# Network Function Virtualization (NFV) - A Survey

Nandini Krupa Krishnamurthy
*Clemson University*
*nandink@clemson.edu*

***Abstract*—The goal of this survey is to provide an in-depth knowledge of Network Function Virtualization (NFV) and the contributions made by various researches. NFV has drawn immense attention from all corners. We will In this survey paper, we will discuss about advantages of NFV and how it is related to Software Defined Networking (SDN). We focus on the architecture of NFV and later move ahead to talk on the techniques used to implement this. We also discuss about the chanllenges faced by NFV and the future works related to this area.**

## I. Introduction

Network Function Virtualization has grabbed considerate attention from all the classes of studies. It is important to know what is Network Virtualization before we dive deep into Network Function Virtualization. Network Virtualization is a concept that helps the network functions to be installed, used, manipulated and refined on computer components [1]. In a nutshell, Network Function Virtualization (NFV) is a concept which is used to show the functions implemented in the data plane and executed on the network components. There are various advantages of using NFV. Some of the main ones are that it reduces capital expenses and operating expenses in comparison to the traditional routers and switches or any other middlebox in the system [2]. It also uses in its functions a combination of cloud and virtual technologies to create and develop network services elastically with a great scalability and automation. Therefore, NFV is an increasingly emerging technology in the areas where functions in data plane are actively implemented in custom made hardware. Fig 1 shows NFV architecture.

### *A. NFV Architecture*

NFV ideally composes of 3 components [2] [3]. They are:

- Network Function Virtualization Infrastructure (NFVI)
- Virtual Network Functions (VNFs)
- Management And Network Orchestration (MANO)

Each one is explained as below:

1) **Network Function Virtualization Infrastructure (NFVI):** VNFs are deployed in an environment created by NFVI which consist of both software and hardware components. The physical components present in it consists of Commercial-odd-the-shelf (COTS) computing hardware, storage devices, nodes and links. The functions include storage, processing and linking various components of VNF. The virtual resources are the instances of these functions. The virtualization layer present decouples these resources from the underlying physical resources. Virtual Machines represent are the instances of these resources and virtual links and nodes make up the virtual network connections. The functions of virtual nodes include hosting and routing. A virtual link connects two virtual nodes.

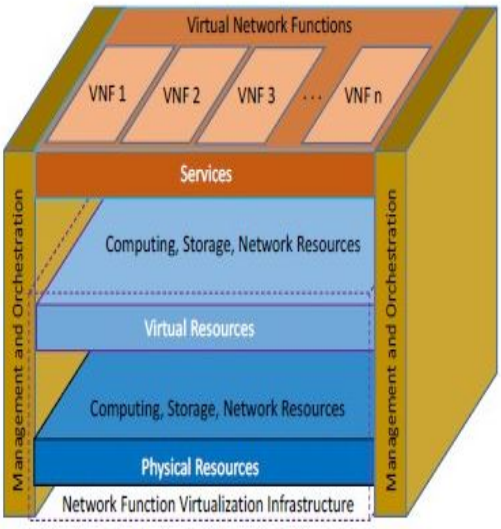


Fig. 1. Network Function Virtualization Architecture [3]

2) **Virtual Network Function:** A Network Function is a functional instance with a well defines an external interface and a well defines a function. Therefore, a VM is actuated as a VNF on a Virtual Machine. There may be the presence of many internal components in a VNF [3]. So, a single VNF can be imposed on many VMs where each VM can hold a component of a VNF. It is important here to note the TSP offers service here and it may contain more than one VNF. The way a service works is totally dependent on the VNFs it contains. This, in turn, is dependent on the behavioral aspects on the service.
3) **Management And Network Orchestration (MANO):** MANO framework was defined by the European Telecommunications Standards Institute (ETSI) [3]. It states that MANO provides the platform needed for implementing VNF and all other related works. As the name suggests, this includes the orchestration and man-

agement of all the resources and components associated with VNFs. In addition to this, it contains a database which stores information. It also holds models which can show the properties of all the functions, services and resources involved in a VM. It also explains the different communication interfaces between components of the system.

### *B. Advantages of NFV*

There are various advantages of NFV. Below are listed some. Fig 2 shows the advantages of some of these concepts.

