Modelling the NFV Forwarding Graph for an Optimal Network Service Deployment

Jordi Ferrer Riera[1], Xavier Hesselbach[2], Mateusz Zotkiewicz[3], Maciej Szostak[3], and Juan-Felipe Botero[4]

[1] *Fundació i2CAT, Distributed Applications and Networks Area, C./ Gran Capità 2-4,*
Barcelona, Catalonia, Spain, 08034, email: jordi.ferrer@i2cat.net

[2] *Universitat Politècnica de Catalunya, Departament d'Enginyeria Telemàtica, Jordi Girona, 1-3,*
C3 Campus Nord, Barcelona, Catalonia, Spain, 08034, email: xavierh@entel.upc.edu

[3] *Warsaw University of Technology, email: m.zotkiewicz@elka.pw.edu.pl, szoscio@gmail.com*

[4] *Universidad de Antioquia. Dept. Ingeniería Electrónica y de Telecomunicaciones. Calle 67 # 53 – 108,*
Medellín – Colombia,. email: juanf.botero@udea.edu.co

ABSTRACT

The NFV is focused at virtualizing network functions such as gateways, path computing devices, resource allocation mechanisms, proxies, firewalls and transcoders, traditionally carried out by specialized hardware devices, and migrating these functions to software-based appliances. In this paper, an analytic model for the NFV forwarding graph is proposed with the aim to optimize the execution time of the network services deployed.

Keywords: network functions virtualisation, NFV forwarding graph, SDN, service chain, optimization.

1. INTRODUCTION

The advent of new ICT access technologies (e.g. Cloud-based services, Big Data, 5G infrastructures, or even Cloud-Radio Access Networks) will strengthen the Digital Society. Over time, any application and service, mobile or not, will be given the potential to connect to anything at any time, from people and communities to physical things, processes, content, working knowledge, pertinent and relevant information, or even goods of all sorts, in entirely flexible and reliable ways. Network management for such a Digital Society becomes a fundamental element in order to ensure that all the requirements from both the end-user and the providers (i.e. service and infrastructure) are met. Software-Defined Networking (SDN) has swiftly become an important aspect of the strategy to address the network management requirements as identified by the commercial players in this market. From a functional perspective, SDN can be considered as the physical separation of the network control plane from the forwarding plane, where a control plane controls several devices [1].

With the advent of SDN technologies, a new generation of network programmable applications emerged [2]. They were built over the northbound APIs, homogenized by the corresponding controller using novel protocols, such as the OpenFlow protocol [3]. Those APIs enabled the development of software-oriented network applications performing specific control functionalities of the network, paving the way towards development of specific network functions. In November 2012, the virtualization term was further developed by the ETSI Industry Specification Group who coined the term NFV. This concept is focused at virtualizing network functions such as gateways, proxies, firewalls and transcoders, traditionally carried out by specialized hardware devices, and migrating these functions to software-based appliances, deployed on standard high volume (SHV) servers. Following this approach, the migration of most of the in-network operations from hardware to software modules leads to various benefits including: (i) efficient management of hardware resources, (ii) rapid introduction of new network functions and services to the market, (iii) easy of upgrade and maintenance, (iv) exploitation of existing virtualization and cloud management technologies for the NFVs, (v) significant CAPEX and OPEX reduction, (vi) enabling a wide variety of ecosystems; and (vii) encouraging openness within the ecosystem [4]. In the optical realm, network functions virtualization is being used mainly for core virtualized network functions that process high-volume traffic, such as session border controllers and/or serving gateways [7]. Optics are the technology to integrate the different network functions, since it provides the required flexibility and packet processing capabilities.

The rest of the paper is structured as follows. Section 2 presents the on-going related work within the community. Then, Section 3 and 4 contain the problem formalisation and the objective metrics to be analysed and evaluated for the metrics. Later, Section 5 describes and specifies the different scenarios to be used for the analysis and evaluation of the proposed solution. Finally, Section 6 concludes the article.

