

Network Function Virtualization in Content-Centric Networks

José Castillo Lema¹, Augusto Venâncio Neto², Flávio de Oliveira³, Sergio Takeo Kofuji¹

¹Escola Politecnica – Universidade de São Paulo

²Departamento de Informática e Matemática Aplicada (DIMAp) – Universidade Federal do Rio Grande no Norte

³Faculdade de Computação, Universidade Federal de Uberlândia
{josecastillo,kofuji}@usp.br, augusto@dimap.ufrn.br, flavio@ufu.br

Abstract. *Network Functions Virtualization (NFV) offers an alternative way to design, deploy, and manage networking functions and services by leveraging virtualization technologies to consolidate network functions into general-purpose hardware platforms. On the past years extensive effort has been made to evolve and mature NFV technologies over IP networks. However, little or no attempts at all have been made to incorporate NFV into Information-Centric Networks (ICN). This work explores the use and implementation of virtual Network Functions (VNFS) in Content-Centric Networks (CCN), and proposes the use of the Named Function Networking (NFN) paradigm as means to implement network functions and services in this kind of networks, distributing the network functions and services through the networks nodes and providing flexibility to dynamically place functions in the network as required and without the need of a central controller.*

Resumo. *Network Functions Virtualization (NFV) oferece uma maneira alternativa de projetar, implantar e gerenciar funções e serviços de rede, aproveitando as tecnologias de virtualização para consolidar funções de rede em plataformas de hardware de uso geral. Nos últimos anos, um grande esforço foi feito para desenvolver e amadurecer as tecnologias de NFV em redes IP. No entanto, pouca ou nenhuma tentativa foi feita para incorporar o NFV em redes orientadas a conteúdo (ICN). Este trabalho explora o uso e implementação de funções de rede virtualizadas (VNFS) neste tipo de redes, e propõe o uso do paradigma Named Function Networking (NFN) como meio de implementar funções e serviços de rede neste tipo de redes.*

1. General Information

Network functions, also known as "middleboxes", are playing an increasingly important role in modern networks, ranging from mobile networks, enterprise networks, to data-center networks. Surveys have been showing for some time that the number of network functions is comparable to that of the forwarding devices, indicating their significance ("A Survey of Enterprise Middlebox Deployments | EECS at UC Berkeley" n.d.). Network functions improve the network performance (e.g., Wide Area Network – WAN Optimizer, web proxy, load balancer), enhance the security (e.g., Intrusion Detection System – IDS / Intrusion Prevention System – IPS) or monitor the traffic (e.g., lawful interception, passive network monitor).

Conventionally, network functions were built in dedicated hardware for performance concerns, which incur high capital investment and operating expense. Furthermore, they were hard to manage. Their replacement and upgrade involve non-trivial human labor. In light of this situation, Network Function Virtualization (NFV) (Li and Chen 2015) was proposed, aimed to address these issues by leveraging virtualization technologies to consolidate network functions into general-purpose hardware platforms. NFV, along with Software-Defined Networking (SDN) paradigm, enables automated management of the whole life cycle of virtual network functions, leading to resource efficiency and expense reduction.

The initial perception of NFV was that virtualized capability should be implemented in data centers. But a service provider should be free to locate NFV in all possible locations, from the data center to the network node to the customer premises. This approach, known as distributed NFV, has been emphasized from the beginning as NFV was being developed and standardized, and is prominent in the NFV European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG) documents ("ETSI GS NFV-SWA 001 V1.1.1 (2014-12)" n.d.).

On the past years extensive effort has been made to evolve and mature NFV technologies over IP networks. However, current routing protocols deployed in IP networks constrain how packets can be deviated from a well-defined path (e.g., shortest path) and thus cannot take full advantage of the great flexibility offered by NFV. Moreover, SDN imposes the use of a central controller, which may lead to performance penalties and jeopardizes scalability capabilities (Bianco et al. 2010). Information-Centric Network (ICN) (Dannewitz et al. 2013) is a different approach that introduces Content Names to decouple the user interests from data location, improving in-network caching, management of mobility, multicast and peer-to-peer communications.