1) Low cost of the system and low power usage and thereby helping to increase throughput and performance.
2) Decreased time of operation hence magnificently reducing maturation cycle.
3) The components are available in various versions which eases the sharing of resources various networks thus allowing a single platform for the use of resources.
4) The services in this network can be scaled up/down according to necessity.
5) Encourages innovation for the use of services and resources and lowers risk.

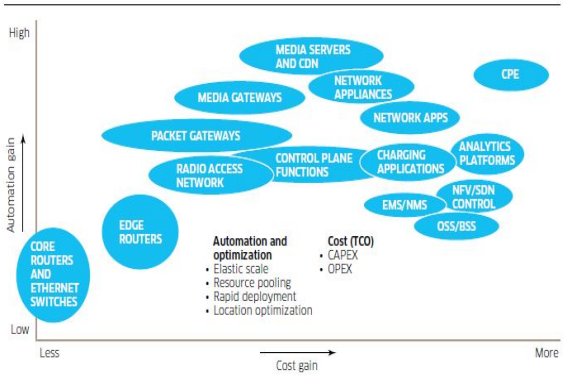


Fig. 2. Advantages of NFVs [11]

### *C. Relationship between NFV and SDN*

Fig. 3 defines the relationship between NFV and SDN. Though both are very different concepts, they are complementary to each other. But, they are independent of each other. Another point to note here is, both of these work independently, but when combined gives greater efficiency and value. NFV architecture can achieve its goals without using SDN, but when approaches of SDN such as separation of control plane and data forwarding plane are used, it can enhance performance by multiple times, aid operations and simplify maintenance.

## II. Main Techniques

In the previous section we have seen what are all the components of NFV. Here we will describe the techniques

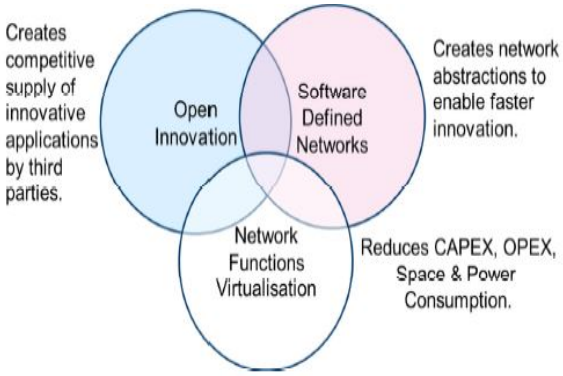


Fig. 3. Relationship between NFV and SDN [10]

each of these components use to enable NFV work smoothly. Recalling, NFV consists of three main components Network Function Virtualization Infrastructure (NFVI), Virtual Network Function and Management And Network Orchestration (MANO). Further, Fig 4 shows the framework of NFV.

- **NFVI** NFV primarily constitutes of both hardware and software. Hardware includes serves and software includes NFVI software. This software is used by the developers to build an efficient NFV system. And also, NFVI framework would help build an NF, or a group of NF to increase the efficiency of an NFV system. NFVI, MANO and NF work together to achieve the goal. We, in this section, defines some of the components/functions which constitutes the basic technique NFV works on. The end result of these is high reliability, throughput, scalability and security.
    1) **ClickOS:** Currently available middleboxes are components of the hardware and are not completely efficient in their performance and may be expensive and inflexible. Additionally, they run on Xen and KVM which Therefore software-based middleboxes are highly desirable. Click-OS [5] [6]is one such VM platform that is used to implement these middlebox functions. It solves problems of high resource usage and low performance. It is composed of MiniOS a Xen-based OS and Click modular router. MiniOS is used to run click for functions such as multi-user support, multiple address support, multiple file systems support saving CPS memory usage and response time and thus increasing efficiency. ClickOS provides high scalability by using lightweight VMs. In addition to this it is an open-source and product.
    2) **Elastic Edge (E2):** ElasticEdge (E2) is a framework offering solutions for dynamic scaling, NF Placement and thus helping to not code each of these every time. It encapsulates data such as details of parallelization, fault tolerance and data distribution. The E2 system consists of 3 components E2

Manager, E2 DataPlane (E2D) and Server agent. E2 manager monitors the E2 system along with creating and deleting Network Functions. A bunch of libraries for the execution of NFs is provided by the DataPlane. NFs are managed by Server Agent. E2 interfaces can be used by a user to define a set of policy statements for a certain traffic class. These are called as pipelets.

