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

Copyright Notice

Table of Contents

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].

```
       FW                   vFW           vFW           vFW
 +-------------+      +-----------+ +-----------+ +-----------+
 | Specialized |      |FW Software| |FW Software| |FW Software| ...
 | Hardware    |----\ +-----------+ +-----------+ +-----------+
 |  +          |----/ +-----------------------------------------+
 | Software    |      |          Virtualization Platform        |
 +-------------+      +-----------------------------------------+
                      +-----------------+  +-----------------+
                      | General Purpose |  | General Purpose |
                      | Server          |  | Server          | ...
                      +-----------------+  +-----------------+
```

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.

```
                       Network Service
 +----------+           +----------+           +----------+
 |  VNF-A#x | data conn |  VNF-B#y | data conn |  VNF-C#z |
 |          |-----------|          |-----------|          |
 +----------+           +----------+           +----------+
```

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.

```
             +----------+     +----------+     +----------+
             |  VNF-A#x |     |  VNF-B#y |     |  VNF-C#z |
             +----------+     +----------+     +----------+
                        \           |          /
               data conn \          |data     /data conn
                          \         |conn    /
                           \        |       /
                           +----------------+
                           |     Client     |
                           +----------------+
```

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.

```
                      +-----------------------+
                      | Service Control Entity |
                      +-----------------------+
                             ^        ^
                             |        |
                 +-----------+        +------------+
                 |                                 |
                 v                                 v
+ - - - - - - - - - - - - - - - +  + - - - - - - - - - - - - - - - +
|  VNF-A  +--------------+      |  |  VNF-B  +--------------+      |
|         | Pool Manager |      |  |         | Pool Manager |      |
|         +--------------+      |  |         +--------------+      |
| + - - - - - - - - - - - - - + |  | + - - - - - - - - - - - - - + |
| |+---------+     +---------+| |  | |+---------+     +---------+| |
| || VNF-A#1 | ... | VNF-A#n || |  | || VNF-B#1 | ... | VNF-B#m || |
| |+---------+     +---------+| |  | |+---------+     +---------+| |
| |         VNF-A Pool        | |  | |         VNF-B Pool        | |
| + - - - - - - - - - - - - - + |  | + - - - - - - - - - - - - - + |
+ - - - - - - - - - - - - - - - +  + - - - - - - - - - - - - - - - +
```

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