2. RELATED WORK

Network functions virtualization is seen as a highly complementary technology to SDN, where NFV can provide a network architecture vision in order to virtualize network node functions into independent, software-based modules or blocks, which can be later on chained and combined to compose specific network services. One of the most relevant research questions that appear in this area is how to concatenate or how to chain the different network functions in order to effectively and efficiently compose the different services. ETSI NFV presents the Use Case entitled "VNF Forwarding Graphs", where a graph defines the sequence that packets traverse [5]. VNF forwarding graphs are the analogue of connecting existing physical appliances via cables, i.e. the forwarding graphs provide the logical connectivity between the virtual appliances that need to be chained, and allocated onto the actual NFV infrastructure.

Authors in [6] provide a model for formalizing the chaining of network functions using a context-free language. They process deployment requests and construct virtual network function graphs that can be mapped to the network. They describe the mapping as a mixed integer quadratically constrained program in order to find the placement of the network functions and chaining them together considering the limited network resources and requirements of the functions. The approach differs from the one proposed in this manuscript in the formalization of the function chains. Besides, authors in [7] propose to use optical connectivity within the Data Centres to support high-throughput connectivity between different functions in order to enable the anywhere, and anytime-fashioned placement of virtual network functions. Additionally, several approaches to practical implementation of NFV-enabled services in different environments have been proposed. In [8] the authors propose the virtualization of the evolved packet core (vEPC) in mobile networks as the solution for future 5G services in the mobile realm. Also authors in [9] provide an efficient placement mechanism for virtualized network functions including hybrid services, i.e. composed of both virtual network functions and physical functions already existing in the substrate.

Last but not least, different standardisation efforts are being carried out. The most relevant is the one taking place at the ETSI NFV industry standard group, where the NFV concept, use cases, and framework have been defined. It is also worth to mention the IRTF NVF Research Group, recently established with the aim of providing focus on research problems associated with these virtualization topics.

3. PROBLEM FORMALISATION

The goal of the VNF scheduling problem is to find time slots in which activities should be processed under given constraints. Therefore, it is necessary to allocate the corresponding time slots for the virtual network functions composing different network services. The problem can be described in the following way:

Let's consider a set of N network services NS_1.. NS_N to be processed, each NS_j consisting of n_j VNFs F_{ij}, $i = 1..$ $n_{j,}$ to be executed in order: F_{1j}, F_{2j},.. $F_{n_j j}$. So, they are service chains.

Let's also consider a set of m multi-purpose servers, M_1.. M_m, assuming that each server can process only a single instance. Each F_{ij} will take p_{ij} time-units to be processed, on a dedicated machine $u_{ij} \in \{M_1, \ldots, M_m\}$, assuming sufficient buffer space available to store services waiting to process the next F_{ij}.

Two objectives are planned: to minimize the completion time of an individual network service chain, and also to minimize the total network service time including all the chains.

To this end, the following convenient notation is considered:

F are the network functions; N the network services; S the servers; $F(n)$ the network functions belonging to network service n; $F(f)$ the network functions that cannot be executed before finishing the execution of network function f. In the model a number of sets are used. Notice that in the problem network services can be understood as chains of network functions. The notion of network services is indirectly modeled using sets $F(f)$ that define relations between network functions, i.e., for each network function we define a set (possibly empty) of network functions that cannot be initiated before the considered network function is successfully executed.

The following constants are defined: $t(f; s)$ is the time needed by a server s to execute the network function f. Constant $t(f; s)$ is responsible for a server classification: By setting different execution times for different network functions on different servers, it is possible to easily create classes of servers. In addition, setting appropriate constants $t(f; s)$ to sufficiently large numbers, some functions can be easily blocked from being executed on particular servers. It is also defined the constant M, so called "big-M" constant, frequently used in MILP programs to express relations between binary and real variables.

The following variables are defined: z is the finish time of the last served network function. v_f is the starting time of executing network function f, e_{fs} is a binary variable (1 if network function f is executed at server s; 0 otherwise) and $a_{ff'}$ is another binary variable (1 if network function f is executed after network function f'; 0 otherwise).

The objective function is defined as follows:

$$\min(A \cdot z + \textstyle\sum_{n \in N} z_n) \tag{1}$$

Where the value of A controls the weight of z. For $A = 0$, the sum of individual service chains time becomes the objective to minimize. For big A, the weight is balanced to z.

Variable z represents a moment in time when all network functions are already executed.

