Department of Computer Science
University of California Davis

2063 Kemper Hall, One Shields Avenue
Davis, CA 95616-5294 USA
http://www.cs.ucdavis.edu
(530) 752-7004

Joint Virtual Network Function Placement and Routing of Traffic in Operator Networks

Abhishek Gupta*, M. Farhan Habib†, Pulak Chowdhury*,
Massimo Tornatore*, and Biswanath Mukherjee*
*{abgupta,pchowdhury,mtornatore,bmukherjee}@ucdavis.edu
†mohammad.f.habib@intel.com

April 20, 2015

Abstract

Network functions (e.g., firewalls, load balancers, etc.) are traditionally provided by proprietary hardware appliances. These hardware-appliances can be hardwired back to back to form a service chain, which provide services to application layer programs. So, hardware-based functions cannot be provisioned on demand since they are embedded in the network topology making creation, insertion, modification, upgrade and removal of service-chains complex while also slowing down service innovation. Hence, operators are interested in Virtual Network Functions (VNFs), which are virtualized over commodity hardware. These VNFs can be placed in Data Centers (DCs) or Network Function Virtualization (NFV)-capable network elements such as routers and switches. However, placement decisions of VNFs need to optimized according to the traffic in the network. In this document, we present a mathematical model for the placement of VNFs which ensures the service chaining required by traffic flows.

1 Introduction

Networks of today are comprised of a large variety of proprietary hardware appliances (middle-boxes) used to support network functions such as firewalls, Network Address Translator (NAT), Quality-of-Service (QoS) analyzers, etc. Hardware-based functions are embedded into the topology and enforce topological constraints on traffic in the network. Topological constraints require the traffic to pass through the node in the topology for satisfying service requirements leading to increased network resource (bandwidth) usage.

With rapid innovation at the application layer, user generated traffic has increased exponentially, leading to increase in bandwidth consumption. Application-generated traffic flows require new services leading to creation of new service chains. Other situation may demand removal/upgrade of service chains for obsolete/outdated services. As service chains are made up of middle-boxes, the cycle of service induction, modification, and upgrade/removal becomes a complex process. Network Functions Virtualization (NFV) gives the tools for operators to deal with future network traffic dynamically and in real time. As shown in Fig. 1, the predominant idea behind NFV is to replace vendor-specific hardware in the network with Commercial-Off-The-Shelf (COTS) hardware such as servers, switches and storage [1] placed in Data centers (DCs) and network nodes.

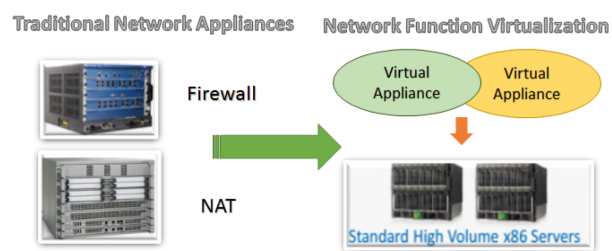

Figure 1: Network Function Virtualization (NFV) approach.

2 Service Chaining

Network functions process traffic flows singularly or in sync with other network functions in a service chain to enable a service. Examples of network functions include firewalls, load balancers, WAN optimizers, etc. The term “service-chaining” is used “to describe the deployment of such functions, and the network operator’s process of specifying an ordered list of service functions that should be applied to a deterministic set of traffic flows” [2]. An example of a service chain in today’s networks is shown as static middle-boxes wired together in Fig. 2. With rapid increases in traffic volume, traffic variety, and service requirements, operators need a more flexible method of service chaining. So, VNF service chains are being deployed for more flexible/agile service-chaining.

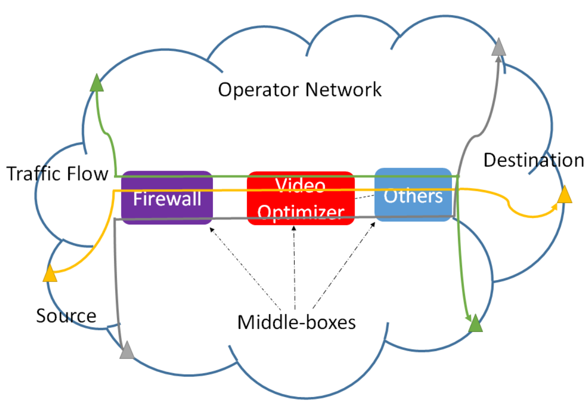

Figure 2: Static service chaining.

