OPSAWG
Internet-Draft
Intended status: Informational
Expires: March 10, 2014

H. Song
Huawei

September 06, 2013

The Problems of Virtual Network Function Configuration
draft-song-opsawg-virtual-network-function-config-00

Abstract

This document describes the problem space of remote service installation and configuration in the provider's network through a centralized management system. This is a typical scenario for virtual function installation and dynamic configuration in network function virtualization (NFV) context. It is also a typical scenario in cloud computing environment where end users do not have to install applications in their end hosts, but can install their own featured powerful software application in the cloud. This specification also identifies the scope that needs standardization based on the problems.

Status of This Memo

Copyright Notice

Table of Contents

1. Background 2
2. Terminology 4
3. Problems of Service Configuration 5
4. Scope for Standardization 6
5. Security Considerations 7
6. References 7
  6.1. Normative References 7
  6.2. Informative References 7
Author's Address 7

1. Background

This document describes the problems in the context of remote software installation and service configuration through a central management system, called a controller. Four main roles are involved, including the user, the software provider, the controller and the infrastructure resources.

Users always have different requirements when they need a software. So a software vendor often provides a "full-set" of functions to satisfy a majority of users in the market. But for each individual user it might only need a few sub-set functions according to his own requirements. For example, a firewall could provide many functions, including anti-DDoS attack, anti-phishing attack, IP filtering function, MAC filtering function, network address translator function, and etc.. But a home user who is going to install a virtual firewall from the network operator, (which may be installed in a virtual machine inside the provider's network and the operator can make the traffic from/to this user go through that virtual machine,) may contempt his own environment and determines that he only needs anti-phishing attack function and MAC filtering function for his traffic. Other functions may be not useful for this user, for example, DDoS attacks may not happen to this user in this context. Typically, these functions exist in the software as different components. There are several possibilities for the user to acquire the relative components that he needs:

(1) A software vendor distinguishes users as several classes, and provides related versions of software to the users accordingly, for example, a "home edition" version, an "enterprise edition"

version and etc. In this case, the specific version may satisfy most users in that class, but for each individual user, it may also contain many function components that the user does not need.

(2) When a user requests a software, the user negotiates with a customer service person from the software vendor about his requirements, and the software vendor makes a specified version of software to the user, in this version, it enables the components that the user need, and disables those unneeded. In this case, it costs more human energy, and is not efficient. The user has to wait days or even longer for his specific software version after the negotiation.

(3) The user get a license and software packet, and with the license, it allows the user to choose inside a range of components for installation. The user enables components that he wants in that range. In this case, it gives the user more flexibility to operate the software components, but from another perspective, it also authorizes the user with more components than the user wants.

These methods either too complex, or authorize the user with more components than what he wants. A real-time, exact matching and flexible way for the user to choose his software components is desirable.

In the context of network function virtualizaition (NFV), more and more network functions become to be available in a virtualized function way. It adopts the common IT infrastructure instead of physical hardware box to implement these network functions. The benefits of this method is to reduce cost through improved infrastructure reusability and lower entry of the industry, which allows more software vendors. Various virtual functions exist in the network. They are deployed into virtual machines through the NFV controller. These virtual functions can be replaced with new virtual functions when needed, with only re-configuring it with a new software through the controller. In this case, NFV controller is just like a broker for many software applications.

The user may also have his own requirements or want to put his constraints on the network properties of the software that is to be installed in the provider’s network. For example, the bandwidth requirements for this software. This is different from the virtual machine level or host level bandwidth limitation. It is an application specific bandwidth limitation. A number of software applications can be installed on a same virtual machine, and they share the bandwidth of this virtual machine. But they are different applications and have different requirements on the bandwidth. In this case, the user specifies his requirements on the bandwidth of this software, and the NFV controller will enforce that constraints to the application level through some kind of configuration.

Besides the network properties, during the remote installation, the user would also need to notify the controller about the virtual network function’s storage space requirements, memory requirements, CPU requirements, operating system requirements, location constraints for the software installation. These constraints can be mapped to a virtual machine if a virtual network function is mapped to a single virtual machine, or a subset resources of a virtual machine if multiple virtual network functions share a virtual machine. It is the controller’s responsibility to select the most appropriate hardware resources for the virtual network function based on some mapping algorithm, but it is totally an implementation issue. These constraints are also relative to the software vendor, and the software vendor should describe the basic hardware and software requirements of the installation environment to the user, and the user should combine it with capacity requirements from himself and make a final decision on these parameters.