The following constraints are considered:

$$z \geq v_f + \textstyle\sum_{s \in S} t(f,s) \cdot e_{fs} \qquad \forall f \in F \tag{2}$$

$$v_f + \textstyle\sum_{s \in S} t(f,s) \cdot e_{fs} \leq v_{f'} \qquad \forall f \in F, \forall f' \in F(f) \tag{3}$$

$$v_f + t(f,s) \leq v_{f'} + M \cdot (a_{ff'} + 2 - e_{f_s} - e_{f'_s}) \qquad \forall f, f' \in F, \forall s \in S \qquad (4)$$

$$a_{ff'} + a_{f'f} = 1 \qquad \forall f, f' \in F : f \neq f' \qquad (5)$$

$$\sum_{s \in S} e_{f_s} = 1 \qquad \forall f \in F \qquad (6)$$

Equation (2) guarantees that z cannot be smaller than the finish time of any network function. In (2), the term $\sum_{s \in S} t(f,s) \cdot e_{f_s}$ is just a time needed to execute function *f* on a selected server represented by e_{fs}. Equation (3) consider network services that impose various constraints on network functions they are built of. There, time of executing network function *f'* that follows another network function *f* (i.e.: *f'* ∈*F*(*f*)) has to be greater than the finish time of executing network function *f*.

Equation (4) prevents network services from being executed in parallel on the same server: it is assumed that each server can process only a single network function at a time. In other words, consider two network functions *f* and *f'*. Assume that *f* is executed before *f'* (thus, $a_{ff'} = 0$) and both are executed at server *s*. If all the conditions are met, constraint (4) reduces to $v_f + t(f,s) \leq v_{f'}$, which means that the starting time of executing network function *f'* has to be after the finish time of executing network function *f*. On the other hand, when at least one of the mentioned conditions is not satisfied ($a_{ff'} = 1$ or $e_{fs} = 0$ or $e_{f's} = 0$), constraint (4) reduces to $a \leq b + c \cdot M$, where $c \cdot M \gg a$, $b \geq 0$; thus, it is always satisfied regardless the values of the variables. Obviously, constant *M* has to be sufficiently large. In the considered problem it can be equal for instance to $min_{s \in S} \sum_{f \in F} t(f,s)$ which is a minimum time needed to execute all network functions on the fastest server.

Finally, constraint (5) ensures that not all variables $a_{ff'}$ can be equal to 1, while constraint (6) ensure that all network functions will be executed.

4. METRICS

This section introduce the metrics proposed in the analysis of the services allocation. As discussed in the previous section, in general, the total execution time is the objective to minimize, but other parameters such as the resources consumed or the utilization of the servers are relevant for the operators in order to reduce the costs (CAPEX and OPEX). The following metrics are defined:

- Total execution time. Therefore, the objective function is Min *z*
- Service execution time (per service): z_n, for all *n*. This metric can also be minimized, regarding *z*.
- Service execution time per server: This is a graphical representation (timeline, chronogram, or other) in each server *s*, for the values *f* $v_f + t(f, s)$, for all *f* running on it.
- Concerning servers utilization, 2 metrics are considered:
  - % Servers utilization (busy time / total time)
  - % each server utilization (and its associated "timeline")
- The service execution time for all the servers altogether. The objective of this metric is to visually detect the “gaps" in the servers (i.e.: idle times).

5. TEST SCENARIOS

Table 1. Test scenarios.

Environment	Number of servers	Computing Capacity	Complexity of the network services	Scenario
Edge computing	High number of available servers (30)	Low (high *t*)	Simple NSs (composed of 3 or less VNFs)	1
			Complex NSs (composed of 6 or more VNFs)	2
Data centre	Low number of available servers (5)	High (low *t*)	Simple NSs (composed of 3 or less VNFs)	3
			Complex NSs (composed of 6 or more VNFs)	4
	High number of available servers (30)		Complex NSs (composed of 6 or more VNFs)	5
Heterogeneous environment	Medium number of servers (10)	Combined high-low	Random service composition	6

Different test scenarios have been defined in order to analyse and evaluate the effectiveness, performance, and efficiency of the proposed allocation mechanism for the forwarding graphs, considering the aforementioned metrics. The scenarios consider variations in terms of: (i) number of servers available in the NFV Infrastructure to process the functions; (ii) the computing capacity of each server to process each allocated function; and (iii) the complexity of the different network services, i.e. the number and topology of network functions composing each service.

