

Network working group
INTERNET-DRAFT
Intended Status: Informational
Expires: April 27, 2015

P.A. Aranda
Telefonica
D. King
Lancaster University
M. Fukushima
KDDI R&D Labs
October 27, 2014

Virtualization of Content Distribution Network Use Case
draft-aranda-vnfpool-cdn-use-case-00

Abstract

This use case document provides requirements for moving Content Distribution Networks (CDNs) from physical servers to a virtualized environment. This new kind of CDN, known as virtualized CDN (vCDN), allows for new constructs that simplify the CDN architecture. The main elements of the CDN are analyzed with regards to the degree of elasticity demanded from them in terms of computation, storage and network resources.

This use case document provides resiliency requirements for virtualization of the Content Distribution Network, known as virtualized CDN (vCDN).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 27, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
1.1	Defining Resilience	3
1.1.1	Resiliency for stateless services	3
1.1.2	Resiliency for stateful services	3
1.2	Terminology	4
2	vCDN Use Case	4
2.1	Terms and definitions	4
2.2	Content Distribution Network Components	5
2.2.1	Cache Node	6
2.2.2	CDN Controller	6
2.2.3	CDN Load Balancer	6
2.2.4	Surrogate Server	6
2.2.5	Content Proxy	6
2.2.6	Content Peering Gateway	7
2.3	vCDN Resiliency Requirements	7
2.4	Service Degradation	7
2.5	Applicability of Virtual Network Function Pool (VNFPool)	7
2.6	Coexistence of Virtualized and Non-virtualized Network Functions	8
3	Security Considerations	9
4	IANA Considerations	9
5	References	9
5.1	Normative References	9
5.2	Informative References	9
6	Acknowledgement	10
	Authors' Addresses	10

1 Introduction

Delivery of content, especially of video, is one of the major challenges of all operator networks due to massive growing amount of traffic.

Growth of video traffic is driven by the shift from broadcast media to unicast delivery via IP. This is also complementary to the growth of today's video on demand traffic.

Additional on-demand content services to Internet end-users, have similar quality constraints as video, high bandwidth and low latency, and stored as close to users as possible.

A Content Delivery Network (CDN) represents a group of geographically dispersed servers deployed to facilitate the distribution of information generated by content providers in a timely and efficient manner.

As physical functions, including CDN components, are migrated to virtual platforms, Virtual Network Functions (VNF), a critical aspect will be ensuring the VNF is resilient. Maintaining that resilience, especially, when virtual resources are dynamically migrated and managed will require co-ordination between VNFs.

This document discusses the key network resilience objectives for the virtualized CDN. It outlines the challenges and risks for the appropriate resilience requirements to negate or ensure minimal impact of CDN-based services.

1.1 Defining Resilience

In the context of this I-D resiliency will ensure the ability to provide and maintain an acceptable level of service or function to the user, in the event of faults and challenges to normal operation.

1.1.1 Resiliency for stateless services

In the case of services that do not require maintaining state information, it is sufficient to move the VNF offering that service to a new Virtual Machine (VM) or hardware entity.

1.1.2 Resiliency for stateful services

When a VNF is moved e.g., for failure mitigation, maintenance or workload consolidation, the offered service and its performance can be maintained, which is regarded as "service continuity" by those entities which are using it.

2 vCDN Use Case

2.1 Terms and definitions

CDN Provider: The service provider who operates a CDN and offers a service of content delivery, typically used by a Content Service Provider or another CDN Provider.

Content: Any form of digital data. One important form of Content with additional constraints on distribution and delivery is continuous media (i.e., where there is a timing relationship between source and sink).

Content Delivery Network (CDN): The network infrastructure in which the network elements cooperate at Layers 4 through 7 for more effective delivery of Content to Users.

Network Service Provider (NSP): Provides network-based connectivity and services to Users.

Over-the-top (OTT): A service, e.g., content delivery using a CDN, operated by a different operator than the Operator to which the users of that service are attached.

Service Continuity: ensure that if a service needs to be relocated to another site due to an anomaly event (e.g. CPU overload, hardware failure or security threat). The configuration of the VNF (e.g. IP address) is preserved; thus (ideally) there is no impact on the end user or node.

Users: The end user that interacts with a Content service. Such communication is not restricted to HTTP and may be via a variety of protocols. Examples of Users include: browsers, Set Top Boxes (STBs), dedicated content applications (e.g., media players), etc.

2.2 Content Distribution Network Components

A number of functional components exist for deployment and operation of Content Distribution Networks (CDN), these include:

- o Content Cache Node to deploy content as close to each user as possible;
- o Content Controller to route the users request for content to the closest available content store or content engine;
- o Content Load Balancing to distribute user requests across one or multiple servers;
- o Surrogate Servers for mirrored web content servers;
- o Content Proxies;

- o Content DNS Servers;
- o GeoIP Information Servers;
- o Content Peering Gateways.

In many CDN deployments, CDN nodes are dedicated physical appliances or software with specific requirements on standard but dedicated hardware. Often physical appliances and servers for different purposes are deployed side-by-side. This comes with a number of disadvantages:

- o The capacity of the devices needs to be designed for peak hours (typically on weekend evenings). During weekdays and business hours, the dedicated hardware appliances and CDN servers are mainly unused.
- o It is not possible to react on unforeseen capacity needs e.g. in case of a live-event as hardware resources need to be deployed in advance.
- o The average peak utilization and resilience of CDN nodes for dedicated purposes or from different partners is lower as it could be if the hardware resources would be shared between virtual appliances on the same infrastructure.
- o Dedicated physical devices and servers from several parties drive the complexity of the operator network and increase the operational expenses.
- o Content delivery is a very volatile market driven by new content formats, protocols, device types, content protection requirements etc. Dedicated designed hardware hinders the necessary flexibility to react on these changes.
- o Content Delivery may imply some Value Added Services, e.g., for Security concerns or for optimizing Performances. It may be valuable for the Network Operator to rely on Outsourcing of a Partner's solution rather than having to operate its own solution.

Therefore, it is important for CDNs to offer service continuity to users during partial failures of key CDN elements, including: load balancers, proxies and surrogate servers.

2.2.1 Cache Node

Operators to deploy their proprietary cache nodes into the ISP network. CDN cache nodes are dedicated physical appliances or software with specific requirements on standard but dedicated hardware.

2.2.2 CDN Controller

A CDN controller objective is to select a cache node (or a pool of cache nodes) for answering to the end-user request, and then redirect the end-user to the selected Cache Node.

The Cache Node shall answer to the end-user request and deliver the requested content to the end user.

The CDN controller is a centralized component, and CDN cache nodes are distributed within the Network and in multiple locations.

2.2.3 CDN Load Balancer

A CDN Load balancer objective is to distribute the demand to different nodes in the CDN taking into account different criteria including geographical proximity, network-wise proximity (number of hops, policies set by the operator like ASN, etc.) or load/performance parameters of the servers in the CDN. Additionally, the CDN load balancer also provides resilience and protection against contents server failure.

2.2.4 Surrogate Server

The CDN surrogate server interacts with other elements of the CDN for the control and distribution of content within it and with User Agents for the delivery of the content to the users. This behaviour corresponds with the surrogate in the WWW context as defined in [RFC3040]. The surrogate server provides resilience and protection against contents server failure.

2.2.5 Content Proxy

The content proxy is a server that acts as an intermediary for requests from clients seeking resources from the CDN content servers. The content proxy provides resilience and protection against contents server failure.

2.2.6 Content Peering Gateway

The content peering gateway is the element that interconnects different CDNs [RFC7337]. This element is a single point of failure.

2.3 vCDN Resiliency Requirements

[TBD - Pedro & Dan]

2.3.1 Automatic scale-out/scale-in for unpredictable traffic variation

One of the significant benefits of virtualization for CDN Providers is the elasticity of resource provisioning. This enables the CDN Providers to mitigate unpredictable traffic variation. Due to the unpredictability, the elastic resource management such as scaling out/scaling in should be automatically performed by Service Control

Entity and Pool Manager, rather than manually performed by human operators.

2.3.2 Quality assurance of content delivery

Since the quality of content delivery service is a key performance indicator of CDN Providers, it is crucial to assure the quality during the process of scaling as well as after the completion of scaling. In particular, geographical locations of added/remaining Cache Nodes should be taken into account. This is because these geographical locations have significant impacts on the quality of content delivery.

2.3.3 Minimum impact on interconnection interfaces

Interconnections between CDNs [RFC7336] have been recognized as a new opportunity of value creation for CDN Providers. In order for vCDN to foster this opportunity further, vCDN should minimize its impact on the interconnection interfaces between CDNs. In particular, scaling in/scaling out vCDN should not require any change of the architecture, protocols, and IP addresses/DNS names of interconnection points.

2.4 Service Degradation

CDN-based services will require suitable monitoring of performance metrics for delivering content. These include:

- o Connection time
- o DNS lookup time
- o Download time
- o First byte response
- o Latency
- o Page load time
- o Response to request time
- o Throughput
- o Error Rate
- o Packet Loss
- o Uptime

In the event of failure or service degradation, the ability to switch between comparative VNFs will be required.

2.5 Applicability of Virtual Network Function Pool (VNFPool)

The use case reveals the potential of VFNPOOL in simplifying the CDN architecture. Today's cache farms could be simplified in the way they are deployed and handled. Imagine you deploy a cache for a certain contents (e.g. newspaper web site) as a VNF. (This, as such, is a compelling use case, because the granularity is much finer and you might lower the minimum requirements for deploying a cache.)

The VFNPOOL protocol would control the way the VNF is deployed onto the VNFPOOL. Then, during its lifetime, it would control how it scales in or out.

Finally, it would also control the way the VNF is decommissioned. Apart from scaling in and out, the VNFPOOL protocol would also check for integrity and control the way a VNF can jump into another for resilience reasons.

This second kind of control should include state transfer and synchronization from the life to the backup VNF in some cases.

2.6 Coexistence of Virtualized and Non-virtualized Network Functions

With a CDN designed as loosely coupled software components a variety of scenarios of co-existing virtualized and non-virtualized components are possible.

Given that the CDN Controller is able to control Cache nodes deployed on virtualized and non-virtualized server instances in parallel the following scenarios are possible:

- o More centralized located Cache nodes can run on virtualized (Cloud) resources while Cache Nodes distributed deeper into the network might run on physical appliances for operational reasons.
- o Centralized cache cluster might run on dedicated non-virtualized server for performance reasons while Cache node instances distributed within in the network are running on virtualized resources available in other network devices
- o Within a migration scenario from non-virtualized to virtualized the legacy cache nodes can be kept in production until the end of their hardware life-cycle is reached (i.e. operation efficiency is still sufficient) while new capacity is added to the CDN by deploying the same software on virtualized resources.

3 Security Considerations

<Security considerations text TBD>

4 IANA Considerations

<IANA considerations text TBD>

5 References

5.1 Normative References

5.2 Informative References

[RFC3040] Cooper, I., Melve, I., and G. Tomlinson, "Internet Web Replication and Caching Taxonomy", RFC 3040, January 2001.

[RFC7337] Peterson, L., Davie, B., and R. Brandenburg, "Framework for CDN Interconnection", RFC7337, June 2014.

[RFC7336] L. Peterson, B. Davie, and R. van Brandenburg, Ed., "Framework for Content Distribution Network Interconnection (CDNI)", RFC 7336, August 2014.

[zong-vnfpool-problem-statement] Zong, N., "Problem Statement for Reliable Virtualized Network Function (VNF) Pool", January 2014.

[TRILOGY2] The trilogy2 Consortium, "trilogy2: Building the Liquid Net", <http://trilogy2.eu/>

6. Acknowledgement

This work is supported by the European FP7 Project "Trilogy2" [TRILOGY2] under grant agreement 317756.

Authors' Addresses

Pedro A. Aranda
Telefonica, I+D; GCTO Unit
Spain
Email: pedroa.aranda@telefonica.com

Daniel King
Lancaster University
UK

Email: d.king@lancaster.ac.uk

Masaki Fukushima
KDDI R&D Laboratories, Inc.
Japan

Email: fukusima@kddilabs.jp

Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 14, 2020

T. Dreibholz
SimulaMet
M. Tuexen
Muenster Univ. of Appl. Sciences
M. Shore
No Mountain Software
N. Zong
Huawei Technologies
September 11, 2019

The Applicability of Reliable Server Pooling (RSerPool) for Virtual
Network Function Resource Pooling (VNFPOOL)
draft-dreibholz-vnfpool-rserpool-applic-09

Abstract

This draft describes the application of Reliable Server
Pooling (RSerPool) for Virtual Network Function Resource
Pooling (VNFPOOL).

Status of This Memo

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 14, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal
Provisions Relating to IETF Documents
(<https://trustee.ietf.org/license-info>) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Abbreviations	2
2. Virtual Network Function Resource Pooling	3
3. Reliable Server Pooling	3
3.1. Introduction	3
3.2. Registrar Operations	4
3.3. Pool Element Operations	5
3.4. Takeover Procedure	5
3.5. Pool User Operations	6
3.5.1. Handle Resolution and Response	6
3.5.2. Pool Member Selection Policies	6
3.5.3. Handle Resolution and Response	6
3.6. Automatic Configuration	7
3.7. State Synchronisation	7
3.7.1. Cookies	7
3.7.2. Businesss Cards	8
3.8. Protocol Stack	8
3.9. Extensions	9
3.10. Reference Implementation and Deployment	9
4. Usage of Reliable Server Pooling	9
5. Security Considerations	10
6. IANA Considerations	10
7. Testbed Platform	10
8. Acknowledgments	10
9. References	10
9.1. Normative References	10
9.2. Informative References	12
Authors' Addresses	15

1. Introduction

1.1. Abbreviations

- o PE: Pool Element
- o PR: Pool Registrar
- o PU: Pool User
- o RSerPool: Reliable Server Pooling

- o SCTP: Stream Control Transmission Protocol
- o VNFPOOL: Virtual Network Function Resource Pooling

2. Virtual Network Function Resource Pooling

Virtualised Network Function (VNF) (e.g. vFW, vLB) -- as introduced in more detail in [I-D.zong-vnfpool-problem-statement] -- provides the same function as the equivalent network function (e.g. FW, LB), but is deployed as software instances running on general purpose servers via virtualisation platform. The main features of VNF include the following aspects:

1. A service consists of a sequence of topologically distributed VNF instances where the data connections are preferably directly established between the instances.
2. There are potentially more factors that cause VNF instance transition or even failure; VNF pool refers to a group of VNF instances providing same network function.

Virtualisation technology allows network function virtualisation operators to build a reliable VNF by pooling the underlying resources, such as CPU, storage, networking, etc. to form a cluster of VNF instances. VNF pool refers to a cluster or group of VNF instances providing same network function. Each VNF pool has a Pool Manager (PM) to manage the VNF instance such as instance selection, monitoring, etc. There will be a redundancy mechanism for a reliable PM to achieve reliable VNF. More details on VNF pool can be found in [I-D.zong-vnfpool-problem-statement].

3. Reliable Server Pooling

3.1. Introduction

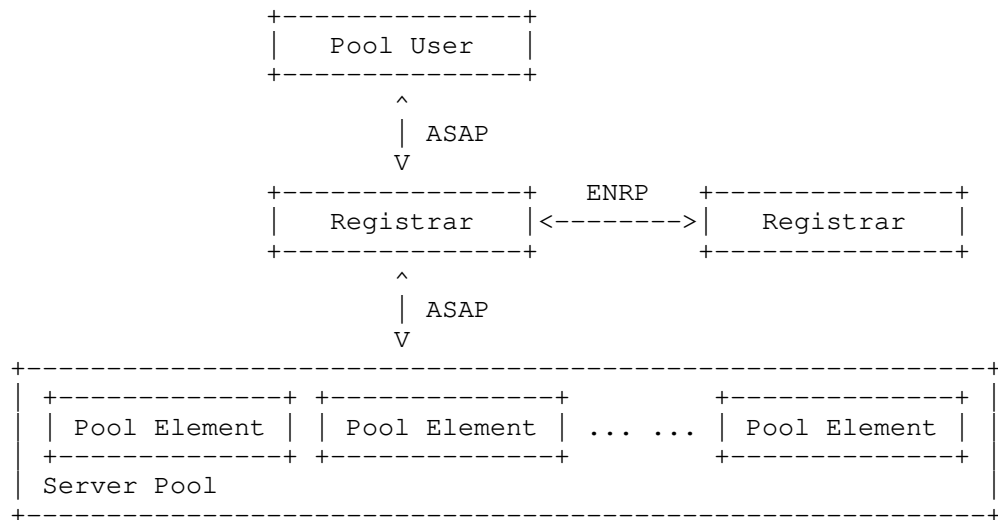


Figure 1

An overview of the RSerPool framework -- which is defined as RFC in [RFC5351] -- is provided in Figure 1. There are three types of components:

- o Pool Element (PE) denotes a server in a pool. PEs in the same pool provide the same service.
- o Pool User (PU) denotes a client using the service of a pool.
- o Pool Registrar (PR) is the management component for the pools.

