POLITECNICO DI MILANO

Bonsai Research Group

Impact of Processing Resource Sharing on Service Chain Placement in Network Functions Virtualization

Marco Savi, Massimo Tornatore, Giacomo Verticale

***Politecnico di Milano**, Milan, Italy*
Department of Electronics, Information and Bioengineering (DEIB)

Italian Networking Workshop 2016, San Candido (BZ) – 13-15 Jan. 2016

Outline

- Introduction
- System model
- MILP optimization problem
- Numerical results
- Conclusion

Introduction

Network Functions Virtualization [1]

- Networks are populated with a huge number of proprietary hardware equipment performing different network functions (*middleboxes*)
  - Finding places to accommodate them is becoming difficult
  - Hardware-based appliances rapidly reach end of life (high costs for the network operators)

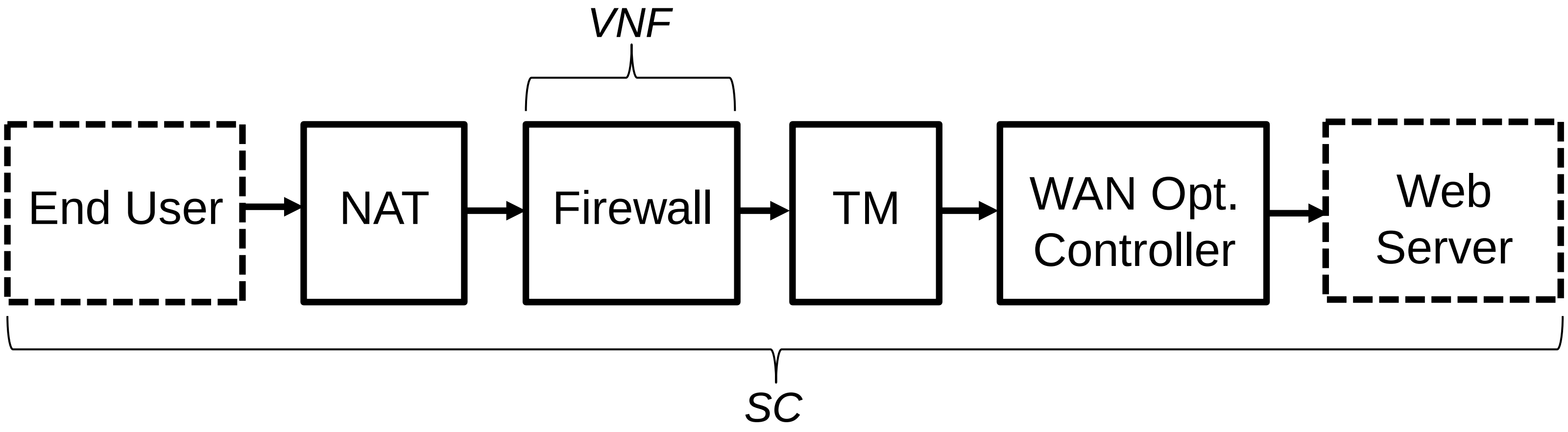
- **Network Functions Virtualization (NFV)** tries to address such issues
  - It leverages standard virtualization tecniques to **consolidate** many network equipment into commercial-off-the shelf (COTS) hardware
    - Network equipment is implemented as **virtual network functions** (VNFs*)* in software
  - The COTS hardware can be located in datacenters, network nodes, customer premises (*NFV nodes*)

[1] *Network Functions Virtualisation, An Introduction, Benefits, Enablers, Challenges & Call for Action, SDN and OpenFlow World Congress, Darmstadt-Germany, 2012*

Introduction

Service Chaining

- The VNFs can be chained together to provide a **service chain (SC)**



- When a *service* is requested between two end-points, one or more SCs must be deployed in the network
  - Different SCs can share the same VNF
  - Different SCs can be shared among different services (e.g., SCs for authentication)

Introduction

Motivation of the work

- NFV adds flexibility to service deployment but it can lead to some drawbacks
  - The consolidation of VNFs leads to performance degradation of the NFV node due to *processing resource sharing*
  - Such performance degradation affects how the VNFs and the SCs are placed in the network
- Related work
  - VNFs and SCs placement in the network considering limited network resources and latency constraints [2]
  - No focus on *processing capacity* of NFV nodes and *processing requirements* of VNFs

[2] *Mehraghdam, S., Keller, M., Karl, H., "Specifying and placing chains of virtual network functions," IEEE 3rd International Conference on Cloud Networking (CloudNet) 2014 , vol., no., pp.7,13, 8-10 Oct. 2014*

