網 路 功 能 虛 擬 化
Network Functions Virtualization

Fang-Tzu Hsu

2018/10/09

Outline

01 Introduction

02 Framework

03 NFV-resource allocation

04 NFV v.s. SDN

05 Joint Optimization of Virtual Function Migration and Rule Update in Software Defined NFV Networks

Introduction

- 網路功能虛擬化 (Network Functions Virtualization, NFV)
  - 將以往必須建基於硬件形式的各種技術虛擬化
    - 負載平衡(Load Balance)、防火牆(Firewall)、IPS（Intrusion Prevention System）
  - 減少硬件成本，包括部署、硬件使用的電力、數據中心空間
  - 一種可程式化控制的新型網路基礎配置，將原本的專屬硬體與功能，改為軟體功能，建置在通用高容量伺服器上
    - 企業標準
    - 多樣性網路服務

Introduction

	傳統網路	NFV
作法	• 分散式的架構 • 專屬功能有專屬硬體機器	• 集中式架構 • 利用虛擬化平台提供專屬機器的網路功能
優點	• 從硬體設計來達成網路傳遞訊息的行為，並且提升交易的處理速度	• 從虛擬化平台來構成網路佈建網路傳遞處理伺服器，透過虛擬化平台彈性調整效能。
缺點	• 當需要改變訊息傳遞行為時必須逐一進行設定或者更換原來的設備，如此將耗費非常多的人力和成本	• 初期軟硬體設備投資高，並受限於虛擬化平台的基礎硬體環境

Introduction

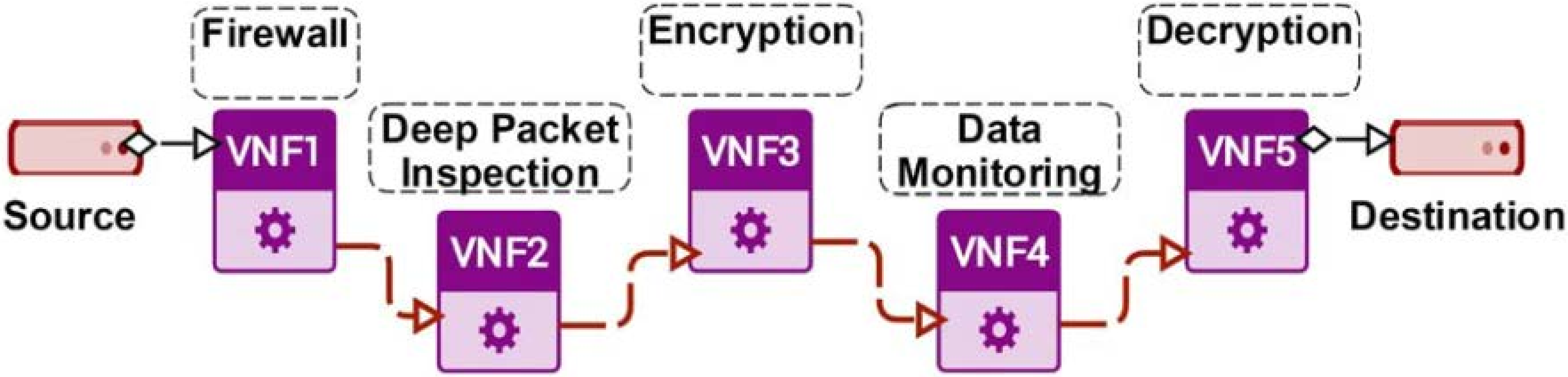
- GS(ETSI Group Specification) :
  - NFV應用案例(Use Cases)
    - https://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf
  - NFV術語文件(Terminology for Main Concepts in NFV)
    - https://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.03.01_60/gs_nfv003v010301p.pdf
  - NFV需求文件(Virtualization Requirements)
    - https://www.etsi.org/deliver/etsi_gs/NFV/001_099/004/01.01.01_60/gs_NFV004v010101p.pdf
  - NFV框架(Architectural Framework)
    - https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

Introduction

- Under the paradigm of NFV, traditional middleboxes are managed as single modules of software, programmed to play the role of a particular Virtual Network Function (VNF)
  - allows modularity and isolation of each function
  - be managed independently
  - NFV facilitates installation and deployment of VNFs on general purpose servers
  - allowing dynamic migration of VNFs from one server to another, that is, to any place of the network

