Computer Networks 114 (2017) 95–110

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Virtualized network functions chaining and routing algorithms

Selma Khebbache[a,1], Makhlouf Hadji[b,2,*], Djamal Zeghlache[a,3]

[a] Instititi Mines-Télécom, Télécom SudParis, 17 Rue Charles Fourier, 91011 Evry, France
[b] Technological Research Institute SystemX, 8 Avenue de la Vauve, 91120 Palaiseau, France

ARTICLE INFO

Article history:
Received 6 July 2016
Revised 26 November 2016
Accepted 15 January 2017
Available online 18 January 2017

Keywords:
NFV
VNF
SFC
Infrastructure
Perfect 2-Matching
Optimization
Forwarding graph

ABSTRACT

We focus on the placement of Virtualized Network Functions (VNFs) chains in the Network Function Virtualization (NFV) context where NFV Infrastructures (NFVIs) are used to host the VNFs. The optimal placement of VNF (service) chains in hosting infrastructures is one of the key issues in the deployment of service functions in large-scale environments. The Virtualized Network Functions Chain Placement Problem VNF-CPP is NP-Hard and there is a need for placement approaches that can scale with problem size and find good solutions in acceptable times. We propose a matrix-based optimization and a multi-stage graph method that are cost efficient and improve scalability by finding solutions in polynomial times. These algorithms are compared with an exact formulation given by the Perfect 2-Matching leading to a polynomial variant of the VNF-CPP for VNF chains with 2 arcs. Simulation results for longer and more complex chains confirm the efficiency and scalability of the proposed methods and their ability to find good suboptimal solutions.

1. Introduction and motivation

Traditional network services deployments are tightly coupled to network topology be it physical, virtualized or hybrid. This limits service delivery and inhibits network and infrastructure providers from optimally using and sharing their resources by dynamically setting up slices per tenant and applications. The introduction of Network Function Virtualization (NFV) removes some of these barriers and provides the opportunity to decouple service deployment from network topology and enable the dynamic establishment of service chains according to tenant and application requirements on virtualized infrastructures. NFV leverages virtualization technologies (Hypervisors, Containers such as Docker, ...) to realize multi-tenant and multiparty services on shared infrastructures. The fact that NFV implements and deploys network functions and services by relocating them from dedicated appliances to pools of generic industry servers provides the required decoupling and flexibility to enable dynamic selection and placement of network services in hosting infrastructures.

One of the key issues, in this customized deployment of service functions in large-scale environments, is the optimal placement of the service chains in hosting infrastructures. Service function chaining defines and instantiates an ordered list of abstract service functions and ordering constraints to apply to application flows through classification.

The optimal placement of these chains in virtualized infrastructures is an NP-Hard problem that has been addressed to some extent in the current state of the art but additional research is required to accomplish efficiently the placement of complex chains in large scale infrastructures. Our central motivation is to realize this placement in polynomial times by proposing appropriate models and solutions that can scale with size of the hosting infrastructures and service chains. More specifically we focus on optimal Service Function Chaining (SFC) as defined by IETF [1], also known as Virtualized Network Functions-Forwarding Graph (VNF-FG) in ETSI [2], according to demand. The work in IETF SFC and ETSI NFV address the placement, control, deployment and management of chains involving both application level and network level services. Since these services are realized in software they can be deployed, migrated, relocated, shut down, activated and upgraded more easily. Typical virtualized services are Deep Packet Inspection (DPI), Firewall, Classifier, Load Balancer, Router/switche, NAT, DNS,. . ..

We are interested in providing optimization algorithms for the ETSI VNF-Forwarding Graph (VNF-FG) (or IETF Service Function

* Corresponding author.
E-mail addresses: selma.khebbache@telecom-sudparis.eu (S. Khebbache), makhlouf.hadji@irt-systemx.fr (M. Hadji), djamal.zeghlache@telecom-sudparis.eu (D. Zeghlache).
1 Selma Khebbache is a research fellow in the Wireless Networks and Multimedia Services Department at Telecom SudParis, Evry, France.
2 Makhlouf Hadji is a senior scientist at the Technological Research Institute - IRT SystemX, France.
3 Djamal Zeghlache is Professor, Head of the Wireless Networks and Multimedia Services Department at Télécom SudParis, one of the education and research centers of Institut Mines-Télécom, Evry, France.

http://dx.doi.org/10.1016/j.comnet.2017.01.008
1389-1286/

Chains - SFC) use case where service providers acquire cloud and networking services and resources in the form of a graph to use in their service creation. For example a service provider may design an end to end network service between two physical network functions involving several intermediate VNFs, fully or partially virtualized, delivered by infrastructure providers that have adopted NFV to support cloud services providers (CSPs). The actual network service is the set of packet flows that traverse the VNF-FG and the related physical network functions (PNFs). The network providers are faced with the mapping of the requested VNF-FGs by the CSPs on their infrastructures and the establishment of the required connectivity between the logical VNF-FG nodes and links while respecting the network forwarding paths or packet flows (PFs) requested by the CSPs. A typical sequence encountered in networks, would correspond to packets traversing, in sequence, a router implemented in a VNF, an intrusion detection system (IDS), a firewall, a classifier and a load balancer to distribute traffic flows across the network towards end points or pools of servers.