3 Problem Description

A network consists of multiple types of traffic flows. Each type of traffic has service requirements expressed as a chain of network functions. Operators need to satisfy service requirements for all the traffic flows by using the minimum number of network resources. Therefore, we model the problem as an optimization problem where the objective is to minimize the network resource consumption such that the service requirement (order of service chain traversal) for all the traffic flows is satisfied.

Virtual Network Functions (VNFs) are deployed to satisfy the service requirements of flows. The requirements of a VNF are similar to those of a VM in terms of computing and storage resources, i.e., both require CPU cores and memory (RAM/hard disk). However, traditional VMs are enterprise applications (e.g., database applications) deployed in cloud-computing environments while VNFs abstract network functions which process network traffic at line rate. This makes VNFs more bandwidth-intensive (virtualized routers) and compute-intensive (virtualized firewall: comptutational overhead for per-packet processing) with respect to enterprise application VMs, which are more memory-intensive and CPU-intensive. Although a VNF instantiation requires a certain amount of memory and disk space, we note that the performance of VNF scaling will depend upon the CPU-core-to-throughput relationship. The CPU-core-to-throughput relationship will depend on the VNF type which can be seen from Table 1, where a NAT (Network Address Translator) performs basic IP addressing functions, making it less CPU-intensive, while a Traffic Shaper needs to identify application traffic and perform operations which are compute-intensive and result in large number of CPU cores being used for higher throughput. So, we take the CPU-core-to-throughput relationship to be an essential characteristic of VNF operation, and we use it as the basis for VNF allocation in our mathematical model.

It has to be noted that a network may have multiple traffic flows generated from different application-layer programs.

Application	Throughput 1 Gbps	Throughput 5 Gbps	Throughput 10 Gbps
NAT	1 CPU	1 CPU	2 CPU
IPsec VPN	1 CPU	2 CPU	4 CPU
Traffic Shaper	1 CPU	8 CPU	16 CPU

Table 1: VNF requirements as per throughput [3].

Each traffic type will have different service requirements, each of which may be satisfied by different service chains implementing the same service. In this preliminary study , we try to reduce simulation time by considering the service chain to be given and by assuming one type of traffic in the network in any instance.

3.1 Problem Statement

Given a network topology, a set of DC locations, set of network nodes with virtualization support (NFV-capable nodes), traffic flows between source-destination pairs, the set of network functions required and the service chain, we determine the placement VNFs to minimize network resource (bandwidth) consumption.

3.2 Input Parameters

- $G(V,E)$: Physical topology of the network; V is set of nodes and E is set of links.
- $V_{DC} \subset V$: Set of DC locations.
- $V_{NF} \subset V$: Set of NFV-capable nodes.
- $\Psi_{(s,d)}$: Set of source $s \in V$ and destination $d \in V$ pairs with requesting traffic flows between them.
- $\Phi_{(s,d)}$: Traffic from source s to destination d.
- $K_{s,d}$: K shortest paths from source s to destination d.
- Γ: Set of network functions.
- Π: The service chain of functions.
- $R^{(i,j)}_{(s,d)}$: Set of paths from source s to destination d passing through link $(i,j) \in E$.
- Θ: Number of cores present per NFV node .
- N_f: Number of cores required by function f to serve 1 Gbps of throughput.
- $L^p_{s,d}$: Path length p between source s and destination d.
- $C_{i,j}$: Bandwidth capacity of link $(i,j) \in E$.
- $S_{u,v}$: Set of node pairs (u,v) such that u and v are nodes on path p with $u \in V_{NF}$ occurring before $v \in V_{DC} \cup V_{NF}$.