Introduction

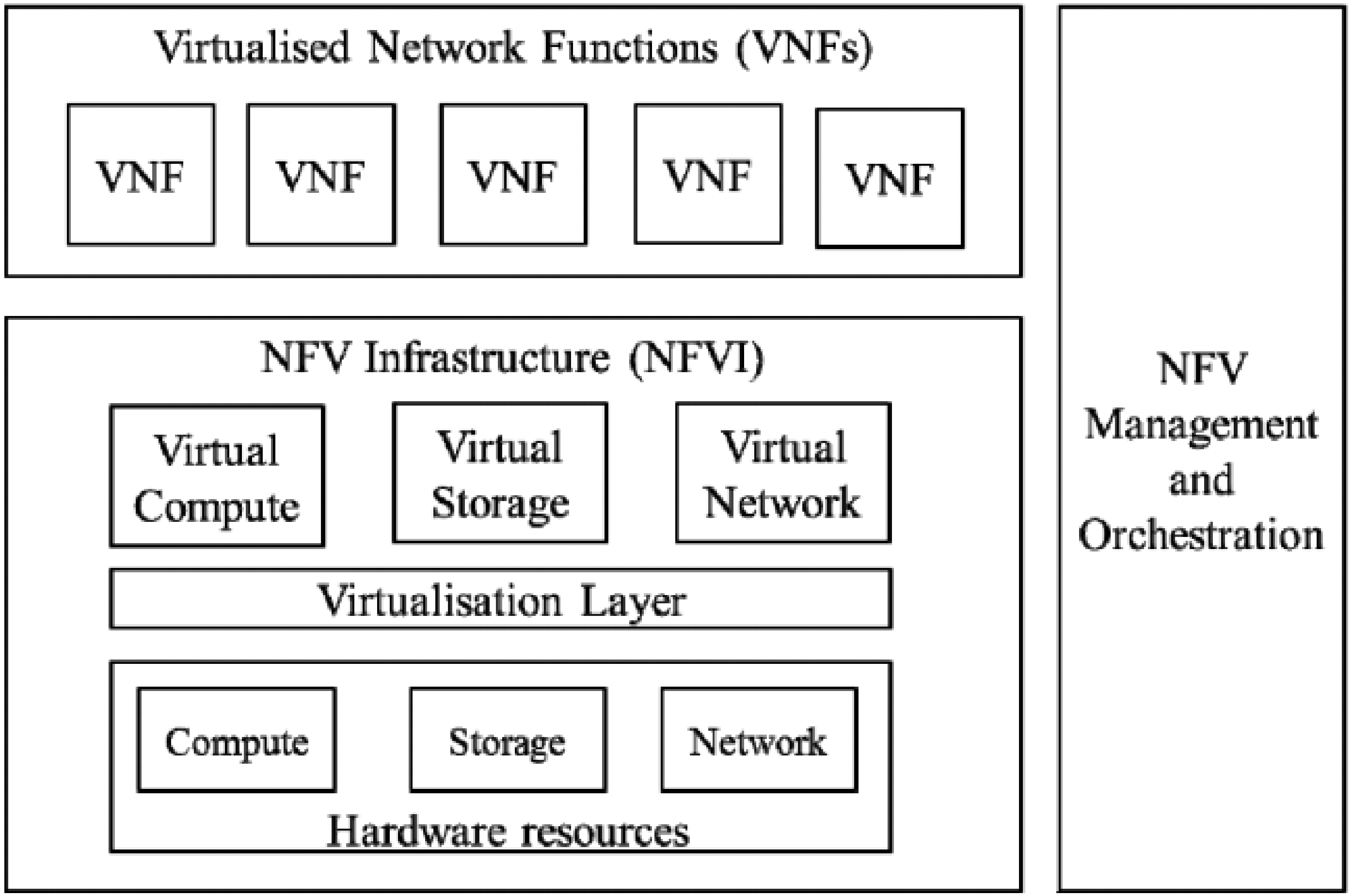
- An NS is built and deployed in NFV by defining its:
  - i) number of VNFs
  - ii) their respective order in the chain
  - iii) the allocation of the chain in the Network Functions Virtualization Infrastructure (NFVI)



