the same code over the same inputs may generate different outputs.

NOTE: Programs which when given the same input are always guaranteed to produce the same output are called deterministic.

stable storage: memory, SSD, or disk storage whose failure conditions are independent of the failure condition of the VNF; stable storage should provide the guarantee that even if the VNF fails, the stable storage will remain available

state: contents of all memory required to execute the VNF, e.g. counters, timers, tables, protocol state machines

thread: concurrent unit of execution, e.g. p-threads or process.h threads

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CDF	Cumulative Distribution Function
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DPDK	Data Plane Development Kit
DPI	Deep Packet Inspection
FTMB	Fault Tolerant MiddleBox
Gbps	Giga bits per second
HA	High Availability
IDS	Intrusion Detection System
IP	Internet Protocol
Kpps	Kilo packets per second
Mpps	Mega packets per second
NAT	Network Address Translation
NFV	Network Function Virtualisation
NFVI	Network Function Virtualisation Infrastructure
NIC	Network Interface Controller
NUMA	Non Uniform Memory Access
QoS	Quality of Service
TCP	Transmission Control Protocol
VF	Virtual Function
VM	Virtual Machine

VNF	Virtualised Network Function
VPN	Virtual Private Network
WAN	Wide Area Network

4 Scalable Architecture and NFV

4.1 Introduction

Traditional reliability management in telecommunications networks typically depends on a variety of redundancy schemes. For example, spare resources may be designated in some form of standby mode; these resources are activated in the event of network failures such that service outages are minimized. Alternately, over-provisioning of resources may also be considered (active-active mode) such that if one resource fails, the remaining resources can still process traffic loads.

The advent of Network Functions Virtualisation (NFV) ushered in an environment where the focus of telecommunications network operations shifted from specialized and sophisticated hardware with potentially proprietary software functions residing on them towards commoditized and commercially available servers and standardized software that can be loaded up on them on an as needed basis. In such an environment, Service Providers can enable dynamic loading of Virtual Network Functions (VNF) to readily available servers as and when needed - this is referred to as "scaling out" (see note). Traffic loads can vary with bursts and spikes of traffic due to external events; alternately network resource failures may reduce the available resources to process existing load adequately. The management of high availability then becomes equivalent to managing dynamic traffic loads on the network by scaling out VNFs where needed and when necessary. This is the current method of managing high availability in Cloud/Data Centres. The goal of the present document is to describe how such scalable architecture methods can be adapted for use in NFV-based Service Provider networks in order to achieve high availability for telecommunications services.

NOTE: It is also possible to reduce the number of existing VNFs if specific traffic types have lower than expected loads; this process is known as "scaling in".

The use of scalable architecture involves the following:

- Distributed functionality with sufficient hardware (servers and storage) resources deployed in multiple locations in a Service Provider's region.
- Duplicated functionality within locations and in multiple locations such that failure in one location does not impact processing of services.
- Load balancing such that any given network location does not experience heavier loads than others.
- Managing network scale and monitoring network state such that the ability of available resources to process current loads is constantly determined. In the event of failures, additional VNFs can be dynamically "scaled-out" to appropriate locations/servers such that high availability is maintained.

The following assumptions are stated for the development of the present document:

- Required hardware (servers and storage) is pre-provisioned in sufficient quantities in all Service Provider locations such that scaling-out new VNFs is always possible at any given location when necessary.
- Required hardware is distributed strategically over multiple locations throughout the Service Provider's network.
- The relationship between the type of service and the corresponding VNFs necessary to process the service type is expected to be known.

4.2 Overview of Current Adoption in Cloud Data Centres

Typical services offered by Cloud providers include web based services and cloud computing. Scalable architectures for managing availability in response to load demands have been successfully implemented by Cloud Service providers. A high level overview of the techniques for achieving high availability is as follows:

- Sizing Functional Components: Cloud providers now craft smaller components in terms of functionality and then deploy very large numbers of such components in Data Centres. Sizing such components is thus important - how much functional software can be loaded onto commercial hardware products. Each hardware resource therefore handles fewer functions than the traditional hardware resources. If one or more such components fail, the impact on service delivery is not expected to be very significant.