Following the ICN line of thinking, Named Function Networking (NFN) (Tschudin and Sifalakis 2014) has proposed that names should not only refer to data but also to functions and computation tasks. In NFN the network's role becomes to resolve names to computations. By naming functions, the network starts acting like a computing

machine, capable of not only caching content but also computation results. While NFN was originally planned for tempering, processing and delivering data, named functions use can also be extended to deal with network management related functions.

The key contribution of this work is the use of named functions and the NFN paradigm as means to implement network functions and services in content-centric networks, distributing the network functions/services through the networks nodes and providing flexibility to dynamically place functions in the network as required. The Dynamic Named Functions Architecture (DNFA) was proposed, a scalable and flexible framework that allows placing functions in the network by leveraging the NFN layer, following the ETSI NFV architectural guidelines. Furthermore, DNFA explores the distributed nature of named functions and takes advantage of its preferential execution opportunism to gain gratuitous parallelism and asynchronous computations. DNFA can bring many benefits, from improving operational efficiency and reducing resource usage to shorter deployment/upgrade intervals and cost efficient realization of network functions in software deployed over commodity hardware, without the need of a central controller.

The remainder of this paper is structured as follows: Section II provides a related work to underline the value of our contribution. Section III presents the proposed framework, with deeper information of its mechanisms and architecture. Section IV explains its implementation and shows some results of evaluations in the control plane.

2. Related work

From the early definitions of ICN, several works implemented various network functions into this kind of networks: access control (“Access Control Enforcement Delegation for Information-Centric Networking Architectures” n.d.), access privacy (Mohaisen 2017), lightweight authentication/secured routing, service discovery and management operations, among others.

The authors themselves of the NFN paradigm have already explored content access (Marxer, Scherb, and Tschudin 2016) and monitoring protocol (Mansour and Tschudin 2016) implementation through NFN. However, these functions are manually deployed in absence of a framework to automate deploy, monitor and scale.

3. Functional Architecture

This section describes conceptually the functional architecture proposed by the Dynamic Named Function Architecture (DNFA), a scalable and flexible framework that allows placing functions in the network by leveraging the Named Function Networking (NFN) layer, following the European Telecommunications Standards Institute (ETSI) Network Function Virtualization (NFV) architectural guidelines.

DNFA aims to distribute network functions and services through the network infrastructure using the distributed nature of named functions. DNFA can bring many benefits, from improving operational efficiency and reducing resource usage to shorter deployment/upgrade intervals and cost efficient realization of network functions in software deployed over commodity hardware, without the need of a central controller.

The DNFA proposal owns the following contributions:

- an integrated management architecture, including an orchestrator platform, for the automated deployment, management, monitoring, optimization and lifecycle management (e.g. instantiation, configuration, update, scale up/down, migration, termination, etc.) of named functions over network infrastructures;
- the distribution of network services/functions through the network infrastructure taking advantage of the distributed nature of named functions, using its preferential execution opportunism to gain gratuitous parallelism and asynchronous computations;
- a native and purely content-based non central controller-dependant network function execution environment, allowing the creation of more agile content-centric networks;

DNFA architecture is based on the ETSI NFV architectural guidelines. Fig. 1 represents the reference architecture of a DNFA framework. The bottom left of the picture represents the NFN Infrastructure, containing the Network Elements (NEs) that support ICN resolution (NFN capable or not). This infrastructure provides the necessary resources to physically span over several locations.