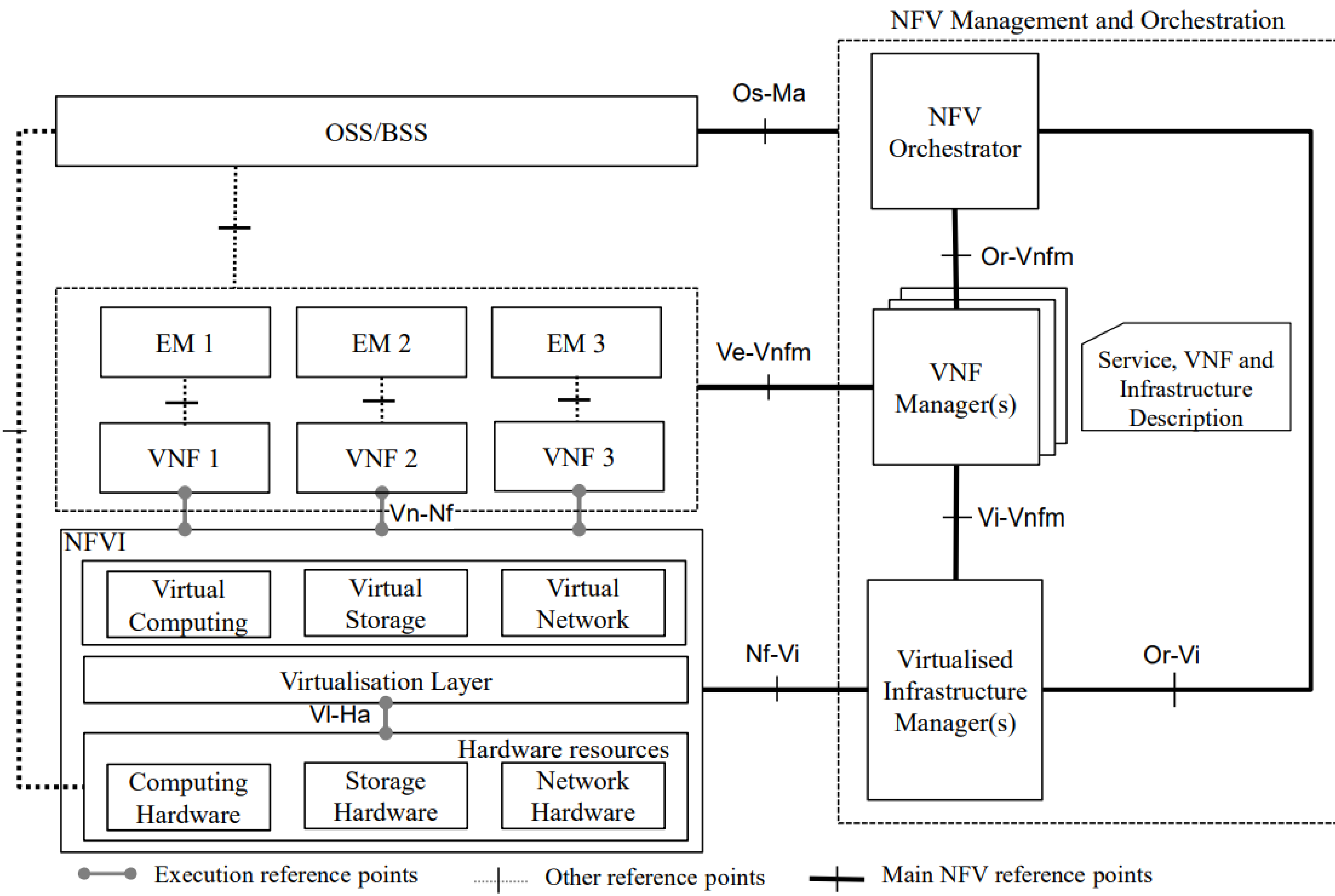
Introduction

- One of the main challenges to deploy NFV is to achieve fast, scalable and dynamic composition and allocation of NFs to execute an NS.
- However, since an NS requires a set of VNFs, achieving an efficient services' coordination and management in NFV raises two questions:
  - 1) how to compose VNFs for a determined NS
  - 2) how to efficiently allocate and schedule the VNFs
- Resource allocation challenge in NFV-based networks.