- Distributed Functionality: Data Centres are located in multiple regions by the Cloud Service Provider. Failure in one Data Centre does not impact the performance of other Centres. Functionality is duplicated simply by deploying large numbers of functional components. The distributed nature of Cloud Data Centres thus permits storage of critical information (service and customer information) within one location and in multiple locations insulated from each other. Failure in one location thus permits the relevant information to be brought online through alternate Centres.

- Load Balancing: Incoming load can be processed though a load balancer which distributes load by some designated mechanism such that no Data Centre system experiences overload conditions. Given multiple locations and multiple storage of critical information, load balancing provides a method to ensure availability of resources even under failure conditions.

- Dynamic Scalability: Again, given the small size of functional components, it is fairly straightforward to scale-out (or scale-in) necessary resources in the event of failure or bursty load conditions.

- Managing Scale and State: Methods for keeping track of the state of a Cloud Service provider's resources is critical. These methods enable the provider to determine whether currently deployed resources are sufficient to ensure high availability or not. If additional resources are deemed necessary then they can be brought online dynamically.

4.3 Applicability to NFV

The main motivating factor for Service Providers for adopting NFV is the promise of converting highly specialized communication centre locations (e.g. Central Offices, Points of Presence) in today's networks into flexible Cloud-based Data Centres built with commercial hardware products that:

1) Continue the current function of communication centres, namely; connect residential and business customers to their networks.

2) Expand into new business opportunities by opening up their network infrastructures to third party services. An example of such a service is hosting services currently offered by Cloud Data Centres.

Embracing an NFV-based design for communication centres allows Service Providers to enable such flexibility. This also incentivizes Service Providers to explore Cloud/Data Centre methodologies for providing high availability to their customers.

Today's communication centres provide a wide range of network functions such as:

- Virtual Private Network (VPN) support

- Firewalls

- IPTV/Multicast

- Dynamic Host Configuration Protocol (DHCP)

- Quality of Service (QoS)

- Network Address Translation (NAT)

- Wide Area Network (WAN) Support

- Deep Packet Inspection (DPI)
- Content Caching
- Traffic Scrubbing for Distributed Denial of Service (DDoS) Prevention

These functions are well suited for an NFV environment. They can be supplemented with additional functions for delivery of Cloud services such as hosting. In principle, all such functions can be managed for high availability via Cloud-based Scalable Architecture techniques.

Traditional reliability management in telecommunications networks typically depends on a variety of redundancy schemes whereby spare resources are designated in some form of active-active mode or active-standby mode such that incoming traffic continues to be properly processed. The goal is to minimize service outages.

With the advent of NFV, alternate methods of reliability management can be considered due to the following:

- Commercial Hardware - Hardware resources are no longer expected to be specialized. Rather than have sophisticated and possibly proprietary hardware, NFV is expected to usher in an era of easily available and commoditized Commercial Off-the-Shelf products.
- Standardized Virtual Network Functions (VNF) - Software resources that form the heart of any network's operations are expected to become readily available from multiple sources. They are also expected to be deployed in multiple commercial hardware with relative ease.

In such an environment, it can be convenient to "scale-out" network resources - rapidly instantiate large numbers of readily available and standardized VNFs onto pre-configured commercial hardware/servers. This results in large numbers of server/VNF combinations each performing a relatively small set of network functions. This scenario is expected to handle varying traffic loads:

- Normal Loads - Typically expected traffic loads based on time-of-day and day-of-week.
- Traffic Bursts - Such situations can arise due to outside events or from network failures. Outside events (e.g. natural disasters, local events of extreme interest) can create large bursts of traffic above and beyond average values. Network failures reduce the available resources needed to process service traffic loads thereby creating higher load volumes for remaining resources.

Scaling out resources with NFV can be managed dynamically such that all types of network loads can be satisfactorily processed. This type of dynamic scale-out process in response to traffic load demands results in high availability of network resources for service delivery.

The present document provides an overview of some of these techniques to ensure high availability of these functions under conditions of network failures as well as unexpected surges in telecommunications traffic.