The Table 1 contains the summary of the different scenarios included for each case, considering both edge computing, and core computing, where big data centres reside. The aim of the scenarios is to identify for each scenario which type of network services may fit better, so the execution of the service will be optimised once the placement of the functions has occurred.

6. CONCLUSIONS AND FUTURE WORK

The paper has discussed and presented a formulation in the NFV forwarding graph modelling for an optimal network service deployment. Since the service chaining problem is, basically, a resource allocation problem, the proposed approaches to solve it may be integrated in resource allocation frameworks for Future Internet architectures such as ALEVIN [10]. Here, common economic metrics such as cost and revenue, performance metrics such as acceptance ratio and network utilization, or environmental metrics such as energy efficiency may be used to evaluate and compare different approaches. Besides, a multi-objective function (minimum z and minimum energy consumed) can be considered. Among open works, we include the management of networks made of contributions from several IPs (Infrastructure Providers), with inter-connected NFVI-PoPs, including the case with a limited number of servers in each IP (distributed server farm partitions). Besides, the inclusion of classes of servers (not all the servers can execute all the VNF), and servers with different capacities. Finally, the impact of priorities can also be analysed.

ACKNOWLEDGEMENTS

This work is partially supported by the projects funded by the "Ministerio de Economía y Competitividad" of the Spanish Government, Europa Investigacion "Arquitectura con conocimiento del entorno de la futura Internet" EUIN2013-51199 and the project "Redes troncales y de acceso inteligentes definidas por software" TEC2013-47960-C4-1-P. Also by the European Commission through the T-NOVA (Virtual Network Functions as a Service over Virtual Infrastructures) and the IST-XXYY CONTENT (Convergence of Optical Wireless Network and IT Resources in support of Cloud Services) projects, under grant agreement no. 619520 and no. 318514 respectively.

REFERENCES

[1] ONF SDN Definition https://www.opennetworking.org/sdn-resources/sdn-definition (retrieved 2015-04-01).

[2] B.A.A. Nunes, M. Mendonca, Nguyen Xuan-Nam, K. Obraczka, T. Turletti: A Survey of software-defined networking: Past, present, and future of programmable networks, *Communications Surveys & Tutorials, IEEE* , vol. 16, no. 3, pp. 1617-1634, third quarter 2014.

[3] Openflow switch spec. 1.4.0. Online https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf (retrieved 2015-04-01).

[4] ETSI ISG NFV group. GS NFV 002. Network Functions Virtualization (NFV); Architectrual Framework. Available online at http://www.etsi.org/technologies-clusters/technologies/nfv (retrieved 2015-04-01).

[5] ETSI ISG NFV group. GS NFV 001. Network Functions Virtualization (NFV); Use Cases. Available online at http://www.etsi.org/technologies-clusters/technologies/nfv (retrieved 2015-04-01).

[6] S. Mehraghdam, M. Keller, H. Karl: Specifying and placing chains of virtual network functions, in *Proc. 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, 2014.

[7] M. Xia, M. Shirazipour, Y. Zhang, H. Green, A. Takacs: Network function placement for NFV chaining in packet/optical datacenters, *Journal of Lightwave Technology*, vol.: PP , no. 99, DOI: 10.1109/JLT.2015.2388585, 2015.

[8] H. Hawilo, A. Shami, M. Mirahmadi, R. Asal: NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC), *IEEE Network Mag.*, vol. 28 , no. 6, pp. 18-26, Nov. 2014.

[9] H. Moens, F. De Turck: VNF-P: A model for efficient placement of virtualized network functions, in *Proc. 2014 10th International Conference on Network and Service Management (CNSM)*, pp. 418-423, 2014.

[10] A. Fischer, J. Botero, M. Duelli, D. Schlosser, X. Hesselbach, and H. De Meer: ALEVIN – A framework to develop, compare, and analyze virtual network embedding algorithms, *Electronic Communications of the EASST*, vol. 37, 2011.