This paper addresses the mentioned optimal service chaining (or equivalently optimal VNF-forwarding graph Chain Placement Problem, noted by VNF−CPP) in providers infrastructures (identified as carrier networks in ETSI NFV).

1.1. Objectives and contributions

Our objective is to propose efficient and scalable algorithms that can find good solutions (ideally close to optimal) for the VNF−CPP. In order to have a reference for performance comparisons, we propose an exact solution based on “Perfect 2-Matching theory” providing a polynomial variant of the VNF−CPP. This exact solution (polynomial for VNF chains with 3 nodes) is used to assess the performance at small scales of our proposed heuristic algorithms for the VNF−CPP. Larger scale evaluation is performed for the heuristics only as the exact solution does not scale. The proposed methods are:

1. An exact algorithm based on Perfect 2-Matching theory to solve in polynomial time the case of VNF chains with 3 VNFs. This polynomial case will serve as a lower bound in tandem with a new family of valid inequalities (facet) to accelerate the resolution of the VNF−CPP for VNF chains with larger lengths (chains with at least 4 VNFs).
2. An approach based on matrix analysis: it combines products of matrices with a simple linear programm to find an optimal steering of the traffic flows on each placed VNF.
3. An approach based on multi-stage graphs: constructed as a new extended multi-stage graph representing servers available for hosting the required VNFs and their interconnections. Chains of the Forwarding Graph are then placed according to a maximum flow between vertices of the different stages of this graph.

We propose several algorithms that we feel can each solve independently the VNF−CPP and compare them to assess their relative performance and to provide insight on their efficiency. There is no intent of combining these algorithms that are in fact alternative and competing models proposed to address the VNF−CPP. Our goal is to show that they scale with problem size and achieve good placement of service chains in shared infrastructures.

1.2. Paper organization

Section 2 presents the related work. Section 3 describes a model for the VNF−CPP that includes the VNFs'sequencing obligations and the physical resources limitations, complexity of the VNF−CPP and the proposed Perfect 2-Matching solution, the matrix-based approach and the multi-stage heuristic. Section 4 reports the results of performance evaluation and comparison of the algorithms that highlight their efficiency, improved scalability and their ability to find good suboptimal solutions.

2. Related work

Authors in [3] address the VNF forwarding graph placement problem using a mathematical model based on linear integer programming to achieve VNF chaining and placement while taking into account two essential constraints : i) cloud computation constraints and ii) physical networking resources limitations. In other words, [3] trades off computation and the communication overhead during VNF chain placement. Due to the NP-hardness of the VNF-FG placement problem, the authors resort to a Greedy solution to scale. The performance of Greedy solutions depends strongly on the order in which VNF chains are placed and their sizes and complexity. The authors do not take into account network resources in the optimization and focused only on the hosting requirements and reducing communications costs without considering or meeting the bandwidth or flow requirements of the VNF chains. The ordering and the flow requirements in VNF chains have to be included in the objectives.

In [4], the focus is on improving execution time for network services deployed in network operators' infrastructures. Authors of [4] view this problem as equivalent to a classical resource allocation problem with multiple criteria such as cost, revenue, performance, etc. They proposed a simple linear programm without considering the order of the VNF chains and their flow requirements (bandwidth required on selected links, etc.

The problem of mapping and scheduling VNFs can also be found in [5] that proposes three greedy solutions and a tabu-search based heuristic to realize online joint mapping and scheduling of the VNFs. Performance for the proposed algorithms is evaluated using indicators such as acceptance ratio, cost and revenue also used as criteria for the optimization. The authors consider only the node hosting requirements of the VNFs in the mapping and scheduling process and disregard completely the link (arcs in the graph) requirements. The order of the VNFs and the flow requirements (bandwidth on links and hence latency between the VNFs) of the chains are not considered in the model.

NFV management is provided by the project Stratos [6] that proposes a detailed architecture to orchestrate VNFs outsourced to a remote cloud taking into account various constraints such as traffic engineering, VNFs horizontal scaling, etc. Authors in [6] propose a VNF deployment solution based mostly on input workload and hence do not fully address VNF chains placement and deployment.

The important objective of guaranteeing QoS to tenants and consumers in NFV when sharing infrastructures attracted a lot of interest lately. References [7–12] propose mathematical models for the VNF chains placement with routing constraints. The proposed models describe, however, a limited number of linear constraints and capture only a negligible part of the convex hull of the VNF−CPP. This causes scalability problems since the models all rely on Integer Linear Programming approaches. These proposed exact solutions do not scale for large problem instances[13]. and [14] also propose mathematical models, provide Integer Linear Programming solutions in some cases and Mixed Integer Linear Programming approaches in other cases (see [14] for instance). To cope with scalability problems, some references propose heuristic solutions but performance remains unsatisfactory or inadequate for the *VNF−CPP*. More in depth modeling and characterization of the VNF−CPP convex hull is required to obtain near optimal solutions fast (execution times in order of seconds) and to scale for large problem instances (problem sizes in hundreds and thousands of nodes and links). Instead of proposing yet another mathematical model of the VNF−CPP that takes considerable time to reach optimal solutions, we propose coupling linear programming approach