The set of all pools within an operation scope (for example: an organisation, a company or a department) is denoted as handlespace. Clearly, a single PR would be a single point of failure. Therefore, PRs also have to be redundant. Within the handlespace, each pool is identified by a unique pool handle (PH).

3.2. Registrar Operations

The PRs of an operation scope synchronise their view of the handlespace by using the Endpoint haNdlespace Redundancy Protocol (ENRP, defined as RFCs in [RFC5353], [RFC5354]). In contrast to for instance the Domain Name System (DNS), an operation scope is restricted to a single administrative domain. That is, all of its components are under the control of the same authority (for example: a company). This property leads to small management

overhead, which also allows for RSerPool usage on devices having only limited memory and CPU resources (for example: telecommunications equipment). Nevertheless, PEs may be distributed globally to continue their service even in case of localised disasters (like for example an earthquake). Each PR in the operation scope is identified by a PR ID, which is a randomly chosen 32-bit number.

3.3. Pool Element Operations

Within their operation scope, the PEs may choose an arbitrary PR to register into a pool by using the Aggregate Server Access Protocol (ASAP, defined as RFCs in [RFC5352], [RFC5354]). The registration is performed by using an ASAP_REGISTRATION message. Within its pool, a PE is characterised by its PE ID, which is a randomly chosen 32-bit number. Upon registration at a PR, the chosen PR becomes the Home-PR (PR-H) of the newly registered PE. A PR-H is responsible for monitoring the availability of its PEs by ASAP_ENDPOINT_KEEP_ALIVE messages (to be acknowledged by a PE via an ASAP_ENDPOINT_KEEP_ALIVE_ACK message within a configured timeout). The PR-H propagates the information about its PEs to the other PRs of the operation scope via ENRP_UPDATE messages.

PEs re-register regularly in an interval denoted as registration lifetime and for information updates. Similar to the registration, a re-registration is performed by using another ASAP_REGISTRATION message. PEs may intentionally deregister from the pool by using an ASAP_DEREGISTRATION message. Also like for the registration, the PR-H makes the deregistration known to the other PRs within the operation scope by using an ENRP_UPDATE message.

3.4. Takeover Procedure

As soon as a PE detects the failure of its PR-H (that is: its request is not answered within a given timeout), it simply tries another PR of the operation scope for its registration and deregistration requests. However, as a double safeguard, the remaining PRs also negotiate a takeover of the PEs managed by a dead PR. This ensures that each PE again gets a working PR-H as soon as possible. The PRs of an operation scope monitor the availability of each other PR by using ENRP_PRESENCE messages, which are transmitted regularly. If there is no ENRP_PRESENCE within a given timeout, the peer is assumed to be dead and a so-called takeover procedure (see also [AINA2009] for more details) is initiated for the PEs managed by the dead PR: from all PRs having started this takeover procedure, the PR with the highest PR ID takes over the ownership of these PEs. The PEs are informed about being taken over by their new PR-H via an ASAP_ENDPOINT_KEEP_ALIVE with Home-flag set. The PEs are requested to adopt the sender of this Home-flagged message as their new PR-H.

3.5. Pool User Operations

3.5.1. Handle Resolution and Response

In order to access the service of a pool given by its PH, a PU requests a PE selection from an arbitrary PR of the operation scope, again by using ASAP. This selection procedure is denoted as handle resolution. Upon reception of a so-called ASAP_HANDLE_RESOLUTION message the PR selects the requested list of PE identities and returns them in an ASAP_HANDLE_RESOLUTION_RESPONSE message.

3.5.2. Pool Member Selection Policies

The pool-specific selection rule is denoted as pool member selection policy or shortly as pool policy. Two classes of load distribution policies are supported: non-adaptive and adaptive strategies (a detailed overview is provided by [Dre2006], [LCN2005], [IJIIDS2010], [IJHIT2008]). While adaptive strategies base their selections on the current PE state (which requires up-to-date information), non-adaptive algorithms do not need such data. A basic set of adaptive and non-adaptive pool policies is defined as RFC in [RFC5356].

Defined in [RFC5356] are the non-adaptive policies Round Robin (RR), Random (RAND) and Priority (PRIO) as well as the adaptive policies Least Used (LU) and Least Used with Degradation (LUD). While RR/RAND select PEs in turn/randomly, PRIO selects one of the PEs having the highest priority. PRIO can for example be used to realise a master/backup PE setup. Only if there are no master PEs left, a backup PE is selected. Round-robin selection is applied among PEs having the same priority. LU selects the least-used PE, according to up-to-date application-specific load information. Round robin selection is applied among multiple least-loaded PEs. LUD, which is evaluated by [ICDS2008-LUD], furthermore introduces a load decrement constant that is added to the actual load each time a PE is selected. It is used to compensate inaccurate load states due to delayed updates. An update resets the load to the actual load value.

3.5.3. Handle Resolution and Response

PE may fail, for example due to hardware or network failures. Since there is a certain latency between the actual failure of a PE and the removal of its entry from the handlespace -- depending on the interval and timeout for the ASAP_ENDPOINT_KEEP_ALIVE monitoring -- the PUs may report unreachable PEs to a PR by using an ASAP_ENDPOINT_UNREACHABLE message. A PR locally counts these reports for each PE and when reaching the threshold MAX-BAD-PE-REPORT (default is 3, as defined in the RFC [RFC5352]), the PR may decide to remove the PE from the handlespace. The counter of a PE is reset

upon its re-registration. More details on this threshold and guidelines for its configuration can be found in [IJAIT2009].

3.6. Automatic Configuration

RSerPool components need to know the PRs of their operation scope. While it is of course possible to configure a list of PRs into each component, RSerPool also provides an auto-configuration feature: PRs may send so-called announces, that is, ASAP_ANNOUNCE and ENRP_PRESENCE messages which are regularly sent over UDP via IP multicast. Unlike broadcasts, multicast messages can also be transported over routers (at least, this is easily possible within LANs). The announces of the PRs can be heard by the other components, which can maintain a list of currently available PRs. That is, RSerPool components are usually just turned on and everything works automatically.

3.7. State Synchronisation

RSerPool has been explicitly designed to be application-independent. Therefore, RSerPool has not intended to define special state synchronisation mechanisms for RSerPool-based applications. Such state synchronisation mechanisms are considered as tasks of the applications themselves. However, RSerPool defines two mechanisms to at least support the implementation of more sophisticated strategies: Cookies and Business Cards. Details on these mechanisms can also be found in Subsection 3.9.5 of [Dre2006].

3.7.1. Cookies

ASAP provides the mechanism of Client-Based State Sharing as introduced in [LCN2002]. Whenever useful, the PE may package its state in form of a state cookie and send it -- by an ASAP_COOKIE message -- to the PU. The PU stores the latest state cookie received from the PE. Upon PE failure, this stored cookie is sent in an ASAP_COOKIE_ECHO to the newly chosen PE. This PE may then restore the state. A shared secret known by all PEs of a pool may be used to protect the state from being manipulated or read by the PU.

While Client-Based State Sharing is very simple, it may be inefficient when the state changes too frequently, is too large (the size limit of an ASAP_COOKIE/ASAP_COOKIE_ECHO is 64 KiB) or if it must be prevented that a PU sends a state cookie to multiple PEs in order to duplicate its sessions.

3.7.2. Businesss Cards

Depending on the application, there may be constraints restricting the set of PEs usable for failover. The ASAP_BUSINESS_CARD message is used to inform peer components about such constraints.

The first case to use a Business Card is if only a restricted set of PEs in the pool may be used for failover. For example, in a large pool, each PE can share its complete set of session states with a few other PEs only. This keeps the system scalable. That is, a PE in a pool of n servers does not have to synchronise all session states with the other $n-1$ PEs. In this case, a PE has to tell its PU the set of PE identities being candidates for a failover using an ASAP_BUSINESS_CARD message. A PE may update the list of possible failover candidates at any time by sending another Business Card. The PU has to store the latest list of failover candidates. Of course, if a failover becomes necessary, the PU has to select from this list using the appropriate pool policy -- instead of performing the regular PE selection by handle resolution at a PR. Therefore, some literature also denotes the Business Card by the more expressive term "last will".

In symmetric scenarios, where a PU is also a PE of another pool, the PU has to tell this fact to its PE. This is realised by sending an ASAP_BUSINESS_CARD message to the PE, providing the PH of its pool. Optionally, also specific PE identities for failover may be provided. The format remains the same as explained in the previous paragraph. If the PE detects a failure of its PU, the PE may -- now in the role of a PU -- use the provided PH for a handle resolution to find a new PE or use the provided PE identities to select one. After that, it can perform a failover to that PE.

3.8. Protocol Stack

The protocol stack of a PR provides ENRP and ASAP services to PRs and PEs/PUs respectively. But between PU and PE, ASAP provides a Session Layer protocol in the OSI model. From the perspective of the Application Layer, the PU side establishes a session with a pool. ASAP takes care of selecting a PE of the pool, initiating and maintaining the underlying transport connection and triggering a failover procedure when the PE becomes unavailable.

The Transport Layer protocol is by default SCTP (as defined in [RFC4960]) -- except for the UDP-based automatic configuration announces (see Section 3.6) -- over possibly multi-homed IPv4 and/or IPv6. SCTP has been chosen due to its support of multi-homing and its reliability features (see also [Dre2012]).

3.9. Extensions

A couple of extensions to RSerPool are existing: Handle Resolution Option defined in [I-D.dreibholz-rserpool-asap-hropt] improves the PE selection by letting the PU tell the PR its required number of PEs to be selected. ENRP Takeover Suggestion introduced in [I-D.dreibholz-rserpool-enrp-takeover] ensures load balancing among PRs. [I-D.dreibholz-rserpool-delay] defines a delay-sensitive pool policy. [RFC5525] defines an SNMP MIB for RSerPool.

3.10. Reference Implementation and Deployment

RSPLIB is the Open Source reference implementation of RSerPool. It is currently -- as of February 2016 -- available for Linux, FreeBSD, MacOS and Solaris. It is actively maintained. Particularly, it is also included in Ubuntu Linux as well as in the FreeBSD ports collection. RSPLIB can be downloaded from [RSerPoolPage]. Further details on the implementation are available in [Dre2006], [GlobeCom2010-Demo].

RSerPool with RSPLIB is deployed in a couple of Open Source projects, including the SimProcTC Simulation Processing Tool-Chain for distributing simulation runs in a compute pool (see [OMNeTWorkshop2008] as well as the simulation run distribution project explained in [Dre2012] for a practical example) as well as for service infrastructure management in the NorNet Core research testbed (see [ComNets2013-Core], [PAMS2013-NorNet]).

4. Usage of Reliable Server Pooling

**** TO BE DISCUSSED! ****

The following features of RSerPool can be used for VNFPOOL:

- o Pool management.
- o PE selection with pool policies.
- o Session management with help of ASAP_BUSINESS_CARD.

The following features have to be added to RSerPool itself:

- o Support of TCP including MPTCP as additional/alternative transport protocols.
- o Possibly add some special pool policies?

- o See also [I-D.dreibholz-rserpool-nextgen-ideas] for ideas on a next generation of RSerPool.

The following features have to be provided outside of RSerPool:

- o State synchronisation for VNFPOOL.
- o Pool Manager functionality as an RSerPool-based service.

5. Security Considerations

Security considerations for RSerPool can be found in [RFC5355]. Furthermore, [IJIIDS2010] examines the robustness of RSerPool systems against attacks.

6. IANA Considerations

This document introduces no additional considerations for IANA.

7. Testbed Platform

A large-scale and realistic Internet testbed platform with support for Reliable Server Pooling and the underlying SCTP protocol is NorNet. A description of and introduction to NorNet is provided in [PAMS2013-NorNet], [Sydney2019], [Haikou2019-NorNet-Tutorial]. Further information can be found on the project website [NorNet-Website] at <https://www.nntb.no>.

8. Acknowledgments

The authors would like to thank Xing Zhou for the friendly support.

9. References

9.1. Normative References

- [I-D.dreibholz-rserpool-asap-hropt]
Dreibholz, T., "Handle Resolution Option for ASAP", draft-dreibholz-rserpool-asap-hropt-24 (work in progress), March 2019.
- [I-D.dreibholz-rserpool-delay]
Dreibholz, T. and X. Zhou, "Definition of a Delay Measurement Infrastructure and Delay-Sensitive Least-Used Policy for Reliable Server Pooling", draft-dreibholz-rserpool-delay-23 (work in progress), March 2019.

- [I-D.dreibholz-rserpool-enrp-takeover]
Dreibholz, T. and X. Zhou, "Takeover Suggestion Flag for the ENRP Handle Update Message", draft-dreibholz-rserpool-enrp-takeover-21 (work in progress), March 2019.
- [I-D.dreibholz-rserpool-nextgen-ideas]
Dreibholz, T., "Ideas for a Next Generation of the Reliable Server Pooling Framework", draft-dreibholz-rserpool-nextgen-ideas-11 (work in progress), March 2019.
- [I-D.zong-vnfpool-problem-statement]
Zong, N., Dunbar, L., Shore, M., Lopez, D., and G. Karagiannis, "Virtualized Network Function (VNF) Pool Problem Statement", draft-zong-vnfpool-problem-statement-06 (work in progress), July 2014.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<https://www.rfc-editor.org/info/rfc4960>>.
- [RFC5351] Lei, P., Ong, L., Tuexen, M., and T. Dreibholz, "An Overview of Reliable Server Pooling Protocols", RFC 5351, DOI 10.17487/RFC5351, September 2008, <<https://www.rfc-editor.org/info/rfc5351>>.
- [RFC5352] Stewart, R., Xie, Q., Stillman, M., and M. Tuexen, "Aggregate Server Access Protocol (ASAP)", RFC 5352, DOI 10.17487/RFC5352, September 2008, <<https://www.rfc-editor.org/info/rfc5352>>.
- [RFC5353] Xie, Q., Stewart, R., Stillman, M., Tuexen, M., and A. Silverton, "Endpoint Handlespace Redundancy Protocol (ENRP)", RFC 5353, DOI 10.17487/RFC5353, September 2008, <<https://www.rfc-editor.org/info/rfc5353>>.
- [RFC5354] Stewart, R., Xie, Q., Stillman, M., and M. Tuexen, "Aggregate Server Access Protocol (ASAP) and Endpoint Handlespace Redundancy Protocol (ENRP) Parameters", RFC 5354, DOI 10.17487/RFC5354, September 2008, <<https://www.rfc-editor.org/info/rfc5354>>.
- [RFC5355] Stillman, M., Ed., Gopal, R., Guttman, E., Sengodan, S., and M. Holdrege, "Threats Introduced by Reliable Server Pooling (RSerPool) and Requirements for Security in Response to Threats", RFC 5355, DOI 10.17487/RFC5355, September 2008, <<https://www.rfc-editor.org/info/rfc5355>>.

- [RFC5356] Dreibholz, T. and M. Tuexen, "Reliable Server Pooling Policies", RFC 5356, DOI 10.17487/RFC5356, September 2008, <<https://www.rfc-editor.org/info/rfc5356>>.
- [RFC5525] Dreibholz, T. and J. Mulik, "Reliable Server Pooling MIB Module Definition", RFC 5525, DOI 10.17487/RFC5525, April 2009, <<https://www.rfc-editor.org/info/rfc5525>>.