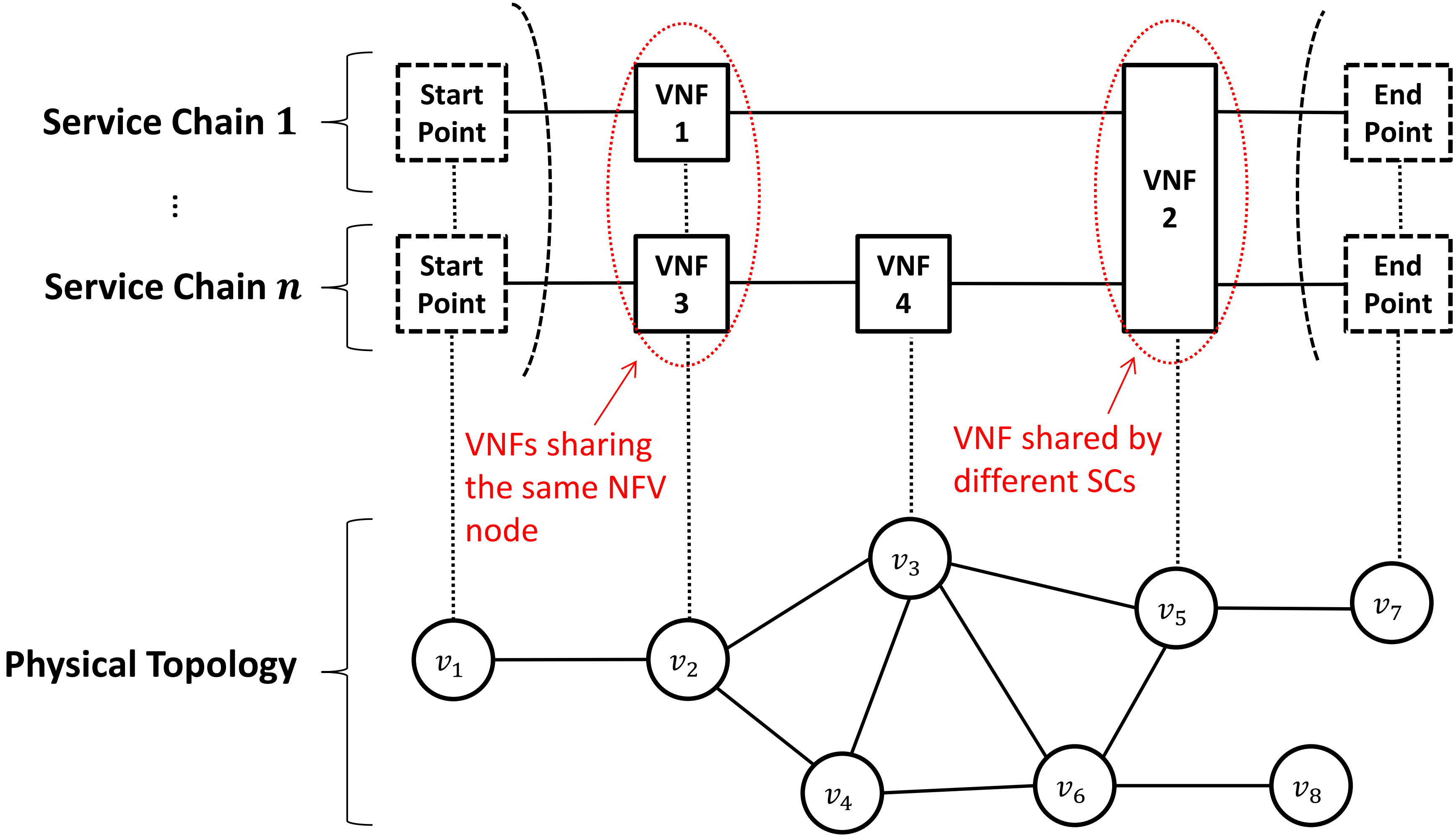
Introduction

Our contribution

- We introduce the concept of *size* of a VNF
  - The more processing resources are assigned to a VNF, the *bigger* is the VNF
  - The bigger is a VNF, the more SCs can share that VNF

- We model two processing resource sharing costs
  - Context switching costs
  - Upscaling costs
- We evaluate how such costs impact on SC and VNF placement

System model
Overview



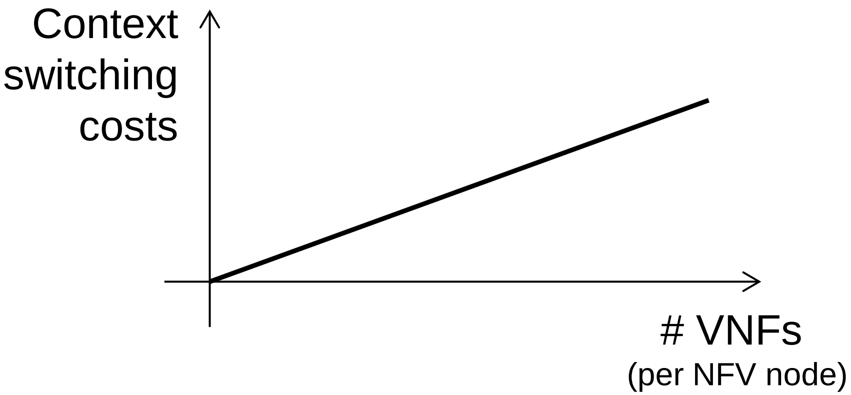
VNFs sharing the same NFV node

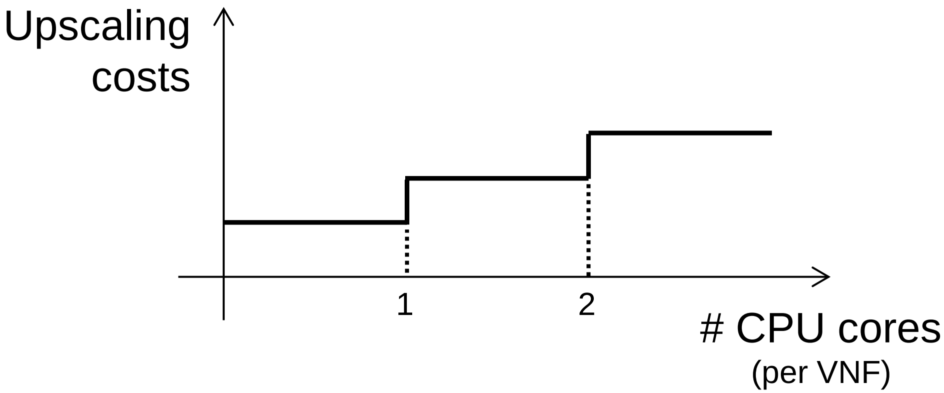
VNF shared by different SCs

System model

Processing resource sharing costs

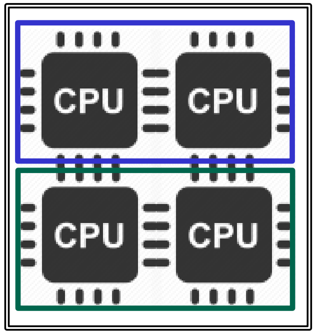
- We assume that *multi-core CPUs* are adopted by NFV nodes
- We consider two processing resource sharing costs
  1. ***Context switching*** *costs*
     - Increase linearly with respect to the *number of* VNFs placed in the NFV node
     - Related to the needs of saving/loading the context (i.e., state) of VNFs
  2. ***Upscaling*** *costs*
     - Step function with respect to the number of CPU cores required by each VNF
     - Related to the needs of balancing traffic among different cores