Framework


Virtualised Network Functions (VNFs)
VNF
VNF
VNF
VNF
VNF
NFV Infrastructure (NFVI)
Virtual Compute
Virtual Storage
Virtual Network
Virtualisation Layer
Compute
Storage
Network
Hardware resources
NFV Management and Orchestration

Framework



NFV Management and Orchestration
OSS/BSS
Os-Ma
NFV Orchestrator
Or-Vnfm
EM 1
EM 2
EM 3
VNF 1
VNF 2
VNF 3
Ve-Vnfm
VNF Manager(s)
Service, VNF and Infrastructure Description
Vn-Nf
Vi-Vnfm
NFVI
Virtual Computing
Virtual Storage
Virtual Network
Virtualisation Layer
Vl-Ha
Hardware resources
Computing Hardware
Storage Hardware
Network Hardware
Nf-Vi
Virtualised Infrastructure Manager(s)
Or-Vi
Execution reference points
Other reference points
Main NFV reference points

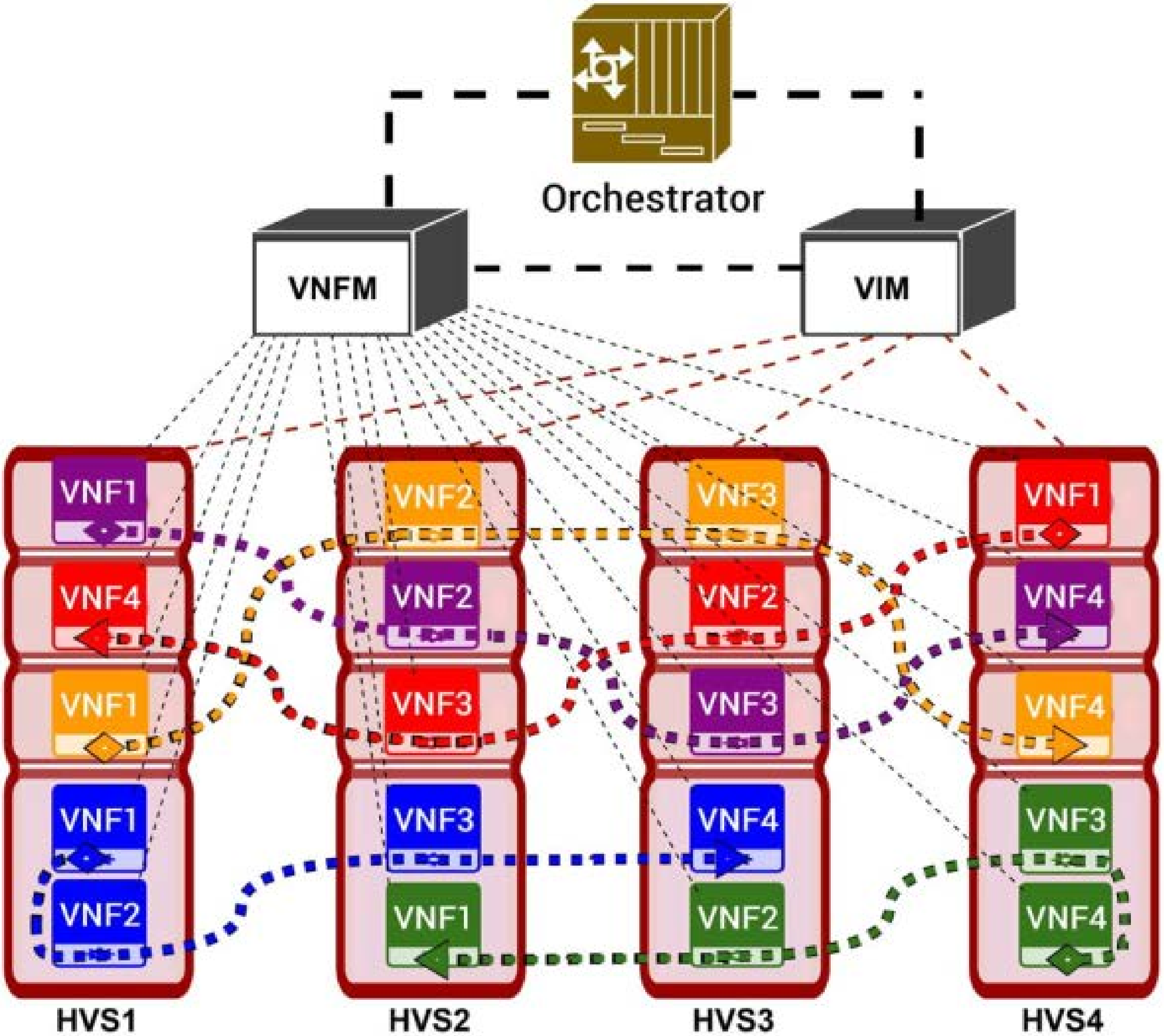


Fig. 4. NFV Management and Orchestration.

NFV-resource allocation

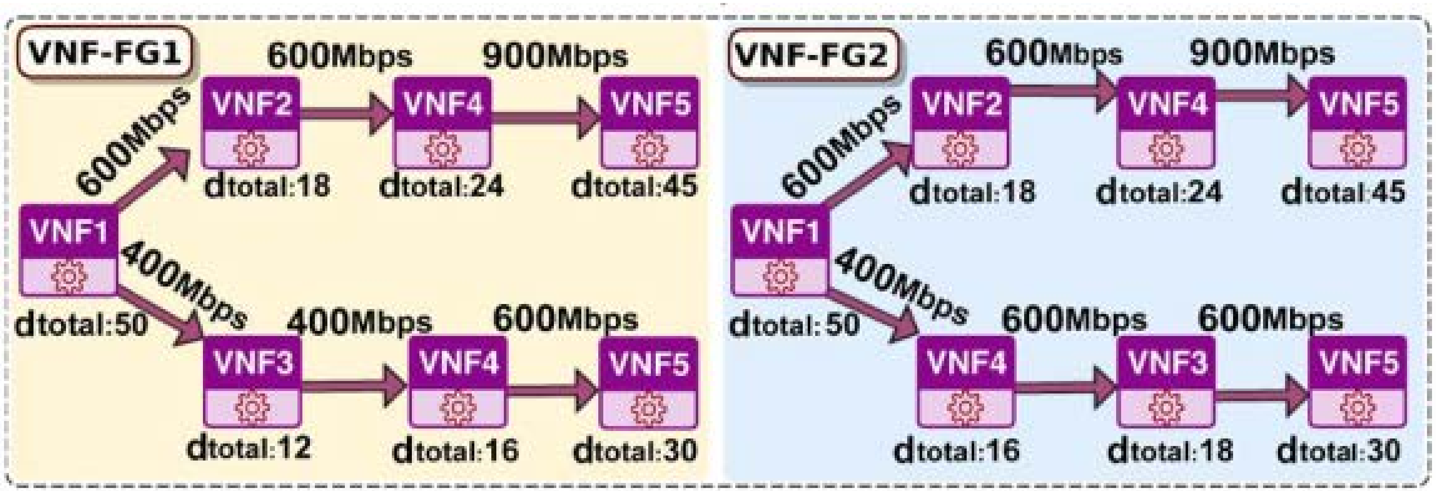
- Virtual Network Embedding (VNE)
  - embedding virtual networks in a SN
  - VNE deals with the allocation of virtual resources both in nodes and links
  - the VNE problem can be either offline or online.
    - all the virtual network requests are known and scheduled in advance or arrive dynamically and stay for an arbitrary duration
  - VNE is known to be NP –hard
    - mixed Integer Linear Programming (ILP) models
    - heuristic or metaheuristic algorithms
    - minimal complexity
  - embedding cost, link bandwidth, QoS, economical profit, network survivability, energy efficiency, security

NFV-resource allocation

- NFV-RA's
  - input is a network service request composed of a set of VNFs with precedence constraints
  - resource demands that can be denoted by several Virtual Network Function Forwarding Graphs (VNF-FGs)
- Resource demands may change depending of the traffic load directed to them :
  - bandwidth demands change depending on the ordering of VNF instances, whereas resource demands are mostly static in VNE