9.2. Informative References

- [AINA2009] Zhou, X., Dreibholz, T., Fa, F., Du, W., and E. Rathgeb, "Evaluation and Optimization of the Registrar Redundancy Handling in Reliable Server Pooling Systems", Proceedings of the IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA) Pages 256-262, ISBN 978-0-7695-3638-5, DOI 10.1109/AINA.2009.25, May 2009, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/AINA2009.pdf>>.
- [ComNets2013-Core] Gran, E., Dreibholz, T., and A. Kvalbein, "NorNet Core - A Multi-Homed Research Testbed", Computer Networks, Special Issue on Future Internet Testbeds Volume 61, Pages 75-87, ISSN 1389-1286, DOI 10.1016/j.bjp.2013.12.035, March 2014, <<https://www.simula.no/file/simulasimula2236pdf/download>>.
- [Dre2006] Dreibholz, T., "Reliable Server Pooling - Evaluation, Optimization and Extension of a Novel IETF Architecture", March 2007, <https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-16326/Dre2006_final.pdf>.
- [Dre2012] Dreibholz, T., "Evaluation and Optimisation of Multi-Path Transport using the Stream Control Transmission Protocol", Habilitation Treatise, March 2012, <https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-29737/Dre2012_final.pdf>.
- [Globecom2010-Demo] Dreibholz, T. and M. Becke, "The RSPLIB Project - From Research to Application", Demo Presentation at the IEEE Global Communications Conference (GLOBECOM), December 2010, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/Globecom2010-Demo.pdf>>.

[Haikou2019-NorNet-Tutorial]

Dreibholz, T., "NorNet at Hainan University: Getting Started with NorNet Core", Tutorial at Hainan University, College of Information Science and Technology (CIST), April 2019, <<https://www.simula.no/file/china2019-nornet-tutorialpdf/download>>.

[ICDS2008-LUD]

Zhou, X., Dreibholz, T., and E. Rathgeb, "A New Server Selection Strategy for Reliable Server Pooling in Widely Distributed Environments", Proceedings of the 2nd IEEE International Conference on Digital Society (ICDS) Pages 171-177, ISBN 978-0-7695-3087-1, DOI 10.1109/ICDS.2008.12, February 2008, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/ICDS2008-LUD.pdf>>.

[IJAIT2009]

Dreibholz, T. and E. Rathgeb, "Overview and Evaluation of the Server Redundancy and Session Failover Mechanisms in the Reliable Server Pooling Framework", International Journal on Advances in Internet Technology (IJAIT) Number 1, Volume 2, Pages 1-14, ISSN 1942-2652, June 2009, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/IJAIT2009.pdf>>.

[IJHIT2008]

Dreibholz, T. and E. Rathgeb, "An Evaluation of the Pool Maintenance Overhead in Reliable Server Pooling Systems", SERSC International Journal on Hybrid Information Technology (IJHIT) Number 2, Volume 1, Pages 17-32, ISSN 1738-9968, April 2008, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/IJHIT2008.pdf>>.

[IJIIDS2010]

Dreibholz, T., Zhou, X., Becke, M., Pulinthanath, J., Rathgeb, E., and W. Du, "On the Security of Reliable Server Pooling Systems", International Journal on Intelligent Information and Database Systems (IJIIDS) Number 6, Volume 4, Pages 552-578, ISSN 1751-5858, DOI 10.1504/IJIIDS.2010.036894, December 2010, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/IJIIDS2010.pdf>>.

- [LCN2002] Dreibholz, T., "An Efficient Approach for State Sharing in Server Pools", Proceedings of the 27th IEEE Local Computer Networks Conference (LCN) Pages 348-349, ISBN 0-7695-1591-6, DOI 10.1109/LCN.2002.1181806, November 2002, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/StateSharing-Paper-ShortVersion.pdf>>.
- [LCN2005] Dreibholz, T. and E. Rathgeb, "On the Performance of Reliable Server Pooling Systems", Proceedings of the IEEE Conference on Local Computer Networks (LCN) 30th Anniversary Pages 200-208, ISBN 0-7695-2421-4, DOI 10.1109/LCN.2005.98, November 2005, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/LCN2005.pdf>>.
- [NorNet-Website] Dreibholz, T., "NorNet -- A Real-World, Large-Scale Multi-Homing Testbed", Online: <https://www.nntb.no/>, 2017, <<https://www.nntb.no/>>.
- [OMNeTWorkshop2008] Dreibholz, T. and E. Rathgeb, "A Powerful Tool-Chain for Setup, Distributed Processing, Analysis and Debugging of OMNeT++ Simulations", Proceedings of the 1st ACM/ICST International Workshop on OMNeT++ ISBN 978-963-9799-20-2, DOI 10.4108/ICST.SIMUTOOLS2008.2990, March 2008, <<https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/OMNeTWorkshop2008.pdf>>.
- [PAMS2013-NorNet] Dreibholz, T. and E. Gran, "Design and Implementation of the NorNet Core Research Testbed for Multi-Homed Systems", Proceedings of the 3rd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS) Pages 1094-1100, ISBN 978-0-7695-4952-1, DOI 10.1109/WAINA.2013.71, March 2013, <<https://www.simula.no/file/threfereedinproceedingsreference2012-12-207643198512pdf/download>>.
- [RSerPoolPage] Dreibholz, T., "Thomas Dreibholz's RSerPool Page", Online: <https://www.uni-due.de/~be0001/rserpool/>, 2019, <<https://www.uni-due.de/~be0001/rserpool/>>.

[Sydney2019]

Dreibholz, T., "NorNet at the University of Sydney: From Simulations to Real-World Internet Measurements for Multi-Path Transport Research", Invited Talk at University of Sydney, January 2019, <<https://www.simula.no/file/sydney2019-presentationpdf/download>>.

Authors' Addresses

Thomas Dreibholz
Simula Metropolitan Centre for Digital Engineering
Pilestredet 52
0167 Oslo, Oslo
Norway

Phone: +47-6782-8200
Fax: +47-6782-8201
Email: dreibh@simula.no
URI: <https://www.simula.no/people/dreibh>

Michael Tuexen
Muenster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt, Nordrhein-Westfalen
Germany

Email: tuexen@fh-muenster.de
URI: <https://www.fh-muenster.de/fb2/personen/professoren/tuexen/>

Melinda Shore
No Mountain Software
PO Box 16271
Two Rivers, Alaska 99716
U.S.A.

Phone: +1-907-322-9522
Email: melinda.shore@nomountain.net
URI: <https://www.linkedin.com/pub/melinda-shore/9/667/236>

Ning Zong
Huawei Technologies
101 Software Avenue
Nanjing, Jiangsu 210012
China

Email: zongning@huawei.com

URI: <https://cn.linkedin.com/pub/ning-zong/15/737/490>

VNF BOF
Internet-Draft
Intended status: Informational
Expires: January 5, 2015

S. Hares
Huawei
July 4, 2014

Use Cases for Resource Pools with Virtual Network Functions (VNFs)
draft-hares-vnf-pool-use-case-02

Abstract

This draft describes use cases the author has observed in demonstrations or deployments for virtualized network functions (VNFs) supported by VNF Pools. Several of these demonstrations combined VNF Pools into VNFsets. The use cases were: cloud bursting, parental controls, load balancer for multipath (L1-L7), WAN optimization that runs either between access nodes and Data Centers, WAN optimization between mobile phones and Data Centers (through access nodes), application placement optimization, and optimized placement of web applications utilizing minimal data transfer.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terms	3
3. Use Case List	4
4. Cloud Bursting Use Case	5
5. Stateful Parental Controls	6
6. Load balancer	7
7. Android phone TCP WAN optimization	9
8. SOHO device optimization	10
9. Application Scaling	11
10. IANA Considerations	12
11. Security Considerations	12
12. References	12
12.1. Normative References	12
12.2. Informative References	12
Author's Address	13

1. Introduction

This draft focuses on providing one person's observations on the deployment of Virtualized Network Functions which are supported by VNF Pool where the VNF Pools may be grouped into VNF Sets. This version of the draft no longer needs to explain the basic architecture and problems since [I-D.zong-vnfpool-problem-statement] provides an excellent description of the following:

- o Terminology of VNF, VNF Pools, elements of VNF Pools, VNF Pool Managers, and VNF Sets;
- o Challenges to the reliability of VNFs (without Pools);
- o Challenges to reliability within VNFs (redundancy and state synchronization),
- o Interactions with Service Control Entity managing the VNF functions
- o and the needs for reliable transport

This document simply introduces unique terms, and then describes authors experience the VNF Pools and VNF Managers when the VNF Pools contain only one type of function. The VNF Pools may operate in a

set of VNF Pools. This document no longer examines VNF Set management because is out of the scope of the VNF Charter.

Virtual Network functions supported by Virtual Network Pools and organized into Virtual Sets have been observed to be more reliable and be able to expand (or contract horizontally). By being more reliable, this author observed that individual failures of virtual functions due to software or system constraints (load) were survived by switching over to another NFV function within the VNF Pool. For example, with compatible software functions running, the current and previous software ran a network applications (E.g. open source NAT or open source DPI), a failure on one VNF running the current software could quickly be replaced by a "hot standby" in the Pool running the previous version. Upon increased traffic, one VNF function (for firewalls) could be expanded to multiple firewalls each handling a portion of the traffic. In a sense, the VNF expands horizontally to handle the increased traffic. In the same way, as traffic diminished, this VNF can contract.

This document describes each use case by describing the application and how the VNF function when operating within VNF Pools within the VNF Set that makes up the application. While some of these use cases had multiple VNF Sets, VNF Set management is outside of the scope of the VNF Pool work. Therefore, the explanations have been simplified to consider all the VNF Pools into one set.

One final note, the author knows she has only provided abstract descriptions of these deployments, but out of respect for products and companies the abstract description is best.

2. Terms

The VNF Problem statement [I-D.zong-vnfpool-problem-statement] defines the terms reliability, VNF, VNF Pool, VNF Pool Element, VNF Pool User, VNF Pool Manager, and VNF Set. This draft uses these definitions. The following definitions are not defined within the VNF problem statement: Cloud Bursting, Stateful parental controls, WAN optimization, and application placement. These terms are defined below.

Cloud Bursting: the ability for Virtual processing to burst through the limits of one virtual environment and automatically transfers a portion of the processing to another virtual environment.

Stateful parental controls: the ability for network access devices to have content filters that react to traffic, location, and user. These controls follow the user across multiple access points within a home network, or in a carrier network.

WAN optimization: the ability to optimize traffic across a Wide-Area network. WAN optimization often makes use of TCP FLOW optimizations (with IETF TCP features) and TCP de-duplication of packets,

Application placement: ability for coordinating software to place applications based a combination of compute resources, data storage, network service, and security concerns. Application placement may involve movement of some application data, movement of some applications (data and compute), and movement of network resources to service the applications. One type of network resource movement is the movement of virtual network functions (VNFs) which are defined, created, allocated with resources in a way to provide an integral unit to the application placement control software.

OTT (Over the Top): This industry terms implies an overlay network that is overlaid on existing networks as a virtual network.

Shared risk group (SRG): Shared risk groups occur when different VNFs in a VNF Pool all exist upon the same instance of a virtual form or hypervisor. When a hypervisor fails, all the VNF instances on the same hypervisor will fail,

3. Use Case List

The use cases described in this draft are:

- o Cloud Bursting
- o stateful parental controls implemented in access nodes and firewalls (stateful and regular)
- o load balancer doing multipath (supports L1-L7 optimization),
- o WAN optimization between access nodes and Data Centers,
- o WAN optimization between mobile phones through access nodes to/from Data center (E.g Riverbed WAN),
- o Application placement optimization using optimized DNS and DHCP VNFs,
- o Application placement optimization to minimize data transfer.

The uses cases are done in the order of VNF sets to VNF single operations. The Cloud bursting obviously takes a set of VNF Pools to lift up services in a cloud environment and move these to another cloud environment.

Deployment of VNF functions into critical network functions requires that multiple sources exist to reduce risk of software or hardware issues, and to respond to economic pressure to continually improve while reducing prices. Multi-vendor sources for these VNF, VNF Pools, and VNR sets comes at the price of designing (or adopting an existing) interoperability VNF Pool manager for VNF Pools.

4. Cloud Bursting Use Case

Description:

Three cases of cloud bursting exist. Public clouds adding more resources upon demand. Private clouds adding more resources upon demand from private cloud resources. Private clouds adding more resources from the public cloud. In the public/private cloud, the orchestration system looks within pools of additional resources to fit the request for more resources for a particular time. Verizon provided examples of cloud bursting at ONS 2012, and Terremark utilizes cloud bursting to obtain more resources (<http://www.terremark.com/services/it-infrastructure/cloud-services/enterprise-cloud/architecture/>) operating over open-source hypervisors (2012, 2013).

VNFs within the VNF Pools operate as management systems and networks router/switches (virtual switches, routers, end systems) to spin up additional transport process (TCP/STCP) and move work jobs via standard interfaces (libvirt, CLI, REST, and JASON), and provide standardized value-added functions. These value-added functions include the following:

- o VNFs in VNF Pools of system monitoring and orchestration
- o VNF in VNF Pools for virtual firewall to protect the data
- o VNF in VNF Pools for DPI or DDOS during
- o VNF in VNF specialized DNS that controls private/public cloud move
- o VNF in VNF WAN applications that create a large pipeline for for movement of data and applications within Cloud (Private/Public) or between clouds
- o VNFs in VNF Pools for smart access to the cloud

Why VNF in VNF Pools for network router/switch or host system functions

VNFs in VNF Pools allow cloud bursting to temporarily expand horizontally to take the load as the processing groups move between clouds. Each of the functions has a scaling within its own pool which allows the bursts of effort to grab or release the amount of functions. The VNFs doing system monitoring of the move and the orchestration are also included in the features that grab or release functions.

Why VNF Pools:

Bursty nature of action of Cloud Bursting requires being able utilize VNFs within Pools to expand horizontally for the estimated cloud bursting activities. However, if the cloud bursting expands beyond the resources estimated by the orchestration software then the VNFs within the pool can expand the service.

Why Multi-vendor interoperable VNF Pools?:

Cloud bursting is a critical business infrastructure which needs highly reliable software that can be maintained by Cloud operations. Critical infrastructure requires multi-sources. Either the Cloud operations creates a team to maintain VNF Pool software from Open Source code bases, or the equipment vendors provide interoperable VNF Pool Managers and VNF Pools that run across multiple platforms.

5. Stateful Parental Controls

Description:

Parental content filters are targeted filters that are installed based on an identification of a user. When the centralized controller detects the User (via traffic pattern, role identification (ABFAB, HTTP)), an orchestration manager installs the appropriate software to guarantee filters. Two types of security exist: authentication and authorization. In authentication, ACL and other port based filtering is set per customer for the user. This filtering may block, prioritize, or transfer to a black hole recording device different traffic. In authorization, the systems create a web of trust via an identity server (for HTTP 1.0 SAML template defined by OASIS and IETF ABFAB information for non-http).

The following is a list of some of the VNF functions found in VNF Pools in the Stateful Parental Control Model

- o VNF Pool for the specialized Access filters
- o VNF Pool for open source DPIs (snort, etc.) to find "inappropriate" material,

- o VNF Pool for specialized DPI inspection,
- o VNF Pool probes on hyper-visors,
- o VNF POol for management functions depositing configuration in Open Flow switches, Ethernet Switches, Virtual switches, routers, firewalls, and access nodes.
- o VNF Pool for access firewall
- o VNF Pool for spam filters for mail
- o VNF Pool for DDOS software,
- o VNF Pool for DNS/DHCP servers that allow the linking of the the Public services to a instantly created VNFs for specialized access
- o VNF Pool to move filters within Cloud (Private/Public) or between clouds in anticipation of the persons movement (If in central London, spread to other access nodes along public transportation (Tube) lines or to hotels.).
- o VNF Pool to do additional user identification of the systems

Why VNF Pools

The bursty nature of user access is dependent on the detection of the movement of the user. At the moment the public software identifies the user, this VNF Pool set operates to expand horizontally to provide the necessary service to provide these parental features. The VNF Pools allow groups of these parental ' families to be instantiated.

Why inter-operable VNF Pool Managers

The VNF functions may go between the mobile devices the user moves with (E.g. Android Pad or Android Phone) and the local network systems supported by the Carrier, the hotel, or the airport systems. Inter-operable VNF Pool Managers means that some NVF functions may move from Android Pad /Android Phone to carrier's equipment.

6. Load balancer

Description:

Load balancers (such as Riverbed or Cisco) look to balance traffic in different layers of the stack (L1-L7). SDN meta controllers (OpenDaylight, Vyatta) monitor work with the time-critical OTT

control process (which creates and manages the OTT VPNs (L2/L3/MPLS)) to determine where the load is at any specific time, and to track it over time. The SDN orchestration devices work with the SDN OTT control process to adjust to readjust the load at L1-L7.