The previous two paragraphs describe the explicit way between the user and the controller for resource requirements. But there could also be an implicit way. In the implicit way, the user only describes what software components he needs. But the controller will choose appropriate resources for the user. When the user needs more capacity due to the service expansion, for example, in a scenario where an enterprise has a virtual firewall in the provider’s network, the controller is responsible for the redirection of the traffic from or to this enterprise go through the virtual firewall. The configuration could be that the controller sends flow rules to the network equipment, so as to forward the related traffic to the virtual firewall. And the virtual firewall notifies the controller about its traffic load status. If the load is above some threshold, the controller will automatically create new virtual firewall instances, and allocate additional CPU/memory/storage/bandwidth resources to handle that traffic. The controller will configure new flow rules to make a portion of the traffic go through the new instances. If the traffic volume is shrinking, the controller will automatically reduce the number of virtual instances for the user. The resource requirements for each virtual instance can be from the recommendation of the software provider. This kind of automatic scale-out and scale-in mechanism can make a better utilization of the network, computing and storage resources. And users do not need to have a deep understanding of the resource consumption model, but only pay as much as the resources he used.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. And the following terms used in this document have their definitions from the NFV end to end architecture [NFVE2E].

NFV: network function virtualization. NFV technology uses the commodity servers to replace the dedicated hardware boxes for the network functions, for example, home gateway, enterprise access router, carrier grade NAT and etc. So as to improve the reusability, allow more vendors into the market, and reduce time to market. NFV architecture includes a NFV controller (orchestrator) to manage the virtual network functions and the infrastructure resources. (Note: will use terms defined by ETSI NFV ISG in the next version.)

NF: A functional building block within an operator's network infrastructure, which has well-defined external interfaces and a well-defined functional behaviour. Note that the totality of all network functions constitutes the entire network and services infrastructure of an operator/service provider. In practical terms, a Network Function is today often a network node or physical appliance.

vNF: virtual network function, an implementation of an executable software program that constitutes the whole or a part of an NF that can be deployed on a virtualisation infrastructure.

VM: virtual machines, a program and configuration of part of a host computer server. Note that the Virtual Machine inherits the properties of its host computer server e.g. location, network interfaces.

3. Problems of Service Configuration

There are several problems in the context of remote software installation, which makes it different from the traditional ways.

First, it is a remote operation. For example, in the NFV framework, a software is installed according to the user (home user or enterprise user) requirements through NFV controller. It is not installed locally in the user's equipment, but remotely in the provider's network. NFV's control center needs to coordinate the necessary infrastructure resources for the installation. So the user does not have direct control over the software installation position or the hardware and software resources. But the controller has the direct control. In a result, the user needs to interact with the controller to accomplish the configuration of the components and his preferred locations for a software installation.

Second, the NFV controller is just like a broker for various software applications. There are different methods for the software installation. A proprietary method is that every software vendor has a plug-in in the NFV controller platform, and each end user uses the proprietary messages to interact with that software vendor's plug-in for the software installation. Another way is using standard messages to allow users to select their preferred software components for all different kinds of software installation. The drawback for every software vendor has its own proprietary messages for the software installation component configuration, is that it will make both the controller and the user environment more complex. A uniform and standard component configuration is more appropriate for this context.

Third, if the software vendor does not provide a clear description of these software components, then users do not know how to choose among those components. So the controller also needs a standard format to communicate with the software vendors, so as to acquire the detailed descriptions of the software components.

Fourth, dynamic configuration is another problem. A user may want to change its service configuration when the software is running. In the traditional context, a user logs into the server, and changes the service template in the server, then save it. It may become effect immediately or after reboot. But in the context of NFV, a user's virtual function may be installed in many virtual machines. It gets coo complex if we let the user maintains the installed virtual machines information and logs into each virtual machine to reconfigure the service template one by one. A centralized service template configuration modification is much more easier. The controller may be or not be aware of the meaning of these dynamic configurations, But it needs to know that this is a configuration file and the range of VMs that it applies to.

There are also resource requirements for the remote software installation, which are complement to the software components selection. There is lack of a standard for a user to tell the controller how much bandwidth, storage, CPU, memory are allocated to a specific software in the provider's network (perhaps there is existing standard for the virtual machine or host level resources), or just tell the controller allocate the resources dynamically for him.

A recommended resource requirements notification for a service instance is also needed between a software vendor and the controller.

4. Scope for Standardization

The key point is the information model. Network Function Virtualization needs standard information model so as to improve the interoperation. How to represent the user's functional and resource requirements, and how to map and apply these requirements to the underlying infrastructure is the key point to success. This specification on the stage only focuses on the virtual network function level at the beginning, but virtual overlay network of network functions should be extended in the near future. The narrowed scope for the current stage is:

(1) A protocol between the user and controller for software installation components choices, dynamic service configuration through the controller, and the resource requirements for the installation.

(2) A protocol between software vendor and the controller for the detailed description of the software components and the recommended resource requirements for service instance.

5. Security Considerations

This document does not introduce any new security threats. But for any solution to solve these problems, authentication is required between a user and the controller to verify whether the user is authorized to install that software. And the messages among the user, the controller and software vendor must be encrypted to prevent from interception attack.

6. References

6.1. Normative References

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

6.2. Informative References

[NFVE2E]   , "Network Functions Virtualisation: End to End Architecture, http://docbox.etsi.org/ISG/NFV/70-DRAFT/0010/NFV-0010v016.zip", .

Author's Address

Haibin Song
Huawei

Email: haibin.song@huawei.com