3) **NetVM and OpenNetVM:** NetVM provides a platform for NFV to allocate network functions on commodity servers. This is an environment built over KVM platform. By pipelining VMs, services are created. Its framework is composed of shared memory for delivering packets from one VM to another without copying. It also improves security, in the sense that packets are handled only in trusted VMs. OpenNetVM is an extension of NetVM which follows a similar architecture. Here network functions are carried out in Docker container. Notice that this is different from NetVMs where the network functions are carried out on KVM. The network functions are deployed in a simpler manner which enables easy replication of network functions thus enhancing scalability. OpenNetVM has dynamic configuration of services.

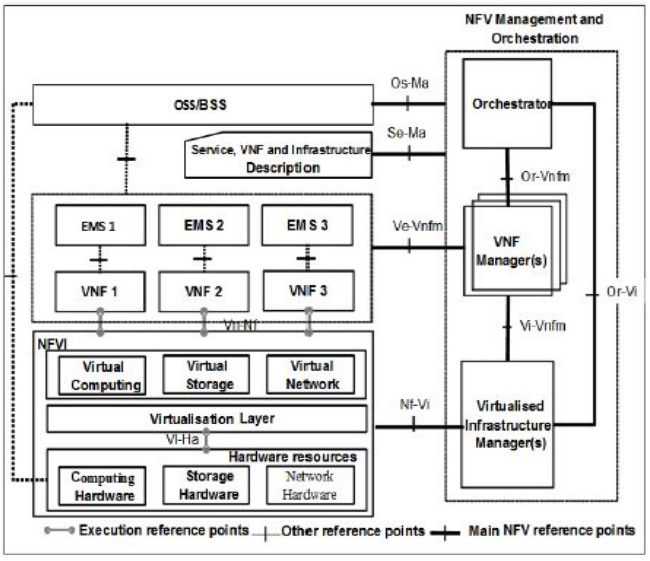


Fig. 4. NFV Framework [2]

- **VNF**
  1) **a. Software switches/routers:** There are two purposes of software switches/routers to forward packets between VMs and to replace the physical switches. For this, there are many technologies used. We will describe some of these here.
     a) **Open vSwitch (OVS):** There is a boom of virtual ports when datacenter is virtualized. Here every virtual port is connected to a physical switch which inversely affects scalability and mobility. OVS comes into the picture here. OVS in turn comprises of two modules user space daemon and data path kernel. The former is OS independent while the latter is OS dependent. A packet is first received by the kernel module, which gives out the instructions on what has to be done with the packet and based on the instructions, that packet is handled. These instructions are recorded for future handling of similar packets. The userspace daemon obtains OpenFlow messages send by a controller.
     b) **SnabbSwitch:** Snabbswitch acts as a virtual switch for KVM hypervisor. This is a open-source software and support various domains. It brings with it two addons LuaJIT and vhost-user. LuaJIT has been executed in Lua and vhost-user makes way for VMs to transfer packets directly to virtual switches in userspace or an example-snabbswitch.Snabbswitch is made up of three parts app, a link and the app engine. Apps are used to extend and alter SnabbSwitch, link acts as a buffer to carry packets between different apps and app-engine instructs apps and links and controls the flow of execution.
     c) **Click, RouteBricks, PacketShader:** Click modular router is an opensource software and helps in packet dropping and services. Routebricks is built with Click software and is in achieving great speeds with the help of multiple cores in a single server. Routebricks is also a open software. PacketShader is speacially designed to run on a system with GPU accelerators.
  2) **Middleboxes:** There are two frameworks which talk about implementation of middle boxes . they are firewalls and NAT. we will talk about these now.
     a) **Firewall:** The challenge in implementing firewall in a NFV system is its ever changing topology and and protecting its dynamism. VNGaurd is a framework that is specially used manage firewalls on a virtual system. These have three main uses:
        - Implemented in high level language and security policies.
        - Computes the ideal placement of virtual firewalls.
        - Ways to adapt virtual firewalls so that they are protected in a virtual network.