The VNF functions that use VNF Pools in the load balancing service are:

- o VNFs for network probes in all devices (mobile phone, ipad, access devices, vswitch, vrouter, tcp optimizer, DPI, hypervisors, VMs dummung storage, VMs creating the network;
- o VNFs for depositing configuration in Ethernet switches (open-flow or IEEE 802.1), routers, firewalls, access nodes;
- o VNFs for firewall;
- o VNFs to do Traffic capacity/load balance calculation;
- o VNFs running orchestrator monitor/change algorithms; and
- o VNFs to users or specific traffic to aid in load balancing.

Why VNF Pools:

True end-to-end Load balancing requires load balancing across multiple layers with VNF pools to support different functions. Multi-vendors solutions will allow meta controllers to balance traffic to reduce costs in networks. Current Enterprise customers find the load balancing operates with TCP WAN optimization to utilize all network bandwidth effectively.

Why inter-operable VNF Pool Managers

Network probes, network traffic capacity calculation, and configuration of changes operate either when traffic thresholds are exceeded or upon period timers. Each of these functions has bursty needs needing the ability to expand horizontally.

Firewalls are traffic based which may be bursty or steady state depending on the application profiles. VNF Pools allow for the horizontal expansion during bursts.

Long lived traffic flows may be identified by looking for users or application traffic patterns. This type of processing function has a "DPI-Like" processing quality that make require quick examination of some data. VNF support in VNF Pools allows the assurance of this type of support

7. Android phone TCP WAN optimization

Description:

Android phones and Android tablets often communicate across the LTE/WiFi connections. Optimization of the link for the low-bandwidth of LTE or Wifi connections, and the switch between LTE and WiFi requires monitoring of traffic, choosing link, optimizing TCP (Window and removing duplicates).

The VNFs that are aided by VPN Pools in this application includes:

- o VNFs for probes in all devices (mobile phone, mobile pads, Wifi enabled nodes, LTE IP RAN nodes)
- o VNFs for depositing configuration in SDN access nodes (Wifi or LTE)
- o VNFs for to handle remote phone parameter adjustments;
- o VNFs to do firewalls (E.g traffic not allowed over LTE due to customer policy);
- o VNFs for TCP data de-duplication process;
- o VNFs for Traffic capacity/load balance calculation (see Football stadium problem below);
- o VNFs for best processing of Video traffic or best network to pull Video traffic from;
- o VNFs to identify user or user traffic and
- o VNFs to interface to secure data processes.

One scenario to consider is the football stadium scenario. A person takes the IPAD to watch the close up replays or send email. During fourth quarter, the person receive an urgent call to go home and walks with the IPAD down the street to the metro-system to return home. On the way, the person is utilizing the IPAD to send mail, watch the football game, and do Skype calls.

This scenario is similar in needs to the parental controls. The differences are TCP data de-duplication to improve WAN traffic and specialized Video traffic handling, plus the mobile phone management and security.

Why VNF Pools:

The football user case illustrates how the network functions are used in bursts. The VNF Pools allow these functions to expand out to fit the users needs. The football example also shows how events can cause massive numbers of these bursty users to occur at the same time. Again, the expansion out for these events without reducing service is key to the quality of user experience for mobile phone or mobile pad users.

Why Inter-operable VPN Pools handled by VPN Pool Managers:

Phones systems do not want a single vendor for all features. Multiple interoperable access nodes and Android pad/tablet implementations require these VNF pools. The football stadium may require that several mobile operators or mobile or cable operators work together to provide this service.

8. SOHO device optimization

Description:

SOHO devices using SDN VM technology must balance traffic movement between small cells (WiFi or femtocells). Access policies must be configured for restriction on this policy.

The VNFs that VNF Pools in this application are:

- o VNFs for probes in all devices (mobile phone, mobile pads, WiFi enabled nodes, LTE or femtocells)
- o VNFs for VPN to user identification and security.
- o VNFs for depositing configuration in access nodes (Wifi, L),
- o VNFs for handling remote phone parameter adjustments;
- o VNFs for firewall (traffic not allowed over LTE);
- o VNFs for TCP data de-duplication process;
- o VNFs for Traffic capacity/load balancing over single/multiple soho links;
- o VNFs to allow applications load balance across internal soho links based on traffic needs and use policy; and
- o VNFs for VPN to user identification and security.

Why VNF Pools:

SOHO devices will have limited resources for handling probes to find local devices, change configurations in access devices, adjust remote phone parameters, firewall traffic, and perform WAN optimization (TCP de-duplication, prioritizing of traffic (like phones) or load balancing). However, SOHOs may only need the probes, configurations changes, and phone adjustments when users arrive into the home. The data related VNF functions will occur as the SOHO office begins to transfer data. The VNF pools allow the VNF function to scale up/down via horizontal expansion.

VPN Pool Growth/Shrinking:

The VPN Pool Manager can handle increasing or decreasing the VNF Pool size. Cooperating VNF Pool Managers can be seen to be useful in this use case, but the cooperating VNF pool managers are outside the scope of the VNF within a VNF Pool.

9. Application Scaling

Description:

Applications may be placed in a variety of hypervisors. The rapid deployment of applications on services may allow millions of applications to be available within the cloud. Creating a effective lookup for the applications or redirecting applications takes an Network Virtual environment that controls DHCP, DNS, and http access rapidly. 2 Million URI references for each access node is possible given the current growth.

VNF within the cloud must scale up to handle the VNF services required by the network infrastructure. This includes the network information functions of DNS, DHCP, URL processing, AAA (Diameter/Radius). Fast enactment of these network functions allows an on-demand creation of a multi-tenancy overlay (IETF NV03).

The VNFs operate in VNF Pools in this application are:

- o VNFs for AAA functions (Diameter, Radius);
- o VNFs for DNS functions;
- o VNFs for DHCP functions
- o VNFs for specialized URL/URI processing;
- o VNFs for handling remote probes on these virtual information functions;

- o VNFs for handling remote configuration of these virtual information functions;
- o VNFs for Traffic capacity/load balance calculation;
- o VNFs for determine optimum placement of application (and application's backup services) to optimize CPU compute, storage or data
- o VNFs for VPN to user identification and permissions to use data; and

Why VNF in VNF Pools

User load patterns or access patterns will impact how much load the network information VNF functions (DNS, DHCP, URL processing, AAA (Diameter/Radius) encounter. The VNF Pools with a good VNF Pool manager can spread the load locally or between different systems.

The applications and the application usage will also determine how loaded the VNF Function is that monitors CPU utilization, storage, and network resources. Again, the VNF supported by VNF Pools can expand or shrink horizontally.

The rest of the VNF functions needs for VNF Pools have been described above.

10. IANA Considerations

This document includes no request to IANA.

11. Security Considerations

This document has no security issues as just contains use cases.

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

12.2. Informative References

[I-D.zong-vnfpool-problem-statement]

Zong, N., Dunbar, L., Shore, M., Lopez, D., and G.
Karagiannis, "Virtualized Network Function (VNF) Pool
Problem Statement", draft-zong-vnfpool-problem-
statement-06 (work in progress), July 2014.

Author's Address

Susan Hares
Huawei
7453 Hickory Hill
Saline, CA 48176
USA

Email: shares@ndzh.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 31, 2015

D. King
Lancaster University
M. Liebsch
NEC
P. Willis
BT
J. Ryoo
ETRI
January 31, 2015

Virtualisation of Mobile Core Network Use Case
draft-king-vnfpool-mobile-use-case-02

Abstract

Accessing the Internet via mobile data services using smartphones, tablets, and mobile data USB dongles has increased rapidly, as high-speed packet data networks provide the bandwidth required for today's Internet applications. Mobile operators will continue to evolve their core networks to the Long Term Evolution (LTE) Evolved Packet Core (EPC) to meet the mobility, latency and bandwidth requirements for mobile data users.

Network Functions Virtualization (NFV) looks to reduce mobile core network complexity and related operational issues by leveraging standard IT virtualization technologies and consolidate different types of network equipment onto commodity hardware.

This use case document provides resiliency requirements for virtualization of the LTE mobile core network, known as virtualized EPC (vEPC).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
This Internet-Draft will expire on July 31, 2015.

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	2
1.1 Operator Benefits of Virtualization.....	3
2. Terminology.....	3
3. Virtual Evolved Packet Core (vEPC).....	4
3.1 Mobile Core Network Components.....	5
3.1.1 Mobile Network Nodes.....	5
3.1.2 Mobile Network Functions.....	5
3.2 Resiliency Requirements for the vEPC.....	6
3.2.1 Handling Unplanned Traffic Peaks.....	7
3.2.2 Scaling of Resources and Functions.....	7
3.2.3 vEPC Failure Handling.....	10
3.2.4 State Synchronization.....	12
3.3 Applicability of Virtual Network Function Pool (VNF Pool)...	12
3.3.1 VNF Pool Definitions.....	13
4. IANA Considerations.....	13
5. Security Considerations.....	13
6. References.....	13
6.1 Normative References.....	13
6.2 Informative References.....	13
Authors' Addresses.....	13

1. Introduction

Mobile operators have deploying Long Term Evolution (LTE) Evolved Packet Core (EPC) to meet the mobility, latency and bandwidth requirements for a variety of mobile data users. The EPC is the latest evolution of the [3GPP-R8] core network architecture, and is based on IP.

The EPC architecture is said to have a "flat architecture" with

minimal components and functions. Principally the design is intended to minimise the number of function nodes required and protocol conversation of mobile data traffic. However, EPC elements are bespoke stand-alone hardware (i.e., different boxes for different functions). Network operators have identified that this approach costly and inflexible.

The ETSI Network Functions Virtualization (NFV) Industry Steering Group (ISG) published a set of use cases [NFV-ISG-UC]. One key use case described the Virtualisation of Mobile Core Network and IP Multimedia Subsystem (IMS), known as the vEPC.

The NFV approach takes the EPCs functional elements and runs them as software instances (Virtual Appliances) on high-volume industry-standard generic servers. This approach has number of advantages including:

- o Reducing: Cost, Power, Space and Complexity.
- o Increasing: Flexibility, Scalability and Consolidation.

This use case document describes the vEPC architecture, functional components and defines the resiliency requirements for the vEPC use case.

1.1 Operator Benefits of Virtualization

There are a number of Operator Benefits which can be achieved through virtualization of the EPC, these include:

- o Economies of scale through common virtualized platform
- o Enables a Multi-Service (MS) platform
- o Reducing time to market to offer new services
- o Uniformity of operations
- o Simplified high availability
- o Simplified disaster recovery
- o Preferred test and diagnostic tools embedded
- o Simplified in-service software upgrades
- o Reduced training
- o Simplified planning and provisioning
- o Automation of installation
- o Reduced site visits

2. Terminology

Evolved Packet Core (EPC): is an evolution of the 3GPP GPRS system

Home Subscriber Server (HSS): a database that contains user-related and subscriber-related information. It also provides support functions in mobility management, call and session setup, user authentication and access authorization.

Mobility Management Entity (MME) provides the signaling related to mobility and security for Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) access.

Packet Data Network Gateway (PDN GW): is the point of interconnect between the EPC and the external IP networks.

Policy and Charging Rules Function (PCRF): provides policy and service control and the appropriate interfaces towards the mobile charging and billing systems.

Serving GW (SGW): is the interconnect between the radio-side and the EPC. The SGW serves the User Equipment (UE) by routing the incoming and outgoing IP packets.

Virtualized Network Function (VNF): a VNF provides the same functional behavior and interfaces as the equivalent network function, but is deployed as software instances building on top of a virtualization layer.

VNF Pool: a group of VNF instances providing the same network function.

VNF Pool Element: a VNF instance inside a VNF pool.

VNF Pool Manager: an entity that manages a VNF pool, and interacts with the service control entity to provide the network function.

VNF Set: a group of VNF instances that can be used to build network services.

3. Virtual Evolved Packet Core (vEPC)

Deploying and operating mobile core network functions on commodity hardware resources may provide significant network usage efficiency and reductions in operational expenditure. Increased automation would also accommodate scaling of voice and mobile data demands.

The ETSI NFV use case [NFV-ISG-UC] describes requirements for

Internet Draft Virtualisation of Mobile Network January 2015
server and packet gateways used for Packet Data Network
(PDN) connections and IP Multimedia Subsystem (IMS) session (see
Figure 1: Virtualized mobile core network and IMS).

Typically mobile services are typically time dependent and may require a large number of computing resources in proportion to the number of users and/or service requests. Therefore it is desirable to scale them according to their specific computing requirements. The virtualization can be applied to the Evolved Packet Core (EPC) and the IMS to provide end to end service with service availability and resilience.

3.1 Mobile Core Network Components

Within the mobile core network a number of nodes and specific functions are currently provided by dedicated hardware and software for mobile voice and data services, these are described in more detail in the following sub-sections.

3.1.1 Mobile Network Nodes

The EPC is comprised of a variety of nodes, these include:

- o Mobility Management Entity (MME);
- o Serving Gateway (SGW);
- o Packet Data Network Gateway (PDN-GW);
- o Home Subscriber Server (HSS).

3.1.2 Mobile Network Functions

The EPC provides a number of functions to manage mobile user traffic, these include:

- o Firewall (FW);
- o Policy Control (PC);
- o Network Address Translation (NAT);
- o Load Balancing (LB);
- o Deep Packet Inspection (DPI);
- o TCP Optimization of Traffic Flows;
- o HTTP Enrichment of Traffic Flows;

- o Video Stream Optimization;

- o Video Content Caching.

3.2 vEPC Resiliency Requirements

When those virtualized service nodes(e.g., virtualized S/P-GW and IMS functions) are failed or overloaded, dynamic relocation of VNFs can be performed, the relocation of the managed sessions and/or connections must be accordingly managed. It also should be noted in [NFV-REL-REQ] that the traffic in the original VSN must be routed to the new location and it is desirable that the movement of the VSN is transparent to other VSN and or physical network entities such as client application on the UE. That is to say the other VSNs do not require to take any special action to this movement.

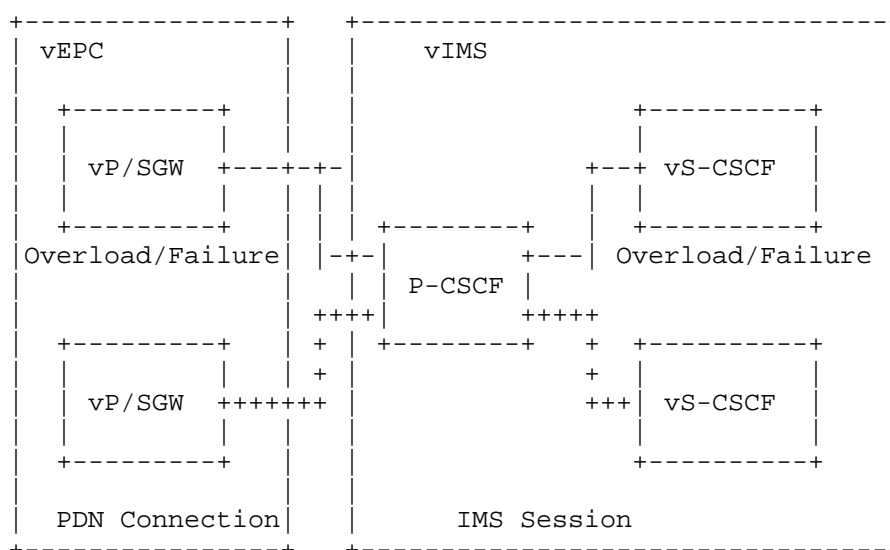


Figure 1: Virtualized Mobile Core Network and IMS

In this architecture, the following general resiliency requirements need to be satisfied:

- o Resource scaling - elastic service aware resource allocation to network functions;
- o State maintenance - network and network function state management during VSN relocation, replication, and resource scaling;
- o Monitoring/fault detection/diagnosis/recovery - appropriate

- o Service Availability - achieving the same level of service availability for the end-to-end virtualized mobile core network as in non-virtualized networks with reduced cost;
- o Minimum impact on other relevant functions.

3.2.1 Handling Unplanned Traffic Peaks

Vendors are currently working with the Japanese Government to demonstrate the capabilities that a vEPC can have in handling unplanned traffic surges due to unforeseen circumstances:

- o A recent earthquake in Japan caused the demand for calls to increase to 150% capacity in the effected area. Calls were dropped due to the network capacity.
- o At the time the capacity in other areas was only 50%. In a vEPC environment the free resources from the other areas could have been used to manage this additional load.