NFV-resource allocation

- NFV-RA
  - 1) VNFs - Chain Composition (VNFs-CC):
    - how to concatenate the different VNFs efficiently in order to compose an NS?
    - deploy customized and dynamic NFV-enabled network services
    - most NFV-RA proposals consider the VNF-FG as an input of the problem



b)

NFV-resource allocation

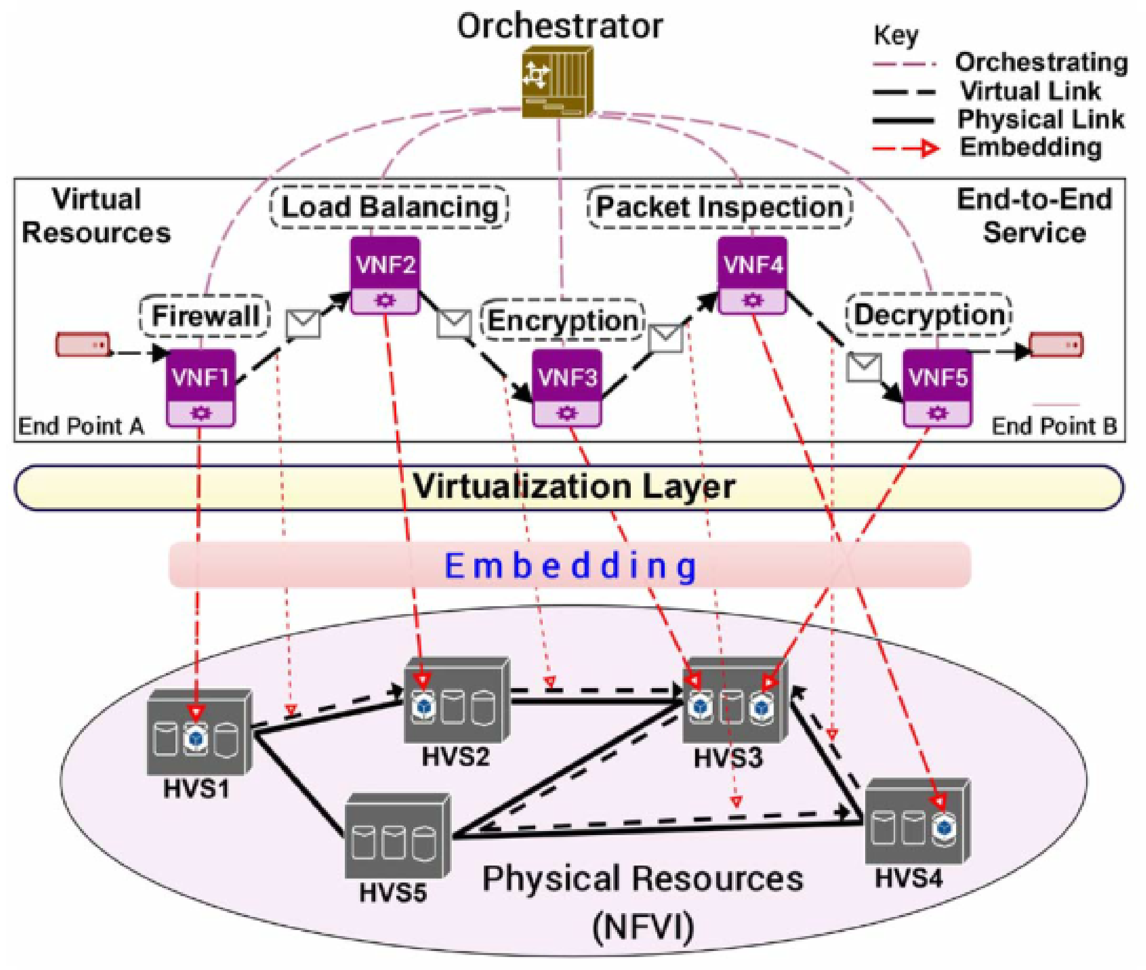
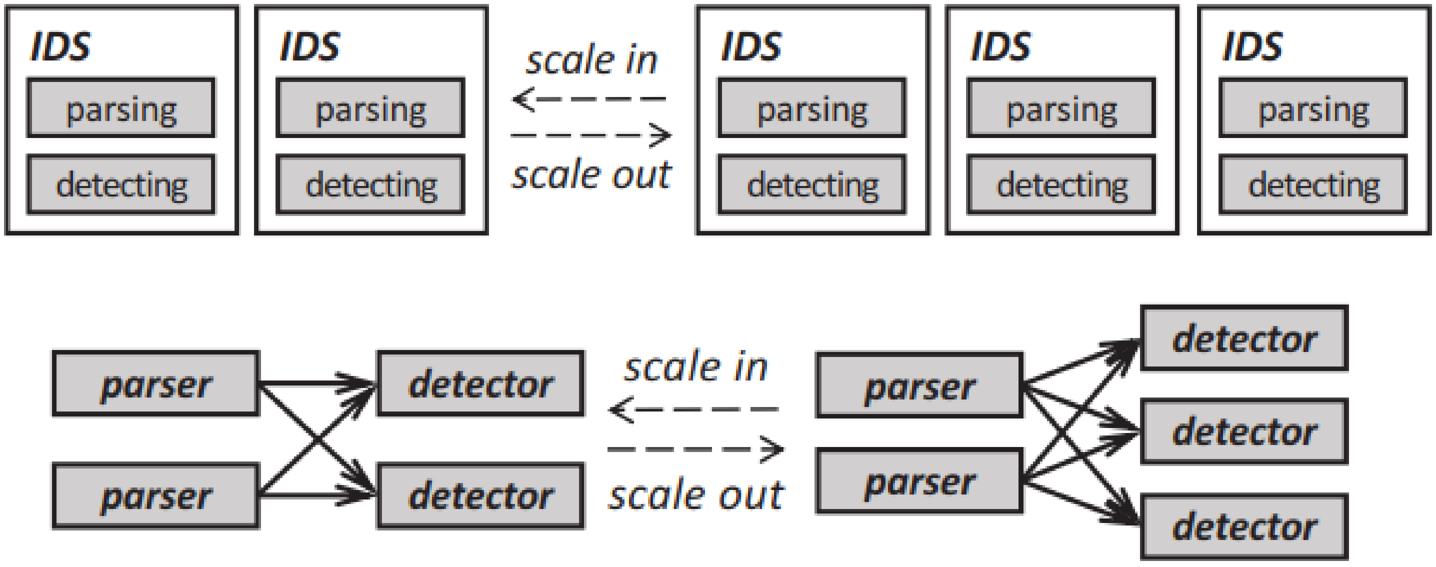