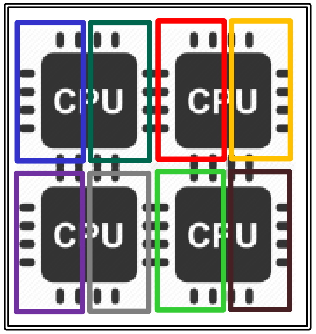
System model
Processing resource sharing costs

- Such costs lead to two performance degradation effects
  1. Increase of *latency* in crossing the NFV node (latency costs)
  2. Decrease of the *actual processing capacity* of the NFV node (processing costs)
- A number/size trade-off for the VNFs sharing a NFV node exists
  - Few big VNFs lead to low context switching costs but high upscaling costs
  - A lot of small VNFs lead to high context switching costs but low upscaling costs



Low context swiching, high upscaling!

NFV node



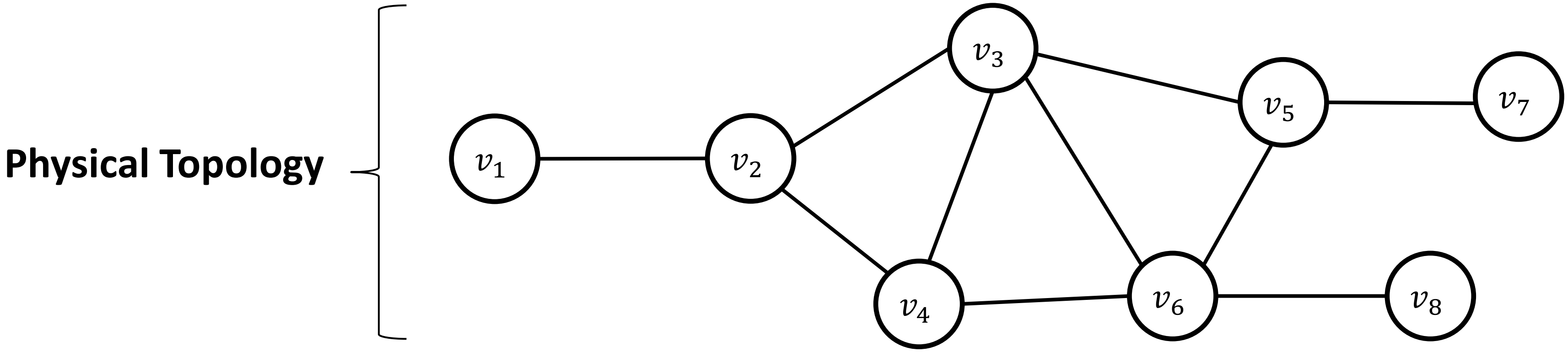
High context swiching, low upscaling!

NFV node

System model

The physical topology

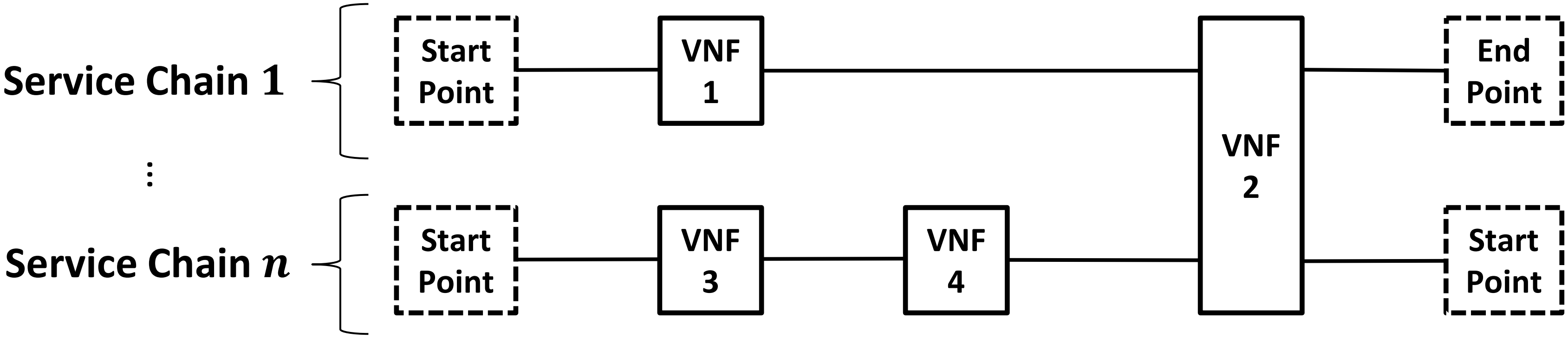
- *Physical nodes*
  - All of them have forwarding capabilities
  - Some of them are *NFV nodes* and are described by
    - The number of CPU cores (processing capacity)
    - The upscaling costs
    - The context switching costs
- *Physical links*: are characterized by
  - Their bandwidth capacity
  - The latency introduced by crossing them