3.2.2 Scaling of Resources and Functions

The Evolved Packet System (EPS) is built from logical network functions, e.g. MME, PDN Gateway, Serving Gateway and Radio Base station (evolved NodeB) which are connected through the specified architectures references points. The 3GPP standard considers load balancing between different logical network functions of the same type. For example, Radio Base stations can choose one out of multiple available MMEs according to load-based weight factors to register an attaching mobile device. Mobile network operators can dimension their network in terms of numbers of required MMEs or data gateways according to statistical figures and thorough network planning, such as busy hour call attempts (BHCA).

Virtualization technology enables adding additional resources as logical network functions by means of instantiation of the relevant functions in virtual machines. The instantiation of additional virtualized PDN Gateways or MMEs requires the announcement of their availability to other network components of the EPS. New attachments can then be balanced and distributed between an increased number of available network functions. Such procedure for scale-out suits the adaptation of the EPS resources to an increasing demand with low time constraints, e.g. due to an expected increase in subscribers or traffic volume.

Unexpected increase in traffic or subscribers' attempt to request mobile service can result from scheduled events, e.g. festivals, or in particular after disaster events, such as an earthquake. The latter case in particular requires the mobile network to handle service requests and traffic from a huge amount of active mobile subscribers.

Communication services during disaster events are essential, not only to provide a communication platform for rescue workers, but also to allow private subscribers to communicate with relatives.

Such unexpected increase in active subscribers and traffic volume should not result in dropped connections, e.g. forced disconnects to offload existing subscriber states and traffic volume. It is preferable to scale-out resources internal to a single logical network function, e.g. an MME or a PDN Gateway. The advantage of such network function-internal resources scaling is the in-dependency of and transparency to external network functions and EPC protocols.

Functionality and resources for a particular Virtualized Network Function (VNF) may be provisioned by the interplay of multiple virtualized Network Function Components (VNFC), whose instances map 1:1, or m:1, to virtual machines. Scaling up internally of a single instance of a VNF may be accomplished by the instantiation of additional VNFC instances. Load on the VNF must then be balanced between the multiple VNFC instances (LB). Such scaling must remain transparent to external network entities and to other VNFs.

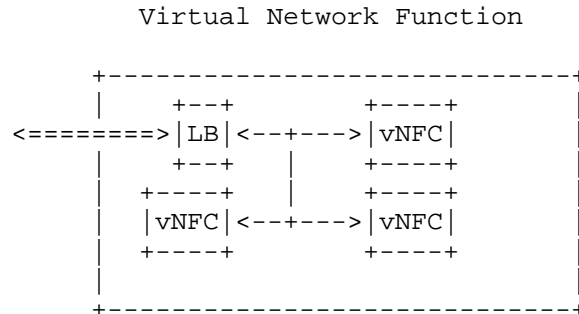


Figure 2: Composition of multiple VNFC instances to build a single VNF.

Technology for VNF scaling must also provide means to scale-in and reduce the number of resources in terms of required VNFCs, which provide the required network function.

Technology for VNF scaling must also provide means to scale-in and reduce the number of resources in terms of required VNFs, which provide the required network function.

Some general requirements for scaling in the view of virtualized EPC network functions:

- o Transparency and compatibility of network functions virtualization to legacy EPS components;
- o Support for scale-out of VNFs, representing additional logical EPC network functions;
- o Inter-working with configuration management (OSS) to configure and announce new Network Functions to the EPS;
- o Automation of scaling and simplified OAM;
- o VNF-internal scale-out and resiliency management;
- o Support of scale-in and associated shut down of VNFC instances; handling of states associated with VNFCs, which are to be shut down (state depletion vs. state transfer/offload);
- o (non-critical: VM aggregation to fewer host servers, e.g. to enable host server power saving).

Service requirements for the scaling of VNFs from VNFPool perspective, based on the current working group scope of work:

- o Balancing load between VNFs within a VNFPool;
- o Inter-working with system-wide (e.g. EPS) load balancing, e.g. cellular-specific selection of VNFs;
- o Compatibility with system-wide addressing of selected VNFs. VNFPool solutions may consider different addressing schemes and associated address mapping within and outside a VNFPool;
- o Coordination of scale-out and scale-in of VNFs within a VNFPool;
- o Coordination of the use, visibility and addressability of additional VNF resources. New VNFs, which carry a new system-wide identifier, need to be announced to the system. New VNFs, which carry only a new VNFPool-internal identifier and provide additional VNF resources for an existing instance of a network function (system is aware of the network function instance's identifier) require only VNFPool-internal coordination.

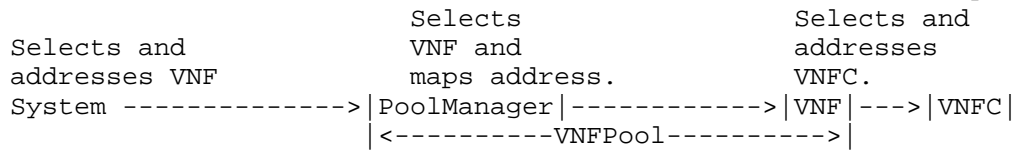


Figure: Scope of VNFPool and coordination between VNFPool-internal and system-wide selection, balancing and addressing of network functions

3.2.3 Failure Handling

During vEPC deployment, various failures can occur, for instance virtual machine failure, hypervisor failure, a broken host server, failure in a datacenter's transport network infrastructure, as well as failure of network links which connect a datacenter to the global network infrastructure.

It is unlikely that a single solution suits the handling of all kind of failures. Typically for today's products, function redundancy and state synchronization as well as failure detection and failover are function and implementation specific.

The detection of VM or hardware failures on a host server, as well as failure of networking equipment may introduce some delay before the system initiates failover to standby or backup resources. It may not be possible for an operator to meet agreed service levels in all cases.

Due to the variety of different failure reasons, detection of the failure type may be required to initiate the appropriate procedure for failover handling. Mobile operators have strong requirements to minimize the time of system outage as experienced by subscribers, hence require minimal detection and failover handling latencies.

Referring to the architecture of a virtualized Network Function as depicted in Figure 2, some VNFCs may require synchronization of states with a standby VNFC instance of the same kind to introduce redundancy on VNFC level. Others may not require state synchronization but rely simply a backup VNFC with the same functionality, as in case of failure, states can be recovered and retrieved from a different VNF, which holds the same or a sub-set of these states. Hence, redundancy management and failover mechanisms can be VNFC-specific.

Disaster events, such as an earthquake, can have impact to the availability of a larger VNF Set (a group of VNFs providing different functions) or even to the access to a complete data center in case the data center's links to the global network infrastructure

breaks. In such case, even the availability of a backup system in a globally and topologically distant data center can meet the requirement of service continuation. Seamless continuation of subscribers' services is unlikely, as it would require maintenance of state synchronization between functions being instantiated in different data centers. But solely the provisioning of backup vNFs allows subscribers to re-attach to the mobile communication system and place new calls. Handling such failover requires macroscopic indirection of the EPC reference points to a set of backup VNFs in a different data center.

Some general requirements for failure detection and failover handling in the view of virtualized EPC network functions:

- o Support function-specific redundancy and failover management;
- o Support different kinds of redundancy for failover (state synchronization between VNF instances, state recovery at backup VNF instances, state re-establishment at a backup VNF instance);
- o Selection of appropriate commodity hardware for backup and failover (resources availability);
- o Minimize state synchronization- and failover latency;
- o Detection of failure;
- o Detection of failure type and level (e.g. VNF, hypervisor, hardware, network);
- o Enforcement of failover strategy according to failure type;
- o Automated detection and failure handling.

Service requirements for failure handling from VNFPool perspective, based on the current working group scope of work:

- o Selection of suitable resources (host server, rack, topological location) for redundant VNFs;
- o Instantiation and installation of redundant resources on VNF-level;
- o Policing and enforcement of different redundancy schemes (e.g. active/standby synchronization, backup VNF);
- o Inter-working between VNF-internal (active/standby VNFC) and external (VNF redundancy) redundancy management;
- o Failover between VNFs within a VNFPool;

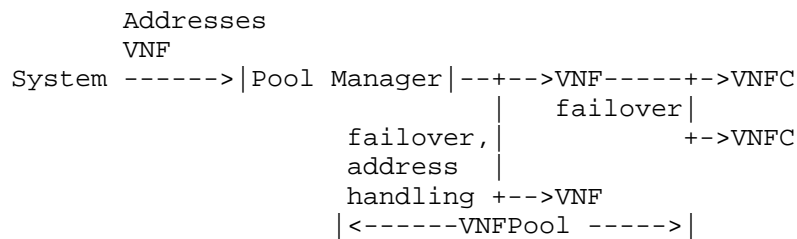


Figure: Scope of VNFPool and coordination between VNFPool-internal when handling failures.

3.2.4 State Synchronization

vEPC components may be split into control (signaling) and forwarding (data) plane traffic. A failure of a control plane traffic may result in the loss of communication between EPC functions. This should not impact user forwarding traffic, and it may be necessary for control functions to have state maintained and synchronized with back-up VNF instances hosting control elements.

Also it may be necessary for data plane state to also be synchronized so certain connections continue to be operational and capable of forwarding traffic during from one VNF to another.

3.3 What does that mean for Virtual Network Function Pool (VNF Pool)?

For VNF Pool in the view of EPC, it is to be investigated where an IETF-based generalized functional architecture and common protocol can support vEPC scaling, failure detection and handling. Such common protocol components should allow inter-working with VNF-specific and possibly proprietary but highly efficient mechanisms for redundancy and fault management.

The granularity of a VNF Pool Manager[zong-vnfpool-problem-statement] may be a VNF, VNF Pool or VNF Set. It is assumed that a Pool Manager handles VNFs with the granularity of EPC network functions (MME, PDN Gateway).

A VNF Pool Manager's role for load balancing between PEs is to be investigated, taking additional and independent load balancing instances for macroscopic (system-wide) load balancing within the EPS and for microscopic load balancing (between multiple VNFs of a single logical VNF instance) into account.

3.3.1 VNF Pool Definitions

There is a hierarchy of terms used to describe VNF Pool components and their relationship:

- o An instantiation of a VNF is known as a VNF instance;
- o A group of VNF instances is known as a VNF Set;
- o A managed VNF Set is known as a VNF Pool;
- o A VNF pool is managed using a VNF Pool Manager.

These definitions will be moved into the terminology section if they are agreed by the working group.

4. IANA Considerations

This document makes no IANA requests.

5. Security Considerations

[To be discussed.]

6. References

6.1. Normative References

6.2. Informative References

[3GPP-R8]

[NFV-ISG-UC]

"Network Function Virtualisation; Use Cases;", ISG NFV Use Case, June 2013.

[NFV-REL-REQ]

"Network Function Virtualisation Resiliency Requirements", ISG REL Requirements, June 2013.

[zong-vnfpool-problem-statement]

Zong, N., "Problem Statement for Reliable Virtualized Network Function (VNF) Pool", May 2014.

Authors' Addresses

Peter Willis
British Telecom
UK

Internet Draft Virtualisation of Mobile Network
Email: peter.j.willis@bt.com

January 2015

Daniel King
Lancaster University
UK

Email: d.king@lancaster.ac.uk

Jeong-dong Ryoo
ETRI

Email: ryoo@etri.re.kr

Marco Liebsch
NEC Laboratories Europe

Email: liebsch@neclab.eu

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 11, 2015

L. Xia
Q. Wu
Huawei
D. King
Lancaster University
H. Yokota
KDDI Lab
N. Khan
Verizon
November 11, 2014

Requirements and Use Cases for Virtual Network Functions
draft-xia-vnfpool-use-cases-02

Abstract

Network function appliances such as subscriber termination, firewalls, tunnel switching, intrusion detection, and routing are currently provided using dedicated network function hardware. As network function is migrated from dedicated hardware platforms into a virtualized environment, a set of use cases with application specific resilience requirements begin to emerge.

These use cases and requirements cover a broad range of capabilities and objectives, which will require detailed investigation and documentation in order to identify relevant architecture, protocol and procedure solutions to ensure reliance of user services using virtualized functions.

This document provides an analysis of the key reliability requirements for applications and functions that may be hosted within a virtualized environment. These NFV engineering requirements are based on a variety of uses cases and goals, which include reliability scalability, performance, operation and automation.

Note that this document is not intended to provide or recommend protocol solutions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Network Function Virtualization (NFV) Effort	4
1.2. Virtual Network Functions (VNF) Resilience Requirements .4	
1.2.1. Service Continuity	5
1.2.2. Topological Transparency	5
1.2.3. Load Balancing or Scaling	5
2. Terminology	5
3. Virtual Network Function (VNF) Pool Architecture.	7
3.1. VNF Instance Resilience Objectives	8
3.2. Resilience of Network Connectivity	8
3.3. Service Continuity	9
4. General Resilience Requirements For VNF Use Cases	9
4.1. Resilience for Stateful Service	9
4.1.1 State Synchronization	10
4.2. Auto Scale of Virtual Network Function Instances	11
4.3. Reliable Network Connectivity between Network Nodes . . .12	
4.4. Existing Operating Virtual Network Function Instance Replacement	13
4.5. Combining Different VNF Functions (a VNF Set)	14
4.6. VNF Resilience Classes	15
4.7. Multi-tier Network Service	15
5. IANA Considerations	17
6. Security Considerations	17
7. References	17
7.1. Normative References	17
7.2. Informative References	17
Authors' Addresses	17

Network virtualization technologies are finding increasing support among network and Data Center (DC) operators. This is due to demonstrable capital cost reduction and operational energy savings, simplification of service management, potential for increased network and service resiliency, network automation, and service and traffic elasticity.

Within traditional DC networks, varied middleware boxes including FW (Fire Wall), NAT (Network Address Translation), LB (Load Balancers), WoC (Wan Optimization Controller), etc., are being used to provide network functions, traffic control and optimization. Each function is an essential part of the entire operator and DC network, and overall service chain (required traffic path for users) Combined these functions and capabilities.

Currently, a significant amount of network functions are being migrated into virtualized entities, in essence the middleware capability is implemented in software on commodity hardware using well defined industry standard servers. Thus allowing the creation, modification, deletion, scaling, and migration of single or groups of network functions, across few or many servers.

These virtual network functions (VNF) may be location independent, i.e., they may exist across distributed or centralized DC hardware. This architecture will pose new issues and great challenges to the automated provisioning across the DC network, while maintaining high availability, fault-tolerant, load balancing, and plethora of other requirements some of which are technology and policy based.

Today, architecture and protocol mechanisms exist for the management and operation of server hardware supporting applications, these hardware resources are known as server node pools, which may be accessed by other servers and clients. These server node pools have a well-established set of requirements related to management, availability, scalability and performance.

[I-D.zong-vnfpool-problem-statement] provides an overview of the problems related to the reliability of a VNF set, and also introduces briefly a VNF pooling architecture. This document provides an analysis of the key reliability requirements for applications and functions that may be hosted within a virtualized environment. These Network Functions Virtualization (NFV) engineering requirements are based on a variety of uses cases and goals , which include reliability scalability, performance, operation and automation.

This document is not intended to provide or recommend solutions. The

Internet-Draft Requirements and Use Cases for VNF November 2014
intention of this document is to present an agreed set of objectives
and use cases providing network function using virtualized instances,
identification of key requirements across use cases.

1.1. Network Function Virtualization (NFV) Effort

NFV, an initiative started within the European Telecommunications Standards Institute (ETSI), aims to transform the way that network operators architect networks by evolving standard IT virtualization technology to consolidate many network equipment types to industry standard high volume servers, switches and storage.

The objectives for NFV being specified within the ETSI organization include:

- o Rapid service innovation through software-based deployment and operationalization of network functions and end-to-end services;
- o Improved operational efficiencies resulting from common automation and operating procedures;
- o Reduced power usage achieved by migrating workloads and powering down unused hardware;
- o Standardized and open interfaces between network functions and their management entities so that such decoupled network elements can be provided by different players;
- o Greater flexibility in assigning Virtual Network Functions (VNF) to hardware;
- o Improved capital efficiencies compared with dedicated hardware implementations.

1.2. Virtual Network Functions (VNF) Resilience Requirements

Deployment of NFV-based services will require the transition of resilient capabilities from physical network nodes, which are typically highly available, entities running Virtual Network Functions (VNFs) on abstracted pool of hardware resources.