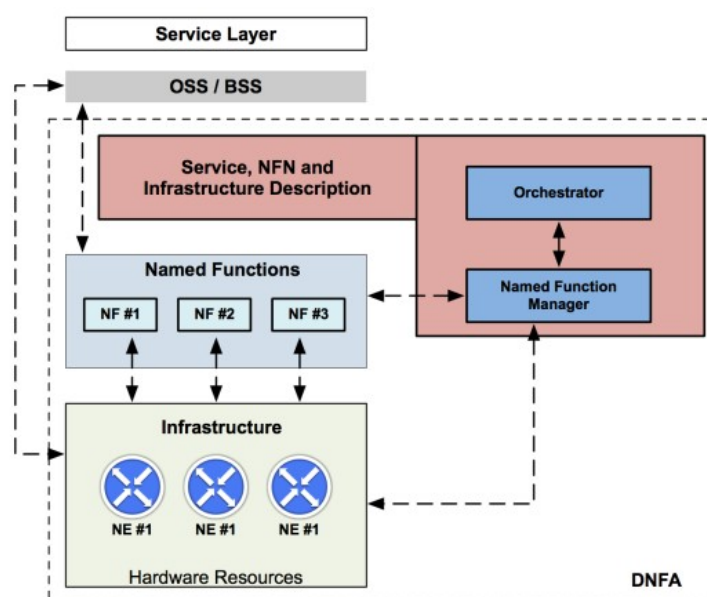


Figure 1. Functional Architecture

The middle left side of Fig. 1 contains the Named Functions, which use the resources provided by the NFN Infrastructure. The right side contains the management and orchestration elements. The Named Function Manager is responsible for the lifecycle management of named functions (placement, configuration, update, scale up/down, migration, termination, etc.). Finally, the Orchestrator is responsible for the management and implementation of the network services on the Network Infrastructure.

The top left corner are the Operation Support Systems (OSS) / Business Support Systems (BSS) of an operator as well as the Service, NFN and Infrastructure Description. The Service, NFN and Infrastructure Description provides information about a particular service (description, objective, limitations, etc.) and NFN information models. Finally, the Service Layer represents where services provided to the end user are modeled.

4. Experiments

In order to implement the reference architecture, CCN-lite (“CCN-Lite Project” n.d.) was chosen. CCN-lite is a reduced and lightweight (yet functionally interoperable) implementation of the CCN protocols. It supports named functions for letting clients express results instead of accessing only raw data. It admits Java, Scala or Python to host function execution and to interface to a NFN network.

All experiments were conducted on by network emulation experimentation through Mininet through a DPI VNF implementation (whose execution takes roughly 60 ms), considering the topology depicted in Fig. 2 with four network nodes. Relays D has the desired content and five relays A and B act as clients. Relays P is the only NFN-enabled node of the topology, and has been implemented both as a traditional IP-based DPI VNF and a NFN DPI function.

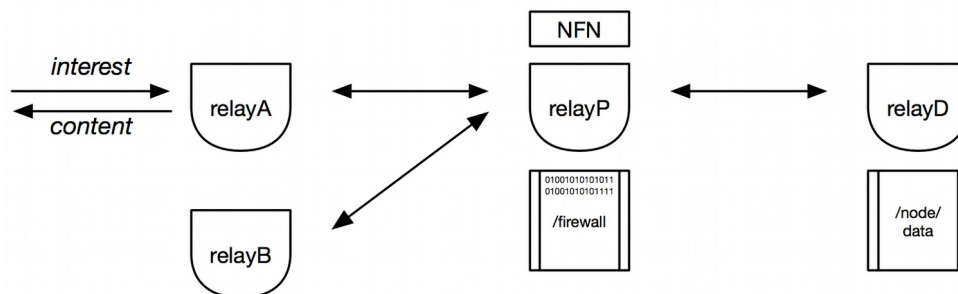


Figure 2. Hybrid CCN/NFN Infrastructure Topology

Network nodes have been assigned constant cache sizes, and a cache expiration time of 10 seconds. 20 users were spawned at a hatch rate of 2 user spawned per second, equally distributed between both relays RA and RB. Each users made consecutive and random requisitions from 100 ms to 200 ms. Content requests have been modeled as Poisson processes with $\lambda = 10$ to simulate content popularity.

Fig. 3 shows the requisitions per second (RPS) delivered by each implementation against the number of users. The CCN-based NFN implementation caches both most popular content and the computation result of the network function invocation managing to deliver more than 5 times requisitions per second than its IP-based alternative.