System model

The SCs and the VNFs

- Every *service chain* consists in
  - Two fixed end points
  - A set of chained VNFs
  - A set of virtual links chaining End Points/VNFs
- Every *service chain* is associated to
  - A maximum tolerated latency
  - A requested bandwidth
- Every *VNF* is characterized by its processing requirement



MILP optimization problem

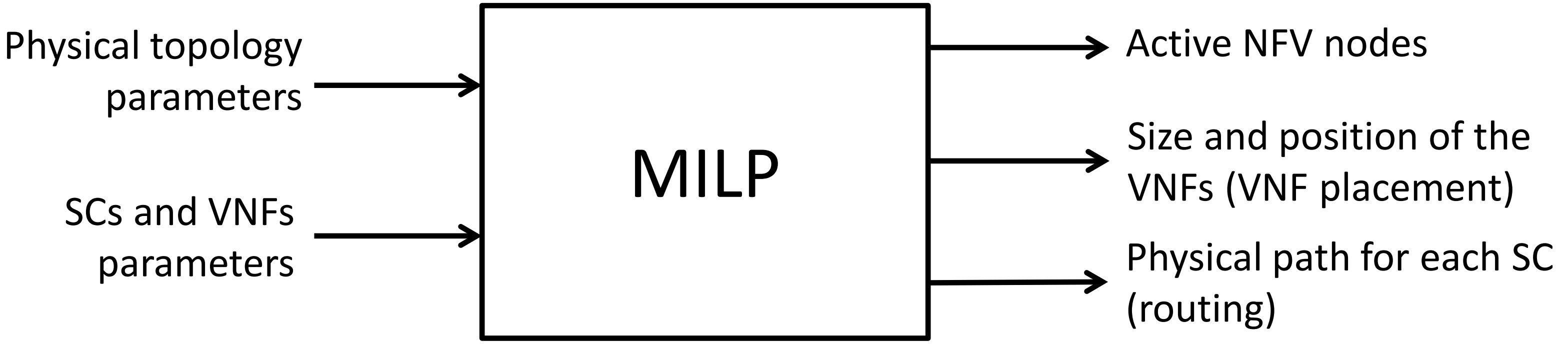
Consolidation of VNFs

- We formulate a Mixed Integer Linear Programming (MILP) model to capture the optimal placement of a set of SCs
- Our **objective** is to *consolidate* as much as possible the deployed VNFs
  - Maximum consolidation ⇔ Minimization of the number of *active* NFV nodes

MILP optimization problem

Inputs and outputs

- Our MILP model decides
  - *Where* the VNFs are placed
  - *What* is the *size* of the placed VNFs
  - *How* the traffic between VNFs is routed in the physical network for each SC



Physical topology parameters

SCs and VNFs parameters

MILP

Active NFV nodes

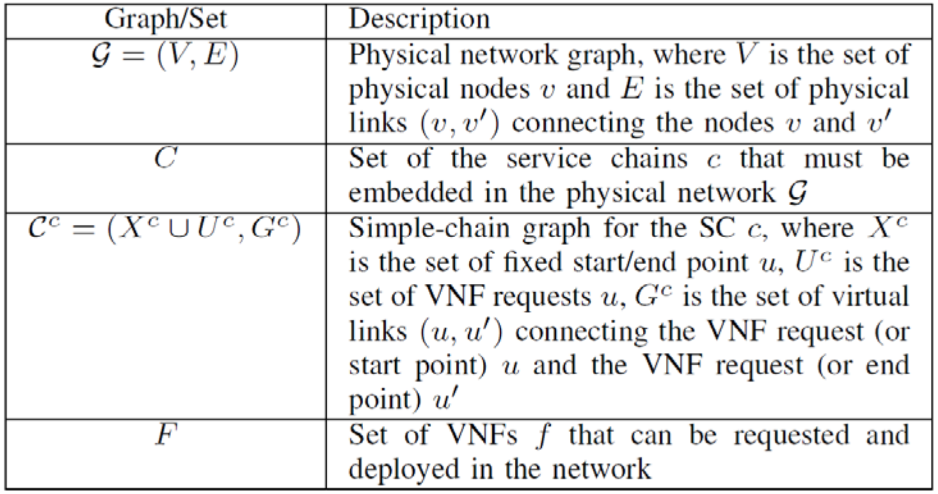
Size and position of the VNFs (VNF placement)

Physical path for each SC (routing)