Thus, it is critical to ensure that end-to-end user services which may require a variety of virtualized functions to be reliable, and in the event failure would support seamless failover when required to negate or minimize impact on user services.

A number of requirements have been discussed and documented within the NFV Industry Steering Group (ISG) working groups, including [ETSI-HA-USECASE] and are highlighted in following sub-sections.

VNFs provide the capability to execute and operate network functions on varying types of Virtual machines (VMs), and subsequently physical equipment. It should be possible to inherently provides resiliency at the function level, as well as physically.

Network Functions (NFs) are assigned session IDs, Sequence IDs and Authentication IDs. This information may be static, dynamic and temporal so will need to be replicated and maintained as needed for failure scenarios.

Hardware entity such as a storage server or networking node are assigned a unique MAC address, which is often pre-configured (hardware encoded) and static.

In the event of a hardware failure or capacity limits (memory and CPU) hosting VMs and therefore VNFs, it may be necessary to move VNFs to another VM, and/or hardware platform. Therefore, service continuity must be maintained with no or negligible impact to users using with services being provided by the NFs.

1.2.2. Topological Transparency

Redundant systems are typically configured as an active and standby nodes, running a specific NF in the same LAN segment. It is possible that they are assigned duplicate IP addresses, and sometimes the same MAC address as well. In the event of an active node failure the standby node can take over transparently. This should be architecture supported by any eventual solution.

In order to achieve topological transparency and seamless hand-over the dependent nodes should replicate and maintain the necessary information so that in the event of failure the standby node takes over the service without any disruption to the users.

1.2.3. Load Balancing or Scaling

When load-balancing or scaling of sessions, the working session may be moved to a new VNF instance, or indeed a new VM on another hardware platform. Again, service continuity must be maintained.

2. Terminology

The following terms have been defined by the ETSI Industry Steering Group (ISG) responsible for the specification of NFV, and are reused in this document:

Network Function (NF): A functional building block within a network infrastructure, which has well-defined external interfaces and a functional behavior. In practical terms, a Network Function is today often a network node or physical appliance.

NFV Orchestrator: The NFV Orchestrator is in charge of the network wide orchestration and management of NFV Infrastructure (NFVI) and resources. The NFV Orchestrator has control and visibility of all VNFs running inside the NFVI. The NFV Orchestrator provides GUI and external NFV-Interfaces to the outside world to interact with the orchestration software.

Service Continuity: The continuous delivery of service in conformance with service, functional and behavioral specification and SLA requirements, both in the control and data planes, for any initiated transaction or session till its full completion even in the events of intervening exceptions or anomalies, whether scheduled or unscheduled, malicious, intentional or unintentional. From an end-user perspective, service continuity implies continuation of ongoing communication sessions with multiple media traversing different network domains (access, aggregation, and core network) or different user equipment.

Hypervisor: Software running on a server that allows multiple VMs to run on the same physical server. The hypervisor manages and provide network connectivity to Virtual machines [RFC7365].

Network Functions Virtualization (NFV): Moving network function from dedicated hardware platforms onto industry standard high volume servers, switches and storage.

Set-top Box (STB): This device contains audio and video decoders and is intended to connects to a variety of home user devices media servers and televisions.

Virtual Machine (VM): Software abstraction of underlying hardware.

Virtual Application (VA): A Virtual Application is the more general term for a piece of software which can be loaded into a Virtual Machine. A VNSF is just one type of VA amongst many others, which may not relate to any VNF (e.g. SW-tools or NFV-Infra-internal applications).

Virtualized Network Function (VNF): a VNF provides the same functional behavior and interfaces as the equivalent network function, but is deployed as software instances building on top of a virtualization layer.

The VNF Problem statement [I-D.zong-vnfpool-problem-statement]

Internet-Draft Requirements and Use Cases for VNF November 2014
 defines the terms reliability, VNF, VNF Pool, VNF Pool
 Manager, and VNF Set. This draft also uses these definitions.
 In addition to the terms described above, this document also
 uses the following additional terminology:

VNF Pool: a group of VNF instances providing the same network
 function.

VNF Pool Manager: an entity that manages a VNF pool, and interacts
 with the service control entity to provide the network function.

VNF Set: a group of VNF instances that can be used to build network
 services.

3. Virtual Network Function (VNF) Pool Architecture

Shifting towards virtual network function presents a number of
 challenges and requirements, this document focuses on those
 related to network function availability and reliability. In large
 DC environments, a virtual server may need to deal with traffic
 from millions of hosts. This represents a significant scaling
 challenge for Virtual network function deployment and operation.

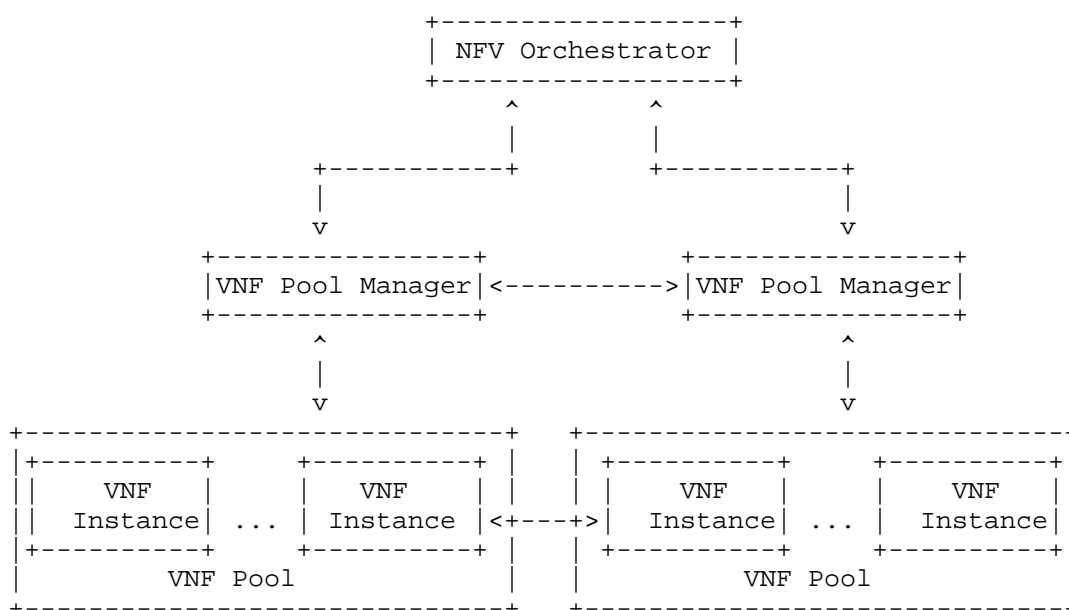


Figure 1: Typical VNF Pool Network Architecture

As shown in Figure 1, the overall architecture of VNF Pool-based
 network includes:

- o VNF Instances

- o VNF Pool

- o VNF Pool Manager

Rserpool [RFC5351] has the similar architecture to provide high-availability and load balancing, However Rserpool are only used to manage physical servers and can not deal with VNF instance when it was designed.

3.1. VNF Instance Resilience Objectives

In order to manage VNF-based nodes and provide fault tolerant and load sharing across nodes, the VNF instances may be initiated and established as logical element. A set of VNFs providing the same service type, is known as a VNF Pool, or groups of network functions (FW, LB, DPI) running on multiple VNFs, is known as a VNF Set.

Considering the reliability requirements of a VNF-based node architecture it should support several key points detailed below:

- o Application resource monitoring and health checking;
- o Automatic detection of application failure;
- o Failover to another VNF instance;
- o Transparency to other VNF instances;
- o Isolation and reporting of failures;
- o Replication of state for active/standby network functions.

3.2. Resilience of Network Connectivity

The other category of reliability requirements concerns the network connectivity between any two VNFs, across a VNF set, or between VNF Pool Manager.

The connectivity between the VNF Pool Manager and the VNF instance is used to provide registry service to the VNF Set. A set of VNF Pool managers might be configured to provide reliable registration.

When one VNF instance cannot obtain a register response from the assigned VNF Pool Manager, it should be capable of fail-over to

The connectivity between Pool Managers is used to maintain synchronization of data between VNFs located in different VNF Pools or VNF Sets. This allows every Pool Manager to acquire and maintain overall information of all VNFs and provide protection for each other.

For all types of network connectivity discussed previously, the key reliability requirements stay consistent and include:

- o Automatic detection of link failure;
- o Failover to another usable link;
- o Automated routing recovery.

3.3. Service Continuity

It is critical to ensure end-to-end service continuity over both physical and virtual infrastructure. A number of requests exist to maintain user services in the event of network or VNF instance failure, these include:

- o Storage and transfer of state information within the VNFs;
- o VNF capacity (memory and CPU) limitations per instance to avoid overbooking, and failure of end-to-end services;
- o Automated recovery of end-to-end services after failure situations;

4. General Resilience Requirements For VNF Use Cases

4.1. Resilience for Stateful Service

In the service continuity use case provided by the European Telecommunications Standards Institute (ETSI) Network Function Virtualization (NFV) Industry Specification Group (ISG) [NFV-REL-REQ], which describes virtual middlebox appliances providing layer-3 to layer-7 services may require maintaining stateful information, e.g., stateful vFW. In case of hardware failure or processing overload of VNF, in addition to the replacement of VNF, it is necessary to move its key status information to new VNF for service continuity. See Figure 2 (Resilience for Stateful Service) for clarification.

In case of multiple vFWs on one VM and not enough resources are available at the time of failure, two strategies can be taken: one is

Internet-Draft Requirements and Use Cases for VNF November 2014
to move as many vFws as possible to a new place according to the
available resources, and the other is to suspend one or more running
VNFs in the new place and move all vFws on the failed hardware to it.

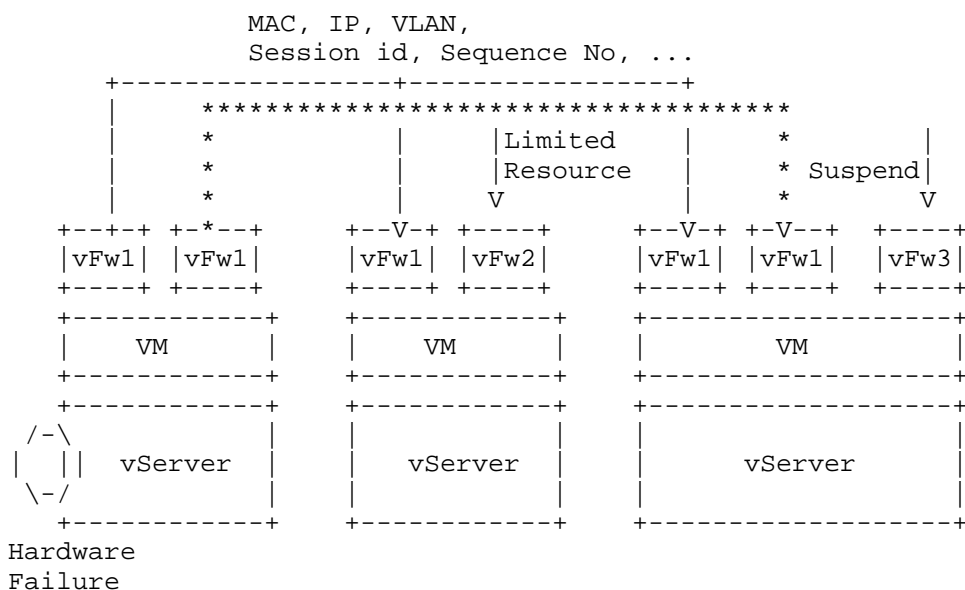


Figure 2: Resilience for Stateful Service

In both scenarios, the following requirements need to be satisfied:

- o Supporting status information maintaining;
- o Supporting status information moving;
- o Supporting VNF moving from one VM to another VM;
- o Supporting partial VNFs moving;
- o Seamless switching user traffic to alternative VMs and VNFs.

4.1.1.1 State Synchronization

As identified in section 4.1 (Resilience for Stateful Service) there is a requirement for for state synchronization. A failure of a vFW would result in the loss of active connections transiting the node. Any connection-orientated or secure sessions, including enterprise and financial transactions, may be critical, and losing them would result in the loss of data.

If required it should be possible to ensure that the VNF Pool infrastructure should minimise or negate session data traffic if a vFW failures. Prior to the failure the vFW might advertise and synch the connection information transitioning its node. The connection state synchronization to other vFWs acting as stand-by nodes would provide fast fail-over and minimal connection interruption to users.

This synchronization mechanism should be supported by the (NFV) infrastructure level, that is, ideally each application does not need to code the redundancy procedures (reserve a VM resource, instantiate one or more backup server(s), copy the state, keep them in sync, etc). Also, such a state can be embedded in each vNF or stored in an external virtual storage, which should be supported by the NFV infrastructure.

4.2. Auto Scale of Virtual Network Function Instances

Adjusting resource to achieve dynamic scaling of VMs described in the ETSI [NFV-INF-UC] use case and [NFV-REL-REQ]. As shown in Figure 3, if more service requests come to a VNF than one physical node can accommodate, processing overload occurs. In this case, the movement of the VNF instance to another physical node with the same resource constraints will create a similar overload situation. A more desirable approach is to replicate VNF instance to one or more new VNF instances and at the same time distribute the incoming requests to those VNF instances.

In a scenario where a particular VNF requires increased resource allocation to improve overall application performance, the network function might be distributed across multiple VMs. To guarantee performance improvement, the hypervisor dynamically adjusts (scaling up or scaling down) resources to each VNF in line with the current or predicted performance needs.

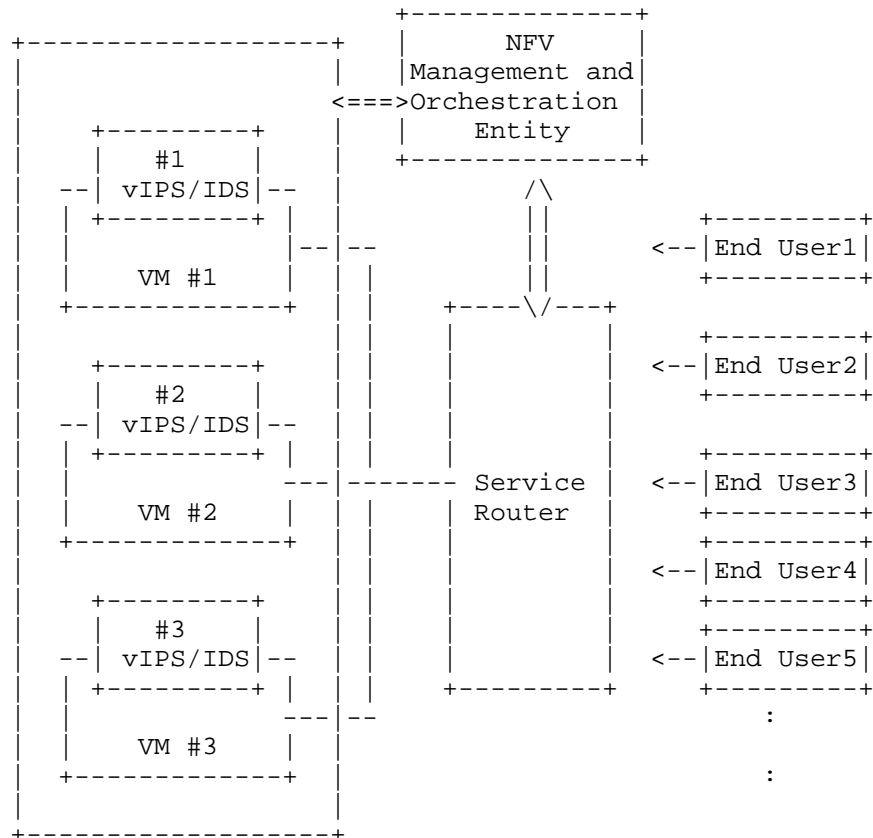


Figure 3: Auto Scaling of Virtual network Function Instances

In this case, the following requirements need to be satisfied:

- o Monitoring/fault detection/diagnosis/recovery - appropriate mechanism for monitoring/fault detection/diagnosis/recovery of all components and their states after virtualization, e.g. VNF, hardware, hypervisor;
- o Resource scaling - elastic service aware resource allocation to network functions.