3.3 Variables

- $r_p^{(s,d)} \in \{0,1\}$: 1 if path p is chosen between source s and destination d.
- $l_v^f \in \{0,1\}$: 1 if function is used at f at node v.
- $q_{p,(s,d)}^{f,v} \in \{0,1\}$: 1 if function f is located on node v of path p between (s,d).
- $j_{p,(s,d),(u,v)}^{f_1,f_2} \in \{0,1\}$: 1 if functions f_1 and f_2 occur in service chain order at nodes u and v of path p between (s,d).

3.4 Problem Formulation

We mathematically formulate the problem through an Integer Linear Program (ILP).

$$Minimize: \quad \sum_{(s,d)\in\Psi_{s,d}} \sum_{p\in K_{s,d}} r_p^{(s,d)} \times L_{s,d}^p \times \Phi_{s,d} \tag{1}$$

$$\sum_{p\in K_{s,d}} r_p^{(s,d)} = 1 \qquad \forall(s,d)\in\Psi_{(s,d)} \tag{2}$$

$$\sum_{(s,d)\in\Psi_{s,d}} \sum_{r_p^{(s,d)}\in R_{(s,d)}^{(i,j)}} r_p^{(s,d)} \times \Phi_{s,d} \leq C_{(i,j)} \qquad \forall(i,j)\in E \tag{3}$$

$$\sum_{(s,d)\in\Psi_{s,d}} \sum_f l_{v,(s,d)}^f \times \Phi_{s,d} \times N_f \leq \Theta \qquad \forall v\in V_{NF} \tag{4}$$

$$q_{p,(s,d)}^{f,v} = l_{v,(s,d)}^f \cap r_p^{(s,d)} \qquad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall f\in\Gamma, \quad \forall p\in K_{s,d}, \quad \forall v\in p \mid v\in V_{DC}\cup V_{NF} \tag{5}$$

$$\sum_{p\in K_{s,d}} q_{p,(s,d)}^{f,v} >= 1 \qquad \forall f\in\Gamma, \quad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall v\in p \mid v\in V_{DC}\cup V_{NF} \tag{6}$$

$$j_{p,(s,d),(u,u)}^{(f_1,f_2)} \geq q_{p,(s,d)}^{f_1,u} \qquad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall p\in K_{s,d}, \quad \forall u\in p \mid u\in V_{DC}, \quad \forall(f_1,f_2)\in\Gamma \mid (f_1\to f_2)\in\Pi \tag{7}$$

$$j_{p,(s,d),(u,v)}^{(f_1,f_2)} = q_{p,(s,d)}^{f_1,u} \cap q_{p,(s,d)}^{f_2,v} \qquad \forall p\in K_{s,d}, \quad \forall(f_1,f_2)\in\Gamma \mid (f_1\to f_2)\in\Pi, \quad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall(u,v)\in S_{u,v} \tag{8}$$

$$\sum_{p\in K_{s,d}} \sum_{(u,v)\in p} j_{p,(s,d),(u,v)}^{(f_1,f_2)} >= 1 \qquad \forall(u,v)\in S_{u,v}, \quad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall(f_1,f_2)\in\Gamma \mid (f_1\to f_2)\in\Pi \tag{9}$$

$$\sum_t j_{p,(s,d),(t,u)}^{(f_1,f_2)} \geq j_{p,(s,d),(u,v)}^{(f_2,f_3)} \qquad \forall p\in K_{s,d}, \quad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall(f_1,f_2,f_3)\in\Gamma \mid (f_1\to f_2\to f_3)\in\Pi, \quad \forall(t,u,v)\in p \mid ((t,u)\in V_{NF}, v\in V_{DC}\cup V_{NF}) \tag{10}$$

$$\sum_{(u,v)} j_{p,(s,d),(u,v)}^{(f_1,f_2)} <= 1 \qquad \forall p\in K_{s,d}, \quad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall(f_1,f_2)\in\Gamma \mid (f_1\to f_2)\in\Pi \tag{11}$$

The objective function in Eq. (1) calculates the total bandwidth consumed by all the requested source-destination traffic flows using the length (number of hops) of the path used by the traffic flow and the bandwidth consumed. We enforce that traffic between a source-destination pair is served by a single path Eq. (2). The number of flows that can be provisioned on a link is constrained by the bandwidth of the link Eq. (3).