MILP optimization problem

Sets and parameters

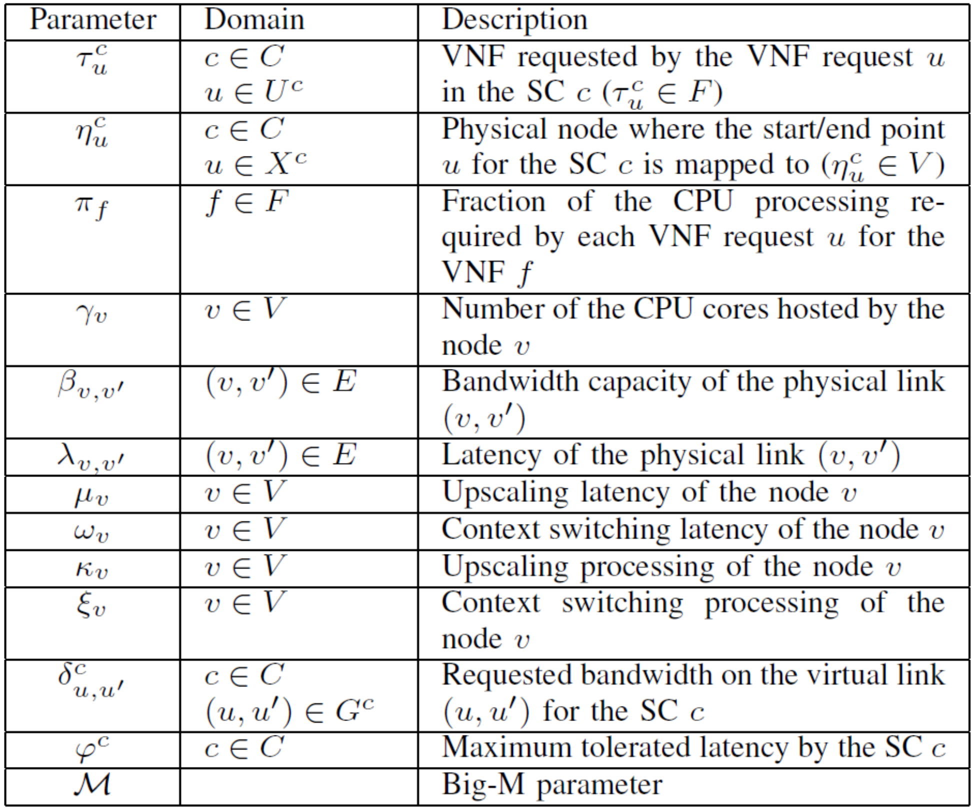
Sets and topologies



Graph/Set	Description
$\mathcal{G} = (V, E)$	Physical network graph, where V is the set of physical nodes v and E is the set of physical links (v, v') connecting the nodes v and v'
C	Set of the service chains c that must be embedded in the physical network $\mathcal{G}$
$\mathcal{C}^c = (X^c \cup U^c, G^c)$	Simple-chain graph for the SC c, where X^c is the set of fixed start/end point u, U^c is the set of VNF requests u, G^c is the set of virtual links (u, u') connecting the VNF request (or start point) u and the VNF request (or end point) u'
F	Set of VNFs f that can be requested and deployed in the network

N.B.: We decouple the concepts of *VNF* f and of *VNF request* u

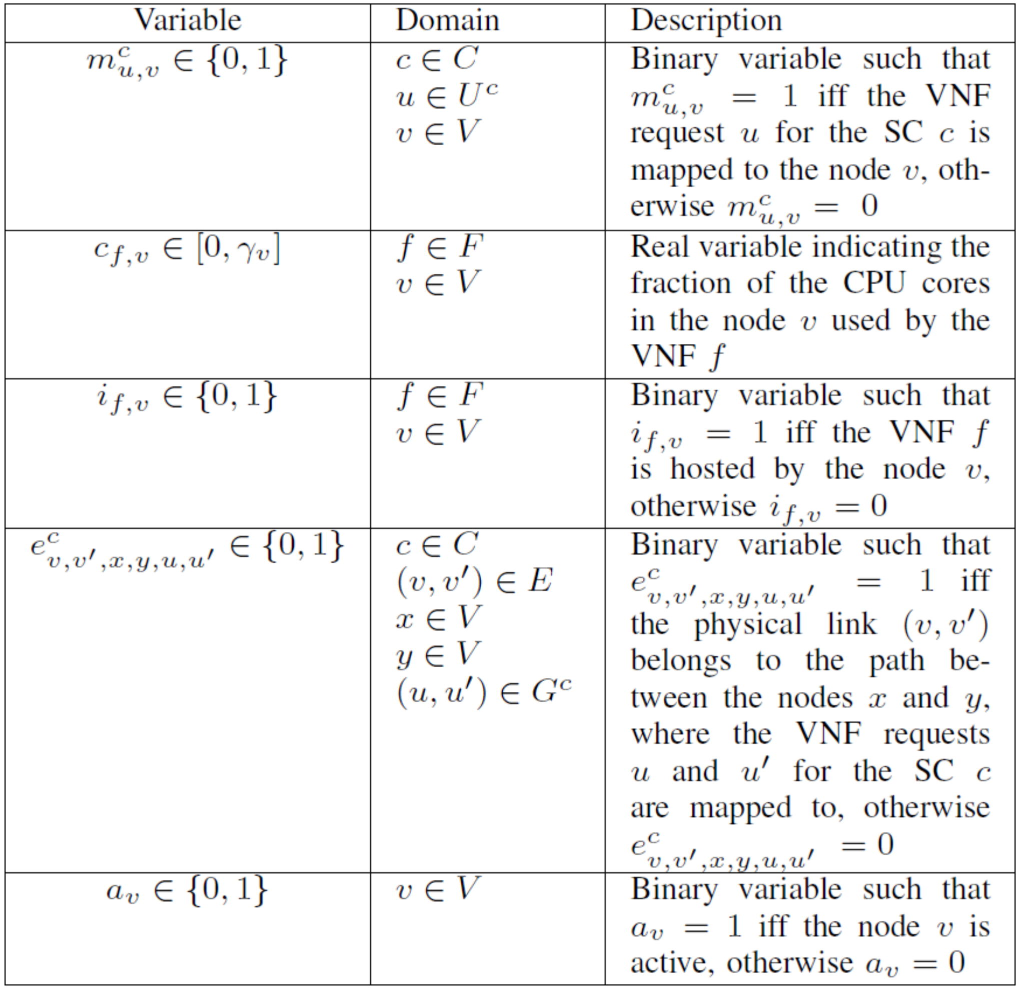
Parameters



Parameter	Domain	Description
τ_u^c	$c \in C$ $u \in U^c$	VNF requested by the VNF request u in the SC c ($\tau_u^c \in F$)
η_u^c	$c \in C$ $u \in X^c$	Physical node where the start/end point u for the SC c is mapped to ($\eta_u^c \in V$)
π_f	$f \in F$	Fraction of the CPU processing required by each VNF request u for the VNF f
γ_v	$v \in V$	Number of the CPU cores hosted by the node v
$\beta_{v,v'}$	$(v, v') \in E$	Bandwidth capacity of the physical link (v, v')
$\lambda_{v,v'}$	$(v, v') \in E$	Latency of the physical link (v, v')
μ_v	$v \in V$	Upscaling latency of the node v
ω_v	$v \in V$	Context switching latency of the node v
κ_v	$v \in V$	Upscaling processing of the node v
ξ_v	$v \in V$	Context switching processing of the node v
$\delta_{u,u'}^c$	$c \in C$ $(u, u') \in G^c$	Requested bandwidth on the virtual link (u, u') for the SC c
φ^c	$c \in C$	Maximum tolerated latency by the SC c
$\mathcal{M}$		Big-M parameter