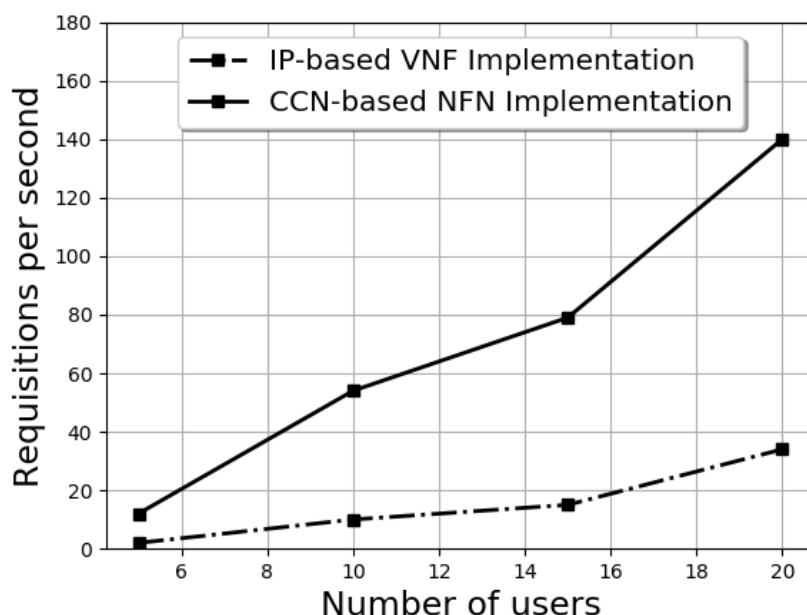


Figure 3. Requests per second

5. Conclusion

This work explored the use and implementation of VNFs in CCNs, and proposed the use of the NFN paradigm as means to implement network functions and services in this kind of networks, distributing the network functions and services through the networks nodes and providing flexibility to dynamically place functions in the network as required and without the need of a central controller. This brings many benefits, improving operational efficiency and reducing resource usage without the need of a central controller.

References

- “A Survey of Enterprise Middlebox Deployments | EECS at UC Berkeley.” n.d. Accessed May 12, 2016. <https://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-24.html>.
- “Access Control Enforcement Delegation for Information-Centric Networking Architectures.” n.d. Accessed December 31, 2018. <https://dl.acm.org/citation.cfm?id=2342507>.

- Bandyopadhyay, Soma, Munmun Sengupta, Souvik Maiti, and Subhajit Dutta. 2011. "ROLE OF MIDDLEWARE FOR INTERNET OF THINGS: A STUDY." *International Journal of Computer Science and Engineering Survey* 2 (3): 94–105.
- Bianco, A., R. Birke, L. Giraudo, and M. Palacin. 2010. "OpenFlow Switching: Data Plane Performance." In *2010 IEEE International Conference on Communications (ICC)*, 1–5. <https://doi.org/10.1109/ICC.2010.5502016>.
- "CCN-Lite Project." n.d. Accessed May 2, 2016. <http://www.ccn-lite.net>.
- Dannewitz, Christian, Dirk Kutscher, Börje Ohlman, Stephen Farrell, Bengt Ahlgren, and Holger Karl. 2013. "Network of Information (NetInf) - An Information-Centric Networking Architecture." *Comput. Commun.* 36 (7): 721–735. <https://doi.org/10.1016/j.comcom.2013.01.009>.
- "ETSI GS NFV-SWA 001 V1.1.1 (2014-12)." n.d. Accessed April 30, 2016. <http://docplayer.net/3007035-Etsi-gs-nfv-swa-001-v1-1-1-2014-12.html>.
- Li, Y., and M. Chen. 2015. "Software-Defined Network Function Virtualization: A Survey." *IEEE Access* 3: 2542–53. <https://doi.org/10.1109/ACCESS.2015.2499271>.
- Mansour, Dima, and Christian Tschudin. 2016. "Towards a Monitoring Protocol Over Information-Centric Networks." In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 60–64. ACM-ICN '16. New York, NY, USA: ACM. <https://doi.org/10.1145/2984356.2984378>.
- Marxer, Claudio, Christopher Scherb, and Christian Tschudin. 2016. "Access-Controlled In-Network Processing of Named Data." In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, 77–82. ACM-ICN '16. New York, NY, USA: ACM. <https://doi.org/10.1145/2984356.2984366>.
- Mohaisen, Abdelaziz. 2017. Systems, devices, and methods for protecting access privacy of cached content. United States US9578125B2, filed May 5, 2014, and issued February 21, 2017. <https://patents.google.com/patent/US9578125B2/en>.
- Tschudin, C., and M. Sifalakis. 2014. "Named Functions and Cached Computations." In *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, 851–57. <https://doi.org/10.1109/CCNC.2014.6940518>.