Each flow has service requirements which need to be satisfied. We deploy Virtual Network Function(VNF) for this purpose. Each VNF depending on the application it virtualizes requires a certain number of CPU cores for processing a unit of throughput (in terms of bandwidth). The CPU core requirement is not a constraint in a DC setting. However, in the case of a NFV-capable node the computation power is limited, and this limitation will impact the assignment of VNFs over that node. This constraint is realized using Eq. (4). Based on service requirements, the flow might be processed by one or more VNFs.

Depending on the functional dependencies between the VNF's, a service chain is created. In order to provide required service to the flow, it has to be mandated that it gets processed in the right order of VNFs along the service chain. Eq. (5) to Eq. (11) ensure that the service chain is implemented and traversed in correct fashion for the traffic flows.

Eq. (5)[1] checks if a function is located on a path taken by the traffic flows, while Eq. (6) enforces that this functions

[1]Eq.(5) can be linearly represented as below.

$$q_{p,(s,d)}^{f,v} \leq l_{v,(s,d)}^f \qquad \forall f\in\Gamma, \quad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall p\in K_{s,d}, \quad \forall v\in p \mid v\in V_{DC}\cup V_{NF} \tag{12}$$

$$q_{p,(s,d)}^{f,v} \leq r_p^{(s,d)} \qquad \forall f\in\Gamma, \quad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall p\in K_{s,d}, \quad \forall v\in p \mid v\in V_{DC}\cup V_{NF} \tag{13}$$

$$q_{p,(s,d)}^{f,v} \geq l_{v,(s,d)}^f + r_p^{(s,d)} - 1 \qquad \forall f\in\Gamma, \quad \forall(s,d)\in\Psi_{(s,d)}, \quad \forall p\in K_{s,d}, \quad \forall v\in p \mid v\in V_{DC}\cup V_{NF} \tag{14}$$

exists atleast on one of the paths, so that there is atleast one path to select. If a particular function is available on a path, we have to enforce that also the next function in the service is available on that path. Eq. (7) enforces service-chaining in a DC, i.e., if a function is located at a DC, then all its successors in the function chain will also be located here. This is only logical as a DC has sufficient resources to deploy VNFs. In the case of a NFV-capable node we need to ensure that the successive function is in the path either on the same node or a different successive NFV-capable node (part of the path), this is enforced using Eq. (8)[2], while Eq. (9) constrains that this dependency is realized in at least one of the paths between the source and the destination. Eq. (10) enforces service-chaining inside a network node by constraining that a later dependency ($f_2 \rightarrow f_3$) is possible only if an earlier ($f_1 \rightarrow f_2$) dependency is satisfied. Eq. (11) enforces that the dependencies are enforced exactly in the path chosen.

4 Related Work

In this section, we discuss related works which focus on the placement on VNFs. The authors of [4] address specification and placement of VNF service chains. They develop a heuristic to specify the VNF service chain and a Mixed Integer Quadratically Constrained Program (MIQCP) for the VNF placement problem. The MIQCP demonstrates the effect of network operator objectives on VNF palcement. In [5], the authors model the VNF placement and routing scenario as a Mixed Integer Linear Program (MILP) to place services optimally for flows and minimize network resource consumption, and they develop heuristics to place services optimally for a large number of flows. The authors of [6] also solve the problem of VNF service chain placement using an MILP and give insights into trade-offs between legacy and NFV-based Traffic Engineering. In [7], the authors solve the problem of determining the number of VNFs required and their placement to optimize operational expenses while adhering to service level agreements using an Integer Linear Program (ILP). Heuristics based on dynamic programming are used to solve larger instances of the problem.