MILP optimization problem

Decision variables



Variable	Domain	Description
$m^c_{u,v} \in \{0,1\}$	$c \in C$ $u \in U^c$ $v \in V$	Binary variable such that $m^c_{u,v} = 1$ iff the VNF request u for the SC c is mapped to the node v, otherwise $m^c_{u,v} = 0$
$c_{f,v} \in [0, \gamma_v]$	$f \in F$ $v \in V$	Real variable indicating the fraction of the CPU cores in the node v used by the VNF f
$i_{f,v} \in \{0,1\}$	$f \in F$ $v \in V$	Binary variable such that $i_{f,v} = 1$ iff the VNF f is hosted by the node v, otherwise $i_{f,v} = 0$
$e^c_{v,v',x,y,u,u'} \in \{0,1\}$	$c \in C$ $(v,v') \in E$ $x \in V$ $y \in V$ $(u,u') \in G^c$	Binary variable such that $e^c_{v,v',x,y,u,u'} = 1$ iff the physical link (v,v') belongs to the path between the nodes x and y, where the VNF requests u and u' for the SC c are mapped to, otherwise $e^c_{v,v',x,y,u,u'} = 0$
$a_v \in \{0,1\}$	$v \in V$	Binary variable such that $a_v = 1$ iff the node v is active, otherwise $a_v = 0$

VNF requests u → NFV nodes v mapping variable

VNFs f → NFV nodes v mapping variables

Virtual links (u, u') → Physical paths $\sum(v, v')$ mapping variable

MILP optimization problem

Objective function and contraints

- **Objective function**

$$\min \sum_{v \in V} a_v$$

  - The objective of the optimization problem is to minimize the number of NFV active nodes in the network

- The **constraints** are grouped in three different categories
  - *Request placement* constraints: correct mapping VNFs → NFV nodes and VNF requests → NFV nodes
  - *Routing* constraints: correct mapping Virtual links → Physical paths
  - *Performance* constraints: guarantee of the performance requirements

MILP optimization problem
Constraints

- Request placement constraints

$$m^c_{u,\eta^c_u} = 1 \quad c \in C, u \in X^c \tag{2}$$

$$m^c_{u,v} = 0 \quad c \in C, u \in X^c, v \in V : v \neq \eta^c_u \tag{3}$$

Mapping of the SC start/end points

$$\sum_{v \in V} m^c_{u,v} = 1 \quad c \in C, u \in U^c \tag{4}$$

$$\sum_{\substack{c \in C \\ u \in U^c : \tau^c_u = f}} m^c_{u,v} \leq \frac{c_{f,v}}{\pi_f} \quad f \in F, v \in V \tag{5}$$

$$c_{f,v} \leq \mathcal{M} \cdot i_{f,v} \quad f \in F, v \in V \tag{6}$$

$$i_{f,v} - c_{f,v} < 1 \quad f \in F, v \in V \tag{7}$$

$$i_{f,v} \leq \sum_{\substack{c \in C \\ u \in U^c : \tau^c_u = f}} m^c_{u,v} \quad f \in F, v \in V \tag{8}$$

$$\sum_{f \in F} c_{f,v} \leq \gamma_v - \psi_v \quad v \in V \tag{9}$$

Mapping of VNF requests and VNFs on the NFV nodes

$$\psi_v = \sum_{\substack{c \in C,\ f \in F \\ u \in U^c : \tau^c_u = f}} m^c_{u,v} \cdot \left(\lceil c_{f,v} \rceil \cdot \kappa_v + \sum_{g \in F} i_{g,v} \cdot \xi_v \right) \quad v \in V \tag{10}$$

Upscaling and context switching processing costs

MILP optimization problem

Constraints

- Routing constraints

Solenoidality constraints

$$e^c_{v,v',x,y,u,u'} \le m^c_{u,x} \cdot m^c_{u',y}$$
$$c \in C, (v,v') \in E, x \in V, y \in V, (u,u') \in G^c \qquad (11)$$

$$\sum_{(x,v)\in E,\ x,y \in V} e^c_{x,v,x,y,u,u'} \cdot m^c_{u,x} \cdot m^c_{u',y} = 1 \quad c \in C, (u,u') \in G^c \qquad (12)$$

$$\sum_{(v,y)\in E,\ x,y \in V} e^c_{v,y,x,y,u,u'} \cdot m^c_{u,x} \cdot m^c_{u',y} = 1 \quad c \in C, (u,u') \in G^c \qquad (13)$$

$$\sum_{(v,x)\in E,\ v \in V} e^c_{v,x,x,y,u,u'} = 0$$
$$c \in C, x \in V, y \in V, x \neq y, (u,u') \in G^c \qquad (14)$$

$$\sum_{(y,v)\in E,\ v \in V} e^c_{y,v,x,y,u,u'} = 0$$
$$c \in C, x \in V, y \in V, x \neq y, (u,u') \in G^c \qquad (15)$$

$$\sum_{(v,w)\in E,\ v \in V} e^c_{v,w,x,y,u,u'} = \sum_{(w,v')\in E,\ v' \in V} e^c_{w,v',x,y,u,u'}$$
$$c \in C, w \in V, x \in V, y \in V, x \neq w, y \neq w, (u,u') \in G^c \qquad (16)$$

$$\sum_{(v,w)\in E,\ v \in V} e^c_{v,w,x,y,u,u'} \le 1$$
$$c \in C, w \in V, x \in V, y \in V, x \neq w, y \neq w, (u,u') \in G^c \qquad (17)$$

$$e^c_{v,v,x,y,u,u'} = 0$$
$$c \in C, v \in V, x \in V, y \in V, x \neq y, (u,u') \in G^c \qquad (18)$$

$$e^c_{v,v',x,x,u,u'} = 0$$
$$c \in C, x \in V, (u,u') \in G^c, (v,v') \in E, v \neq v' \qquad (19)$$