        So VNGuard is not an actual implementation of firewall, but a technique/framework to implement a software based firewall.
     b) **NAT:** NAT, when implemented as a software of

the stateful applications gives better results than those designed for hardware. These are very fast and are referred to as carrier-grade NAT. these carrier-grade NAT consists of three components one, NAT software working on a single host, two, a load balancer which allocates packets to many hosts, and dynamic scaling that is, adding or removing hosts on the go.

- **Management and Network Orchestration (MANO):** NFV MANO consists of
  1) Virtualized Infrastructure Manager (VIM)
  2) VNF Managers
  3) NFV Orchestrator

  VIM helps in managing the network servers and components that make up the NVFI hardware. VNF Managers are used to manage the functions of the network. NFV orchestrator acts as the scheduler and aids in creating by combining together the network functions. The main challenge in service chaining is the allocation of resources. This is managed by orchestrator and hence its basic role is to optimally allocate resources

## III. Issues and Problems

Though Network Function virtualization is a solution to many problems, it has its own challenges to be faced that downgrade its performance and hamper its performance. This section discusses about the issues and problems as faced by NFV [8].

1) **Security:** Security is important in any architecture. NFV should also possess a secure environment that is as near to that of a network function environment [8]. The various components of NFV faces various challenges. Some on the main one are described here. Virtualization environment domain called the Hypervisor may face unauthorized data leakage or unauthorized access. Computer domain may have shared resources which may breach security. Infrastructure domain also called the networking domain may share the logical networking layer which may hamper security. They also share NICs physically which may lead to lowering of privacy.
2) **Computing Performance:** A VNF should have a proper working conditions of all its components. VNF is made of various components which have to sync in working to obtain a high level efficiency [8]. Components which affect performance such as processor, clock rate, cache memory size, memory bandwidth, and speed have a very high impact on VNF. In a nutshell, VNF should have performance as good as network functions on a hardware.
3) **Interconnection of VNFs:** A virtual environment has components connected in a different way compared to a traditional network.in a classical network connections are through the network are by using Layer2 switches [8]. But a virtual environment cannot follow this. VNFs should use accelerated V-switches or NICs for this purposes. But implementing these are not easy.
4) **Portability:** There are various ways to implement VNFs. All these have their own advantages and disadvantages. These implementations have predictable performances which can dramatically increase resource usage and it is not efficient in resource allocation [8]. And this in turn makes it difficult for these software instances to execute in parallel. Again, the software this is working should be operating system dependent. Also, migration of these VNFs may pose a problem.
5) **Operation and Management:** The best implementation of virtual system are through simple drag and drop mechanisms in the orchestration management system. But this is not the case in the current implementations. To get this into existence, VNFs and also the architecture for computing should be defined using standard templates, which at present is not the case. Ideally, these templates also should enable automated management. All these have to followed without giving up on the services.
6) **Co-existence with Legacy Networks:** Since migration from a hardware platform to a virtual environment is still in its development stage, it is important that current virtual networks should co-exist with the legacy systems. Migration should also be easy from the current legacy systems to different virtual environments, but this is not the case as of now. Also, interaction between these two systems should be simple and easy without much affects, but again, this is not the case currently. The next problem is the network forwarding graph may be affected by this coexistence. And also this transition may affect security.

## IV. Future Trends

In this section, we talk about some of the areas that needs attention in the context of Network Function Virtualization. We describe some of the challenges and research trends related to NFV [9] [10].

1) **Portability/Interoperability:** The way the virtual systems are executed vary from one data center environment to the other. It has no standard definition of execution. Also, this variation is more with the ever increasing number of different vendors. Portability/Interoperability is a important point because two virtual systems in different data centers and from different vendors may create a new ecosystem all by itself, thought both may be dependent on each other. Portability also offers the vendor with an ability to allocate resources as needed without any bar, which reduces the efficicnecy. Therefore, one of the future goals may be to standardize the working principle for NFV on all environments and also development of a standard code for building NFV for all vendors.
2) **Performance Trade-off:** The NFV implementation of NFV is currently based on a hardware approach which decreases the performance. Here , basically, the challenge is not to negate the decrease in performance but to keep it as small as possible by using appropriate

hypervisors and softwares thus decreasing latency and increasing throughput. Virtual components should understand what they receive from hardware.