Besides the placement of VNFs in a network, there also exist other challenges in NFV realization, some of which are being addressed in the works that follow. In [8], the authors design an elaborate management system for managing VNFs, cloud infrastructure, and traditional network orchestration (BSS/OSS), describing in detail the function of each logical entity in the framework. They develop a prototype based on the framework and provide results showing the effectiveness of their system. The authors of [9] also design and develop an integrated architecture intended for deploying VNFs, not only for the operator itself but also to offer VNFs as value-added services to subscribers. To make the VNF service chains more resilient, the authors of [10] have designed a model to describe the resource required in a service chain, and a management system that translates these requirements to a deployment. The authors of [11] give a formal model for the complex VNF scheduling problem where the VNF required for a service chain needs to be scheduled in the same time slot bounded by other constraints.

[2]Eq.(8) can be linearly represented as below.

$$j^{(f_1,f_2)}_{p,(s,d),(u,v)} \leq q^{f_1,u}_{p,(s,d)} \qquad \forall (s,d) \in \Psi_{s,d}, \quad \forall p \in K_{s,d}, \quad \forall (u,v) \in S_{u,v}, \quad \forall (f_1,f_2) \in \Gamma \mid (f_1 -> f_2) \in \Pi \tag{15}$$

$$j^{(f_1,f_2)}_{p,(s,d),(u,v)} \leq q^{f_2,v}_{p,(s,d)} \qquad \forall (s,d) \in \Psi_{s,d}, \quad \forall p \in K_{s,d}, \quad \forall (u,v) \in S_{u,v}, \quad \forall (f_1,f_2) \in \Gamma \mid (f_1 -> f_2) \in \Pi \tag{16}$$

$$j^{(f_1,f_2)}_{p,(s,d),(u,v)} \geq q^{f_1,u}_{p,(s,d)} + q^{f_2,v}_{p,(s,d)} - 1 \qquad \forall (s,d) \in \Psi_{s,d}, \quad \forall p \in K_{s,d}, \quad \forall (u,v) \in S_{u,v}, \quad \forall (f_1,f_2) \in \Gamma \mid (f_1 -> f_2) \in \Pi \tag{17}$$

5 Application

We have utilized this mathematical model for the problem setting in [12]. Please refer to the paper for more details on the application of the model and the corresponding results[3].

Acknowledgment

This work was funded by NSF Grant No. CNS-1217978.

References

[1] ETSI, "Network functions virtualisation: Introductory white paper," 2012.

[2] IETF, "Network service chaining problem statement," 2013.

[3] Cisco, "Cisco Cloud Services Router 1000V 3.14 Series Data Sheet," 2015.

[4] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," *arXiv preprint arXiv:1406.1058*, 2014.

[5] A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. K. Ramakrishnan, and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," *Technical Report, The George Washington University*, 2015.

[6] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," *https://hal.inria.fr/hal-01170042/*, 2015.

[7] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions in NFV," *Computing Research Repository*, vol. abs/1503.06377, 2015.

[3]IEEE Network Magazine style constraints do not allow us to present the ILP in detail. So, we have included it here for reader's of the paper.

[8] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, “The dynamic placement of virtual network functions,” in *Proc. Network Operations and Management Symposium (NOMS)*, IEEE, 2014.

[9] G. Xilouris, E. Trouva, F. Lobillo, J.M. Soares, J. Carapinha, M.J. McGrath, G. Gardikis, P Paglierani, E Pallis, L Zuccaro, *et al.*, “T-nova: A marketplace for virtualized network functions,” in *Proc. European Conference on Networks and Communications (EuCNC)*, IEEE, 2014.

[10] M. Scholler, M. Stiemerling, A. Ripke, and R. Bless, “Resilient deployment of virtual network functions,” in *Proc. 5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, IEEE, 2013.

[11] J.F. Riera, X. Hesselbach, E. Escalona, J.A. Garcia-Espin, and E. Grasa, “On the complex scheduling formulation of virtual network functions over optical networks,” in *Proc. 16th International Conference on Transparent Optical Networks (ICTON)*, IEEE, 2014.

[12] A. Gupta, M. Habib, U. Mandal, P. Chowdhury, M. Tornatore, and B. Mukherjee, “On Service-Chaining Strategies using Virtual Network Functions in Operator Networks,” *Submitted to IEEE Network Magazine*, 2015.