Usage of self-loops in the physical topology

MILP optimization problem

Constraints

- Performance constraints

$$\sum_{\substack{c\in C,\ x,y\in V\\(u,u')\in G^c}} e^c_{v,v',x,y,u,u'} \cdot \delta^c_{u,u'} \le \beta_{v,v'} \quad (v,v')\in E \qquad (20)$$

Bandwidth constraint for the links

$$\sum_{\substack{(v,v')\in E,\ x,y\in V\\(u,u')\in G^c}} e^c_{v,v',x,y,u,u'} \cdot l_{v,v'} + \sigma^c \le \varphi^c \quad c\in C \qquad (21)$$

Latency constraint for the Service Chains

$$\sigma^c = \sum_{\substack{f\in F,\ v\in V\\u\in U^c:\tau^c_u=f}} m^c_{u,v} \cdot \left(\lceil c_{f,v}\rceil \cdot \mu_v + \sum_{g\in F} i_{g,v}\cdot \omega_v\right) \quad c\in C \qquad (22)$$

Upscaling and context switching latency costs

$$\sum_{f\in F} i_{f,v} \le \mathcal{M}\cdot a_v \quad v\in V \qquad (23)$$

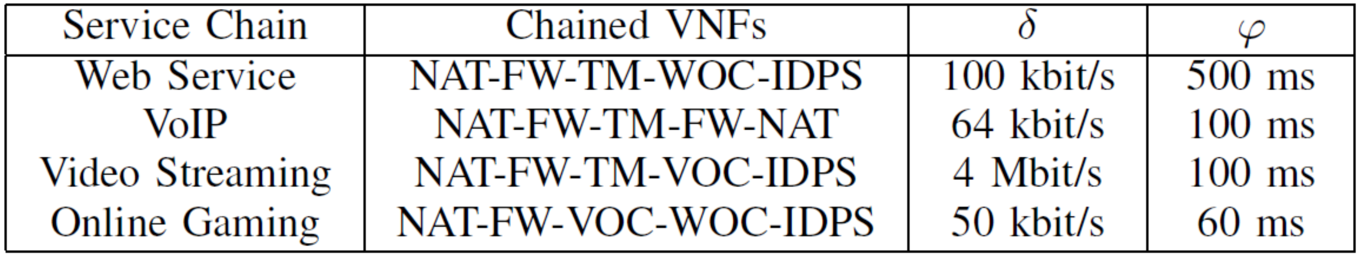
$$a_v \le \sum_{f\in F} i_{f,v} \quad v\in V \qquad (24)$$

Definition of the NFV active nodes

Numerical results

Simulation settings

- We consider the following simulation settings
    - Physical topology: Internet2 network (10 nodes)
    - Service Chains: four different types



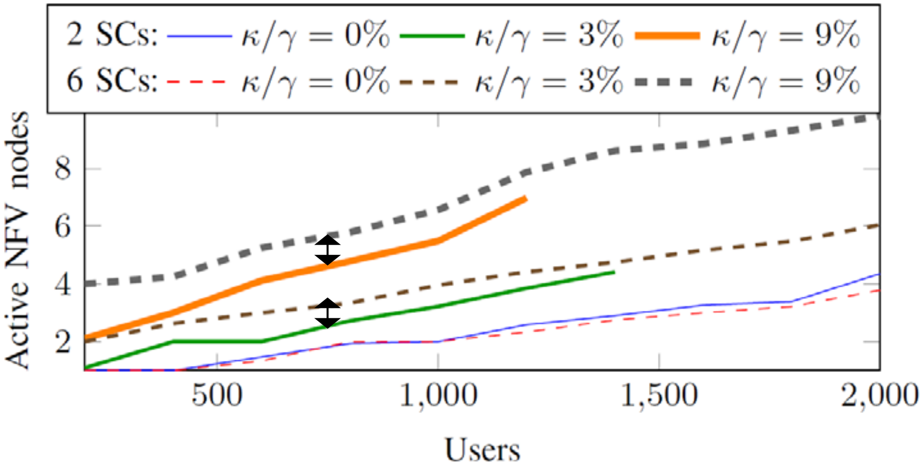
Service Chain	Chained VNFs	δ	φ
Web Service	NAT-FW-TM-WOC-IDPS	100 kbit/s	500 ms
VoIP	NAT-FW-TM-FW-NAT	64 kbit/s	100 ms
Video Streaming	NAT-FW-TM-VOC-IDPS	4 Mbit/s	100 ms
Online Gaming	NAT-FW-VOC-WOC-IDPS	50 kbit/s	60 ms

-
    - We chose the CPU processing requirements for each VNF according to some middleboxes datasheets
- We defined two scenarios
    - *Heterogeneous scenario*: we randomize the choice of start/end points and of the SCs to be deployed
    - *Homogeneous scenario*: we randomize the choice of start/end points but we consider only one type of SC