3) **Migration and co-existence of legacy and compatibility with existing platforms:** The current implementations of NFV must be compatible with the legacy systems and also its must also be compatible with it. The NFV architecture should help in supporting migration from the legacy hardware system to recently developed more intricated virtual systems. In a nutshell, NVF should be compatible with working in a hybrid environment made up of both legacy physical systems and more modern virtual systems.
4) **Management and Orchestration:** Managememt and Orchestration function should be constant. NFV helps SDN to conveniently make management and orchestration to work with the well defined standards. During this integration there is a chance that NFV may control the forwarding functions of the physical switches of the SDN.
5) **Automation:** Automation is always a positive aspect. Scalability can be obtained in great heights if the whole process is automated. This is an interesting research area.
6) **Security and Resilience:** There is no surety for NFV to be completely secure. This made network operators skeptical to use NFV. They have to be first assured about the NFVs availability, resilience and security. This gains utmost importance during the events of cyber attacks, reinstallation or reconfiguration. Potential fluxes might also occur which might have dangerous affects such as sudden decrease in performance or inefficient use of resources. So, research here is to find ways to stabilize network.
7) **Simplicity:** The virtualized network platforms which are available today are way too complex to understand and use. So the need of hour is to simplify these systems without compromising on its services and networks.
8) **Integration:** There are multiple types of virtual systems in the data environment. Standardising them and getting them all together to carry out seamless function is a major challenge because these would be created and sold by different vendors. This has to be done along with not incurring huge integration costs and deadlocks.

## REFERENCES


[1] Sandra McCann and Helen Shaw, Learn About Network Functions Virtualization, See https://www.juniper.net/documentation/en_US/learn-about/LearnAbout_NFV.pdf.

[2] Malathi VEERARAGHAVAN, Takehiro SATO, Molly BUCHANAN, Reza RAHIMI, Satoru OKAMOTO, Naoaki YAMANAKA, Network Function Virtualization: A Survey, IEICE Transactions on Communications, 2017, Volume E100.B, Issue 11, Pages 1978-1991, Released November 01, 2017, [Advance publication] Released May 16, 2017, Online ISSN 1745-1345, Print ISSN 0916-8516, https://doi.org/10.1587/transcom.

[3] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 236-262, Firstquarter 2016.doi: 10.1109/COMST.2015.2477041

[4] Network Functions Virtualization - An Introduction, Benefits, Enablers, Challenges & Call for Action, October 22-24, 2012 at the SDN and OpenFlow World Congress, Darmstadt-Germany, see http://portal.etsi.org/NFV/NFV_White_Paper.pdf

[5] J. Martins, M. Ahmed, C. Raiciu, and F. Huici, Enabling fast, dynamic network processing with ClickOS, Proc. Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN13, pp.6772, ACM, New York, NY, USA, 2013.

[6] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici, ClickOS and the art of network function virtualization, 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), pp.459473, USENIX Association, Seattle, WA, 2014.

[7] S. Palkar, C. Lan, S. Han, K. Jang, A. Panda, S. Ratnasamy, L. Rizzo, and S. Shenker, E2: A framework for NFV applications, Proc. 25th Symposium on Operating Systems Principles, SOSP15, pp.121136, ACM, 2015.

[8] H. Hawilo, A. Shami, M. Mirahmadi and R. Asal, "NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)," in IEEE Network, vol. 28, no. 6, pp. 18-26, Nov.-Dec. 2014. doi: 10.1109/MNET.2014.6963800.

[9] B. Han, V. Gopalakrishnan, L. Ji and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," in IEEE Communications Magazine, vol. 53, no. 2, pp. 90-97, Feb. 2015. doi: 10.1109/MCOM.2015.7045396.

[10] Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges & Call for Action, White Paper, October 22-24, 2012 at the SDN and OpenFlow World Congress, Darmstadt-Germany. This white paper is available at the following link: http://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[11] W. Yang and C. Fung, "A survey on security in network functions virtualization," 2016 IEEE NetSoft Conference and Workshops (NetSoft), Seoul, 2016, pp. 15-19. doi: 10.1109/NETSOFT.2016.7502434