4.3. Reliable Network Connectivity between Network Nodes

In the reliable network connectivity between VNFs use case provided by ETSI [NFV-INF-UC], the management and orchestration entities must be informed of changes in network connectivity resources between VNFs. For example, Some network

Internet-Draft Requirements and Use Cases for VNF November 2014

connectivity resources may be temporarily put in power savings mode when resources are not in use. This change is not desirable since it may have great impact on reachability and topology. Another example, some network connectivity resource may be temporarily in a fault state and comes back into an active state, however some other network connectivity resource becomes permanent in a fault state and is not available for use.

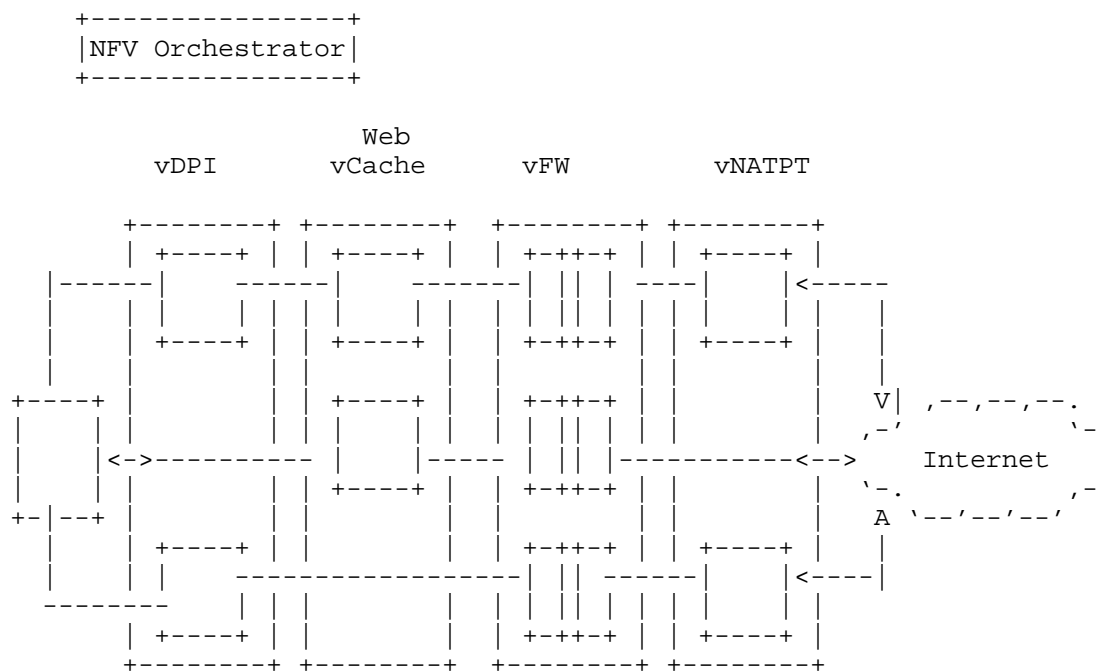


Figure 4: Reliable Network connectivity

In this case, the following requirements need to be satisfied:

- o Quick detection of link failures;
- o Adding or removing VNF instances;
- o Adding or removing network links between VNFs.

4.4. Existing Operating Virtual Network Function Instance Replacement

In the Replacement of existing operating VNF instance use case provided by ETSI [NFV-INF-UC] use case, the Management and Orchestration entity may be configured to support virtualized network function replacement. For example, the Network Service Provider has a virtual firewall that is operating. When the operating vFW

overloads or fails, the Management and Orchestration entity determines that this vFW instance needs to be replaced by another vFW instance.

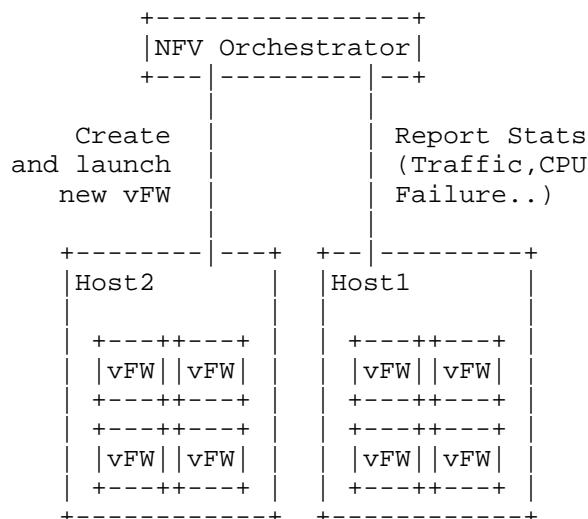


Figure 5: Existing vFW replacement

In this case, the following requirements need to be satisfied:

- o Verifying if capacity is available for a new instance of the VNF at some location;
- o Instantiating the new instance of a VNF at the location;
- o Transferring the traffic input and output connections from the old instance to the new instance. This may require transfer of state between the instances, and reconfiguration of redundancy mechanisms;
- o Pausing or deleting the old VNF instance.

4.5. Combining Different VNF Functions (a VNF Set)

A VNF Set is used to assemble a collection of network functions together to support a type of user or end-to-end service. Connectivity between the VNF sets is known as a VNF Forwarding Graph (a graph of logical links connecting VNFs together for steering traffic between network function). To support the reliability of an end-to-end service, except for satisfying the aforementioned basic use case requirements, a VNF Set presents further requirements of reliability as followed:

- o As a whole, any failures (i.e., VNF failures, link failures, performance degradation, etc) of a VNF Set can be detected and recovered in time;
- o Keeping the VNF order and relation unchanged when the VNF Set is updated;
- o The integrated VNF Set performance is not denigrated after it is updated;

4.6. VNF Resilience Classes

Different end-to-end services(e.g., Web, Video, financial backend, etc) have different classes of resilience requirement for the VNFs.

The use of class-based resiliency to achieve service resiliency SLAs, without "building to peak" is critical for operators.

VNF resilience classes can be specified by some attributes and metrics as followed:

- o Does the VNF need status synchronization;
- o Fault Detection and Restoration Time Objective (e.g., real-time, near-real time, non-realtime) and metrics;
- o Service availability metrics;
- o Service Quality metrics;
- o Service reliability;
- o Service Latency metrics for components.

[More description is needed.]

4.7. Multi-tier Network Service

Many network services require multiple network functions to be performed sequentially on data packets. A traditional model for multi-tier service is shown as below, where for each network function, all instances connect to the corresponding entrance point (e.g. LB) responsible for sending/receiving data packets to/from selected instance(s), and steering the data packets between different network functions.

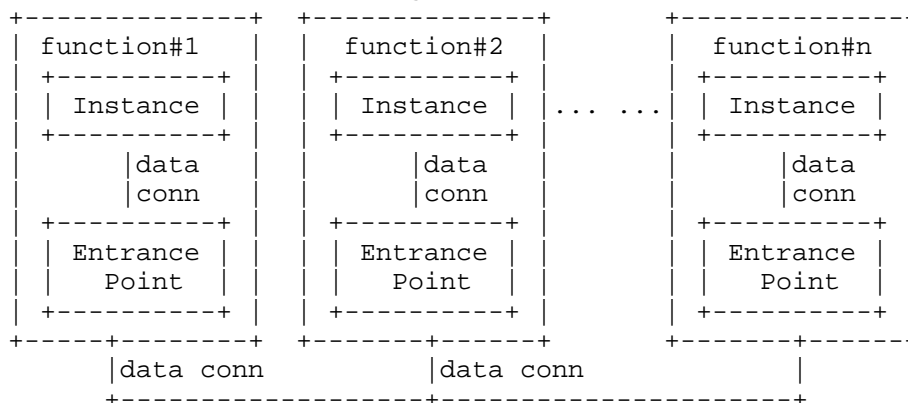


Figure 7: Multi-tier Service

Such model works well when all instances of the same network function are topologically close to each other. However, VNF instances are highly distributed in DC networks, Network Operator networks and even customer premises. When VNF instances are topologically far from each other, there could be many network links/nodes between them for transferring the data packets. For two different VNF instances, it is possible that they are on the same physical server, but the entrance points are many links/nodes away. To improve network efficiency, it is desirable to establish direct data connections between VNF instances, as shown below:

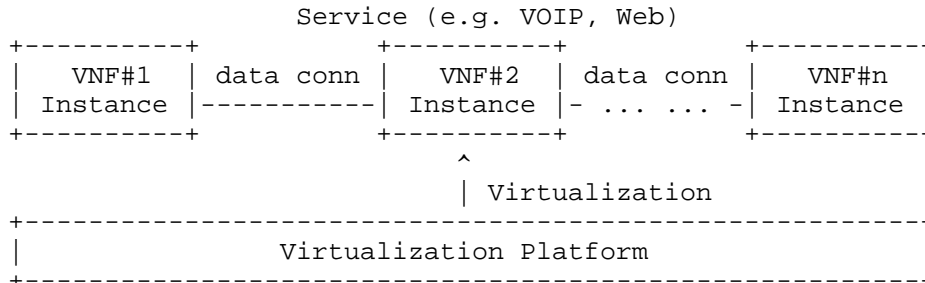


Figure 8: VNF Instances Direct Connection'

In this case, the following requirements need to be satisfied:

- o End to end failure detection of VNFs or links for multi-tier service;

- o Keep running service not be influenced during VNF instance transition or failure in the model of VNF instances direct connection.

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

TBD.

7. References

7.1. Normative References

7.2. Informative References

[NFV-INF-UC]

"Network Functions Virtualisation Infrastructure Architecture Part 2: Use Cases", ISG INF Use Case, June 2013.

[ETSI-HA-USECASE]

"Network Function Virtualisation; Use Cases;", ISG NFV Use Case, June 2013.

[NFV-REL-REQ]

"Network Function Virtualisation Resiliency Requirements", ISG REL Requirements, June 2013.

[I-D.zong-vnfpool-problem-statement]

Zong, N., "Problem Statement for Reliable Virtualized Network Function (VNF) Pool", July 2014.

[RFC7365]

Lasserre, M., et al. "Framework for DC Network Virtualization", RFC7365, October 2014.

[RFC5351]

Lei, P., Ong, L., Tuexen, M., and T. Dreibholz, "An Overview of Reliable Server Pooling Protocols", May 2008.

Authors' Addresses

Liang Xia(Frank)
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: frank.xialiang@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Daniel King
Lancaster University
UK

Email: d.king@lancaster.ac.uk

Hidetoshi Yokota
KDDI Lab
Japan

Email: yokota@kddilabs.jp

Naseem Khan
Verizon
USA

Email: naseem.a.khan@verizon.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 2, 2015

N. Zong
L. Dunbar
Huawei Technologies
M. Shore
No Mountain Software
D. Lopez
Telefonica
G. Karagiannis
University of Twente
July 1, 2014

Virtualized Network Function (VNF) Pool Problem Statement
draft-zong-vnfpool-problem-statement-06

Abstract

Network functions are traditionally implemented on specialized hardware rather than on general purpose servers, but there is a clear trend to implement a number of network functions, such as firewall or load balancer, as software on virtualized computing platforms. These virtualized functions are called Virtualized Network Functions (VNFs), which can be used to build network services. The use of VNFs to build network services introduces additional challenges on reliability, such as additional points of failure and the need to coordinate various VNFs.

This document introduces a general idea of VNF Pool to support reliable function provision by the VNFs. We then highlight the reliability challenges and issues when using the VNFs to build services. Related IETF works are also briefly described.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 2, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	4
3. Background	4
3.1. From Specialized Hardware to Virtualized Network Function	4
3.2. Concept of VNF Set	5
3.3. Challenges to reliability	6
4. VNF Pool	6
5. Challenges and Open Issues	8
5.1. Redundancy model inside VNF	8
5.2. State synchronization inside VNF	8
5.3. Interaction between VNF and Service Control Entity	8
5.4. Reliable transport	9
5.5. Scope Considerations	9
6. Related Works	9
6.1. Reliable Server Pooling (RSerPool)	9
6.2. Virtual Router Redundancy Protocol (VRRP)	10
6.3. Service Function Chaining (SFC)	10
7. Security Considerations	10
8. IANA Considerations	11
9. Acknowledgements	11
10. References	11
10.1. Normative References	11
10.2. Informative References	11
Authors' Addresses	12

1. Introduction

Network functions such as firewall, load balancer, WAN optimizer are conventionally deployed as specialized hardware servers in both network operators' networks and data center networks, as the building blocks of the network services.

A Virtualized Network Function (VNF) provides such network function through its implementation as software instances running on general purpose servers via a virtualization layer (i.e., hypervisor). VNFs potentially offer benefits such as elastic service offering, reduced operational and equipment costs [NFV-WP].

There is a trend to move network functions from specialized hardware servers to general purpose servers based on virtualized computing platforms, in order to build network services by using VNFs. For example, in Service Function Chaining (SFC), a network service can be built using a set of sequentially connected VNF instances deployed at different points in the network [SFC].

Nevertheless, the use of VNFs can pose additional challenges on the reliability of the provided services. For a VNF instance, it typically would not have built-in reliability mechanisms on its host (i.e., a general purpose server). Instead, there are more factors of risk such as software failure at various levels including hypervisors and virtual machines, hardware failure, and instance migration that may make a VNF instance unreliable.

In order to achieve higher reliability, a VNF may adopt a pooling mechanism, where a number of VNF instances with the same function can be grouped as a pool to provide the function. We call such a pool a VNF Pool. Conceptually, a Pool Manager is used to manage a VNF Pool, e.g., selects active/standby VNF instances, and potentially interacts with a Service Control Entity. A Service Control Entity is an entity that combines and orchestrates a set of network functions, e.g., VNFs, to build network services. The major benefit of using VNF Pool is that the reliability mechanisms such as redundancy management are achieved by the VNF Pool inside the VNF and thus transparent to the Service Control Entity. A VNF Pool-enabled VNF still acts as a normal VNF when orchestrated by the Service Control Entity.

We are specifically concerned with the reliability of an individual VNF based on the VNF Pool managed inside the VNF. For example, how to manage the redundancy model, e.g., select active/standby for a VNF instance in a VNF Pool, considering the policy and the infrastructure conditions? How are the service states of a VNF instance held and accessed for efficient synchronization with backup instances in a VNF Pool? What pool states need to be maintained to support the pooling mechanism itself, and how are such states maintained? We also consider the information exchanged between the VNF and Service Control Entity. For example, how can a VNF Pool be addressed by the Service Control Entity? After a VNF instance failover, how does the Pool Manager notify the Service Control Entity of some characteristic changes of the VNF, e.g., capacity change, but without disclosure of the pooling procedure?

Note that we do not address the reliability related control or routing between adjacent VNFs that can form a network service, as such coordination could be done by the Service Control Entity.

This document introduces a general idea of VNF Pool to support reliable functions provision by the VNFs. We then highlight the reliability challenges and issues when using the VNFs to build services. Related IETF works are also briefly described.

2. Terminology

Reliability: capability of a functional entity to consistently provide its function under various dynamic and even unexpected conditions such as fault, overload, etc.

Service Control Entity: an entity of the service provider that decides how to combine and orchestrate the network functions to build network services. Examples of Service Control Entity are orchestrator of DC services, SFC control plane, etc.

Virtualized Network Function (VNF): a VNF provides the same functional behavior and interfaces as the equivalent network function, but is deployed as software instance(s) building on top of a virtualization layer [NFV-TERM].

VNF Pool: a number of VNF instances providing the same network function.

VNF Pool Element: a VNF instance inside a VNF pool.

VNF Pool Manager: an entity that manages a VNF pool, and interacts with the Service Control Entity to provide the network function.

VNF Set: a general set of VNF instances that can be grouped into multiple VNF Pools, where each pool corresponds to a specific VNF and different pools provide different functions.

3. Background

3.1. From Specialized Hardware to Virtualized Network Function

Network functions are traditionally implemented on specialized hardware. There is a trend to implement a number of network functions as software instances on general purpose servers, via virtualized computing platforms. These virtualized functions are called Virtualized Network Functions (VNFs). For example, in Figure 1, virtual firewall (vFW) can be deployed as software instances on general purpose servers, which could be located in Data

Center (DC) networks, network operators' networks, or end user premises. Compared with traditional FW deployed as "standalone box" built by specialized hardware and software, vFW has potential advantages such as agility, scalability [NFV-WP].