5 Scaling State

5.1 Context

This clause presents a high level overview of the context underlying the solution methods that are presented in clause 6. The focus here is on managing high availability of VNF services within a single location; this location may be a cluster deployed within a Service Provider's Central Office, a regional Data Centre, or even a set of racks in a general-purpose cloud. A Service Provider's network will span multiple such locations. The assumption is that there is a network-wide control architecture that is responsible for determining what subset of traffic is processed by which VNFs in each location. For example, the controlling mechanism might determine that Data Centre D1 will provide firewall, WAN optimization and Intrusion Detection services for traffic from customers C1, . . . , Ck. A discussion of this network-wide control architecture is beyond the scope of the present document.

It is critical to note that the focus of the present document is only on meeting the dictates of the network-wide controlling mechanism within a single location, in the face of failure and traffic fluctuations. Some high level descriptions of the architecture utilized for this study are as follows:

- Infrastructure View: It is understood that multiple architectures are possible for the solution infrastructure. The clause 6 solution techniques are based on a high level architecture that comprises a set of general-purpose servers interconnected with commodity switches within each location. The techniques for managing scale are presented in the context of a single rack-scale deployment (i.e. with servers interconnected by a single switch); the same techniques can be applied in multi-rack deployments as well. As shown in figure 1, a subset of the switch ports are "external" facing, while the remaining ports interconnect commodity servers on which VNF services are run. This architecture provides flexibility to balance computing resources and switching capacity based on operator needs. A traffic flow enters and exits this system on the external ports: an incoming flow may be directly switched between the input and output ports using only the hardware switch, or it may be "steered" through one or more VNFs running on one or more servers.

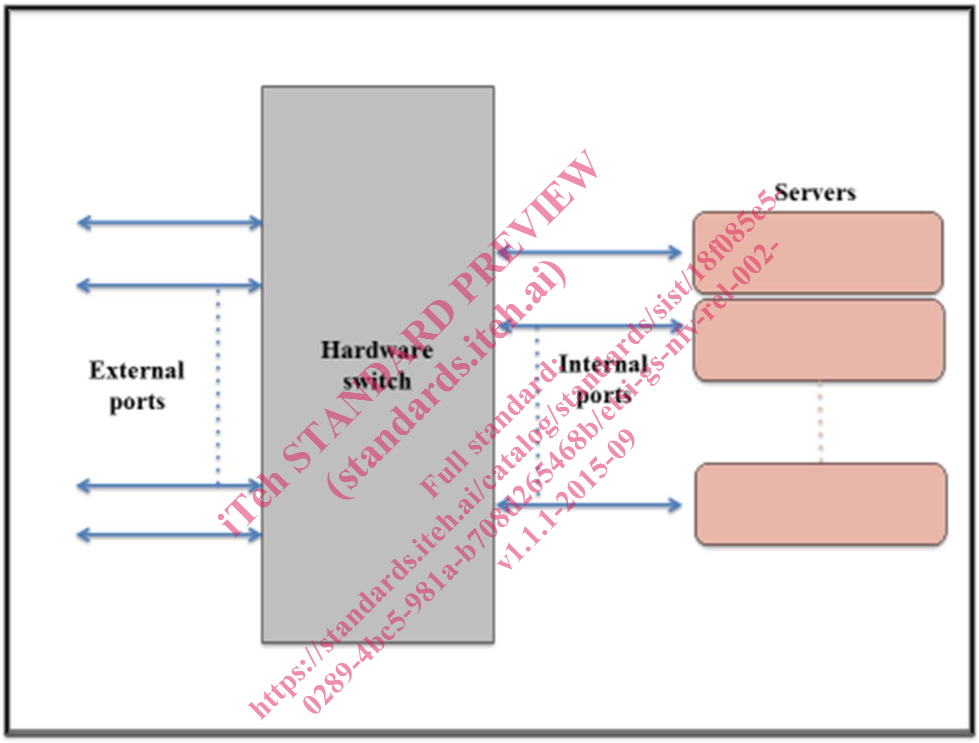

Figure 1: Hardware Infrastructure

- System View: The overall system architecture (within a single location) is illustrated in figure 2. This architecture comprises three components:
    - Logically centralized controlling mechanism (such as an orchestrator) that maintains a system-wide view.
    - Virtual Network Functions (VNFs) implemented as software applications
    - Software switching layer that underlies the VNFs - VNFs implement specific traffic processing services - e.g. firewalling, Intrusion Detection System (IDS), WAN optimization - while the software switching layer is responsible for correctly "steering" traffic between VNFs.