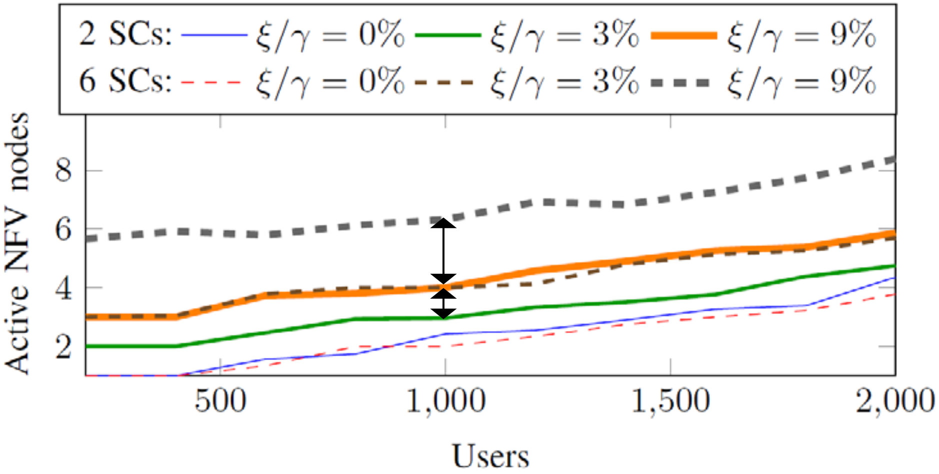
Numerical results

Heterogeneous scenario

Upscaling costs (processing)



Context switching costs (processing)

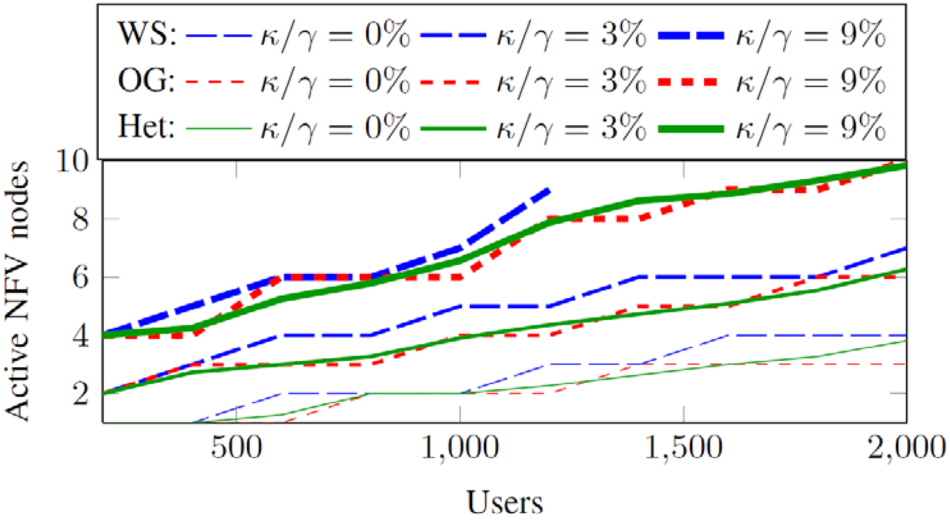


- As the number of users increases, the number of NFV active nodes increases
- The number of SCs does not significantly affect how higher upscaling costs translate into the number of active NFV nodes
- As the number of deployed SCs grows, the impact of higher context switching costs on the number of active NFV nodes is amplified

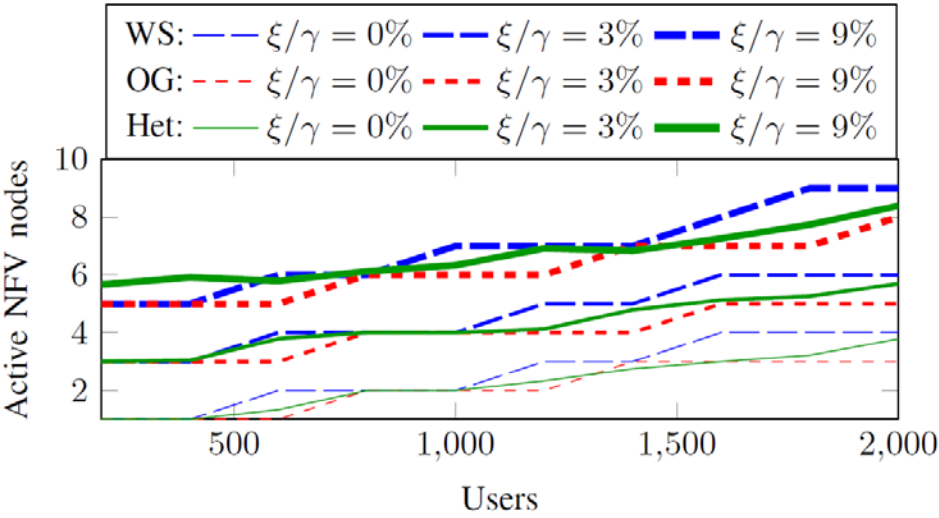
Numerical results

Homogeneous scenario (6 SCs)

Upscaling costs (processing)



Context switching costs (processing)



- We consider the deployment of Web Service (*WS*) and Online Gaming (*OG*) homogeneous SCs
- The deployment of homogeneous types of SCs does not significantly impact on the number of active NFV nodes, even with respect to the heterogeneous (*Het*) case

Conclusion

- We investigated the impact of processing resource sharing among VNFs when multiple SCs are placed in the network
- We took into account the upscaling and the context switching costs
  - Higher context switching costs significantly amplify the number of active NFV nodes when an increasing number of SCs is considered, while the upscaling costs don't
  - The deployment of homogeneous and heterogeneous SCs has a similar impact on the VNF consolidation

- Current work
  - Development of a heuristic algorithm to improve the scalability of the model

THANK YOU!

Questions?