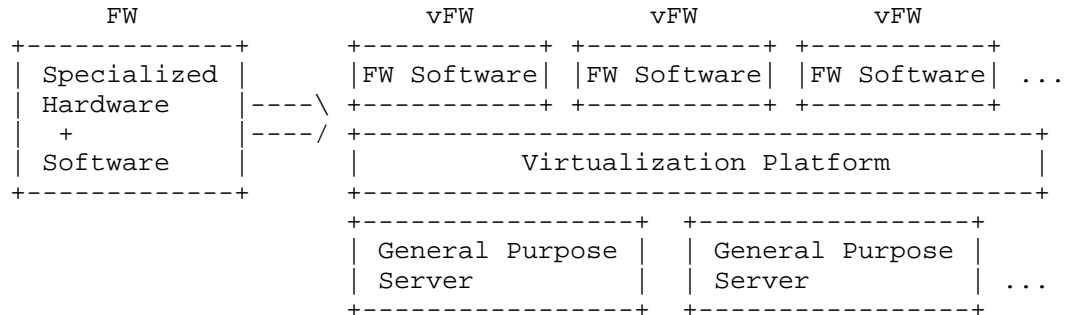


Figure 1: Example of vFW.

3.2. Concept of VNF Set

We call a general set of VNF instances a VNF set. A VNF set can include a single or multiple types of VNF, and each type of VNF may have a number of instances providing the same function. The following examples are all valid VNF sets.

1. n vFW instances: {vFW#1, vFW#2, ..., vFW#n}.
2. m vFW instances and k virtual load balancer (vLB) instances: {vFW#1, ..., vFW#m, vLB#1, ..., vLB#k}.

To be more generic, we denote VNF-A#x the xth instance of a VNF of type A (e.g., vFW), VNF-B#y the yth instance of a VNF of type B (e.g., vLB), and so on.

A VNF set can be used as part of a Service Function Chaining (SFC) [SFC], where the instances of various functions are sequentially connected to build a network service. A simple example is shown in Figure 2.

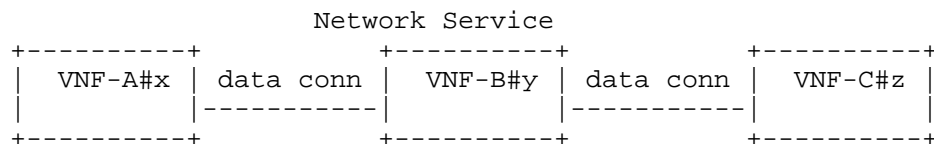


Figure 2: A VNF set used as part of a SFC.

Alternatively, a VNF set can be also used merely as a set of VNFs, where the instances provide network functions in a parallel way. An example is shown in Figure 3.

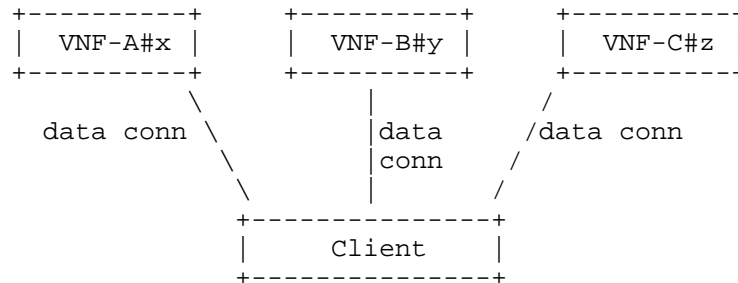


Figure 3: A VNF set used as multiple VNFs.

Some more detailed use cases of VNFs are documented in other drafts [VNFPOOL-UC1] [VNFPOOL-UC2] [VNFPOOL-UC3].

3.3. Challenges to reliability

The use of VNFs introduces additional challenges to the reliability of the provided network services. For a VNF instance, it typically would not have built-in reliability mechanisms on its host (i.e., a general purpose server). Instead, there are more factors of risk that may make VNF instance unreliable.

1. Instance failure due to hardware failure or status change such as server overload.
2. Instance failure due to software failure at various levels including hypervisor, Virtual Machine (VM), VNF.
3. Instance migration caused by instance performance downgrade caused by load (e.g., CPU, memory, disk I/O), server consolidation or other service requirement changes. This is distinct from a hard failure, although it may give the appearance of one.

4. VNF Pool

There are a number of existing technologies for providing reliable functions, such as Reliable Server Pooling (RSerPool) [RFC5351], Virtual Router Redundancy Protocol (VRRP) [RFC5798], amongst many others. Both technologies provide the service with an abstract object (e.g., pool handle in RSerPool, virtual router ID in VRRP) representing a group of identical functional instances. The dynamic mapping of such abstract object to the actual serving instance is

managed internally in the group to cover the failover procedure. The advantage is to provide reliable functions in a transparent manner for both end-hosts and service control entities.

We adopt the similar idea of VNF Pool to provide reliable network functions, as shown in figure 4.

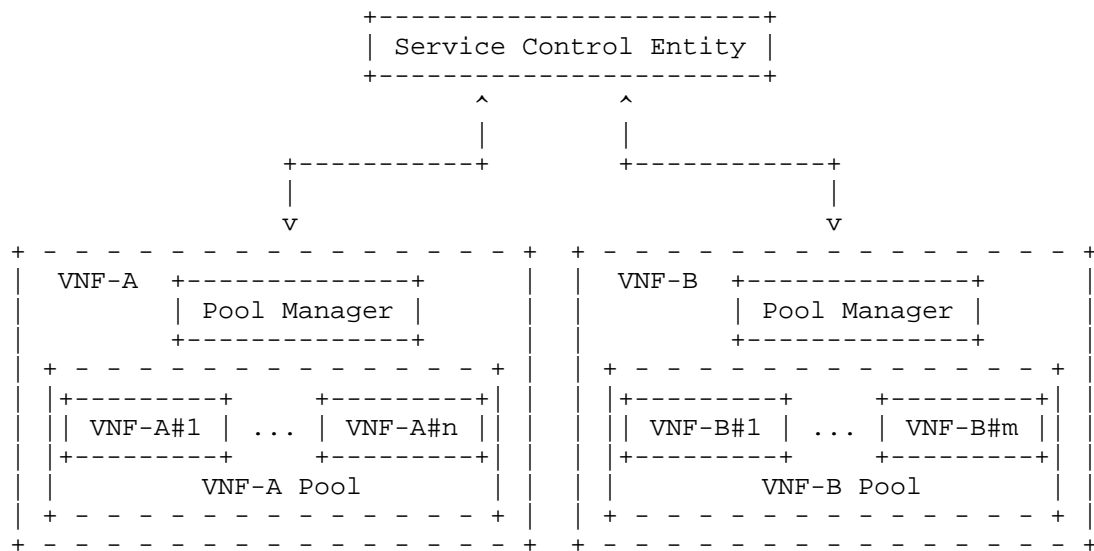


Figure 4: VNF Pool Architecture.

In VNF Pool architecture, each VNF has a VNF Pool containing a number of VNF instances (or VNF Pool Elements) providing the same function. In this sense, a VNF set can be grouped into multiple VNF Pools, where each pool corresponds to a specific VNF, thus different pools provide different functions. Each VNF also has a Pool Manager that manages the VNF instances in the VNF Pool. Pool Manager interacts with the Service Control Entity to provide the network function.

The main benefit of using VNF Pool is that the pooling mechanisms such as redundancy management are achieved by the VNF Pool inside the VNF and thus transparent to the Service Control Entity. The Service Control Entity simply interacts with the Pool Manager in each VNF to request and orchestrate the network functions with desired reliability level. In another word, a VNF Pool-enabled VNF still acts as a normal VNF when orchestrated by the Service Control Entity.

5. Challenges and Open Issues

5.1. Redundancy model inside VNF

Before a live VNF instance fails, one or more backup instances in the same VNF Pool need to be selected. How to select such backup instances? Moreover, there are policies influencing the appropriate selection of backup instance. For example, it should be avoided that a live VNF instance and its backup instances are placed in a single physical server, or locations with shared risks in the network. On the other hand, it would be desirable to place the live and backup instances in geographically closed locations. Information from the underlying network may need to be collected via - e.g., the interface with Application Layer Traffic Optimization (ALTO) [ALTO], or Interface to Routing System (I2RS) [I2RS]. Various infrastructure conditions may also need to be considered for appropriate placement of instances.

5.2. State synchronization inside VNF

Service states related to the specific function performed by a VNF instance, e.g., NAT translation table, TCP connection states, should be synchronized between a live VNF instance and its backup instances for stateful failover. Who is responsible for and how to collect, hold, and access such service states to achieve efficient synchronization? A VNF instance should provide negotiated level of state sharing with the necessary performance to fulfill the service requirements - e.g., state synchronization method, format of state data, location and mechanism to access state data.

Other than service states, pool states could be operational information of VNF pool itself, e.g. redundancy settings, backup location/status, etc. What pool states need to be maintained to support the pooling mechanism itself, and how are such states maintained?

5.3. Interaction between VNF and Service Control Entity

Some information needs to be exchanged between a VNF and the Service Control Entity when the Service Control Entity orchestrates a VNF Pool-enable VNF. For example, how can a VNF Pool be addressed by the Service Control Entity? A Pool Manager can advertise the locator (e.g., IP address) of the active instance - subject to dynamic due to failover. It is also possible to use a virtual address for the whole VNF Pool (similar to RSerPool or VRRP), and map between virtual and actual addresses. Moreover, after a VNF instance failover, how does the Pool Manager notify the Service Control Entity of some

characteristic changes of the VNF, e.g., capacity change, but without disclosure of the pooling procedure?

5.4. Reliable transport

The transport mechanism used to carry the pool control messages, e.g., redundancy management, should provide reliable message delivery. Transport redundancy mechanisms such as Multipath TCP (MPTCP) [MPTCP] and the Stream Control Transmission Protocol (SCTP) [RFC3286] will need to be evaluated for applicability. Latency requirements for pool control message delivery must also be evaluated.

5.5. Scope Considerations

Ideally, the reliability goal is that the network service provided by the VNFs will continue throughout an interruption within the VNFs , and VNF instances failure or migration will not be visible to the external entities. Our work of VNF Pool initially focuses on several reliability mechanisms that are mainly associated with a redundancy model based on a VNF Pool. Additional mechanisms may include pool state maintenance only for pooling purpose. Service state synchronization is out of scope for this phase.

We currently assume that a VNF Pool contains the instances of same functional type, e.g., FW, LB, etc. Different types of VNFs are envisioned to be held in separate VNF Pools. VNF Pool composed of both virtualized and non-virtualized functional instances may be included after further use case and requirements study.

We are specifically concerned with the reliability of an individual VNF based on the VNF Pool managed inside the VNF. We do not address the reliability related control or routing between adjacent VNFs that can form a network service, as such coordination could be done by the Service Control Entity.

We do not intend to resolve the service availability that usually involves more factors including the interruptions in various OSI layers, and even user perception on service performance.

6. Related Works

6.1. Reliable Server Pooling (RSerPool)

RSerPool supports high availability and scalability of the applications through the use of pools of servers [RFC5351]. The main functions of RSerPool involve server pool management, as well as receiving requests from a client to bind to a desired server. The

applicability and gaps of RSerPool to our work of VNF Pool are described in another draft [VNFPOOL-RSP].

6.2. Virtual Router Redundancy Protocol (VRRP)

VRRP specifies an election protocol that dynamically assigns responsibility of a virtual router to one of the VRRP routers called master on a LAN [RFC5798]. The election process provides dynamic failover in the forwarding responsibility should the Master become unavailable. The advantage of VRRP is a higher availability default path without requiring configuration of dynamic routing or router discovery protocols on every end-host.

6.3. Service Function Chaining (SFC)

A service chain defines an ordered set of service functions that must be applied to packets [SFC]. Although the VNFs can be used as part of a SFC, SFC and our work of VNF Pool have different focus.

As mentioned in the section of scope consideration, we mostly consider the reliability of an individual VNF based on the VNF Pool inside the VNF. We do not address the reliability related control or routing between adjacent VNFs in the forwarding graph. Moreover, according to VNF Pool architecture and principles, the VNF Pools will be orthogonal to and invisible to the SFC. A VNF Pool-enabled VNF still acts as a normal VNF when orchestrated by the SFC. Just like the communication between any pool users and VNF Pool, the information exchanged between the VNF Pool and the SFC may include some operational information of the VNF Pool.

7. Security Considerations

Any technology which allows the insertion, deletion, reordering, or manipulation of network functions has the potential to be subverted by an attacker, with serious consequences. Distributed VNFs introduce an additional attack vector, in which bad actors join several VNFs of a service. Replay attacks have the potential to create denials of service, reordering, adding, or removing VNFs. VNF reliability technologies must provide cryptographic protections against spoofing and insertion attacks as well as replay attacks, in the form of client authentication, origin authentication on VNF reliability management (control plane) traffic, and replay protections. There may be circumstances under which an attacker masquerading as a VNF manager can introduce data leakage or similar attacks, and consequently server authentication would be required, as well.

Failing over a VNF or otherwise transferring service state raises issues related to the transfer of security state, including VNF element identity and credentials, session-associated cryptographic state, and so on. Where possible, transfer of security state should be avoided as a matter of good practice, and this will require particular attention as solutions are drafted.

8. IANA Considerations

This document has no actions for IANA.

9. Acknowledgements

The authors would like to thank Chidung Lac from Orange, Daniel King from Lancaster University, Lingli Deng, Zhen Cao from China Mobile, Richard Yang from Yale University, Hidetoshi Yokota from KDDI, Mukhtiar Shaikh from Brocade, Qiang Zu from Ericsson, Marco Liebsch from NEC, Kapil Sood from Intel, Adrian Farrel, and Susan Hares for their valuable comments.

10. References

10.1. Normative References

TBD.

10.2. Informative References

[NFV-WP] NFV Whitepaper: "Network Function Virtualization", issue 1, 2012, http://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[SFC] "Service Function Chaining (SFC)",
<<http://datatracker.ietf.org/wg/sfc/>>.

[NFV-TERM] ETSI GS NFV 003: "Terminology for Main Conceptional Entities in NFV", Version 0.0.4, 2013.

[VNFPOOL-UC1] L. Xia, Q. Wu, D. King, H. Yokota, and N. Khan, "Requirements and Use Cases for Virtual Network Functions", draft-xia-vnfpool-use-cases-00, February 2014.

[VNFPOOL-UC2] D. King, M. Liebsch, P. Willis and J. Ryoo, "Virtualization of Mobile Core Network Use Case", draft-king-vnfpool-mobile-use-case-00, February 2014.

[VNFPOOL-UC3] S. Hares and K. Subramaniam, "Use Cases for Resource Pools with Virtual Network Functions (VNFs)", draft-hares-vnf-pool-use-case-00, January 2014.

[ALTO] "Application-Layer Traffic Optimization (alto)",
<<http://datatracker.ietf.org/wg/alto/>>.

[I2RS] "Interface to the Routing System (i2rs)",
<<http://datatracker.ietf.org/wg/i2rs/>>.

[MPTCP] "Multipath TCP (mptcp)", <<http://datatracker.ietf.org/wg/mptcp/>>.

[RFC3286] L. Ong and J. Yoakum, "An Introduction to the Stream Control Transmission Protocol (SCTP)", RFC3286, May 2002.

[NFV-REL] ETSI GS NFV REL 001: "Network Function Virtualization; Resiliency Requirements", Version 0.0.7, 2014.

[NFV-SWA] ETSI GS NFV SWA 001: "Network Function Virtualization; SW Architecture; Virtual Network Functions Architecture", Version 0.1.0, 2014.

[RFC5351] P. Lei, L. Ong, M. Tuexen and T. Dreibholz, "An Overview of Reliable Server Pooling Protocols", RFC5351, September 2008.

[RFC5798] S. Nadas, "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC5798, March 2010.

[VNFPOOL-RSP] T. Dreibholz, M. Tuexen, M. Shore and N. Zong, "The Applicability of Reliable Server Pooling (RSerPool) for Virtual Network Function Resource Pooling (VNFPOOL)", draft-dreibholz-vnfpool-rserpool-applic-00, October 2013.

Authors' Addresses

Ning Zong
Huawei Technologies

Email: zongning@huawei.com

Linda Dunbar
Huawei Technologies

Email: linda.dunbar@huawei.com

Melinda Shore
No Mountain Software

Email: melinda.shore@nomountain.net

Diego Lopez
Telefonica

Email: diego@tid.es

Georgios Karagiannis
University of Twente

Email: g.karagiannis@utwente.nl