Fig. 6. VNFs forwarding-graph embedding.

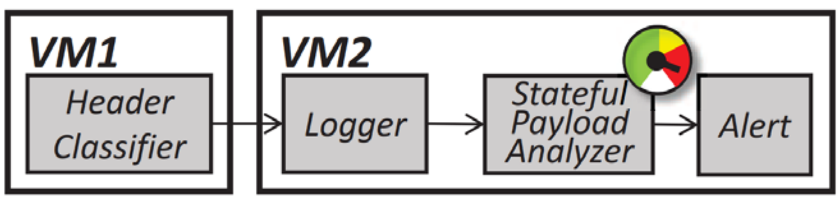
NFV-resource allocation

- A single VNF may be composed of multiple internal components, and hence it could be deployed over multiple Virtual Machines (VMs), in which case each VM hosts a single component of the VNF.

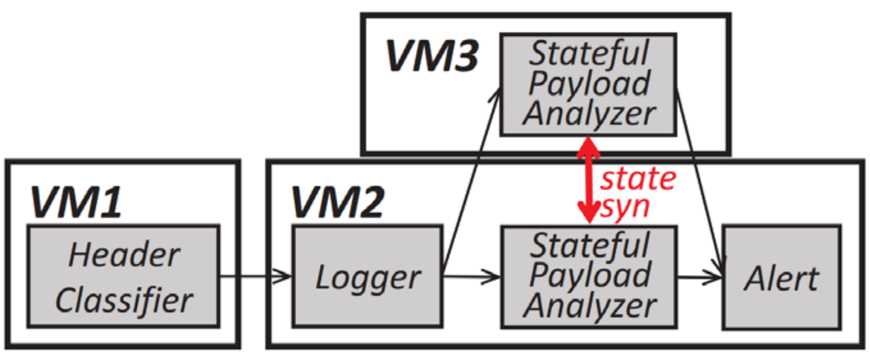


NFV-resource allocation

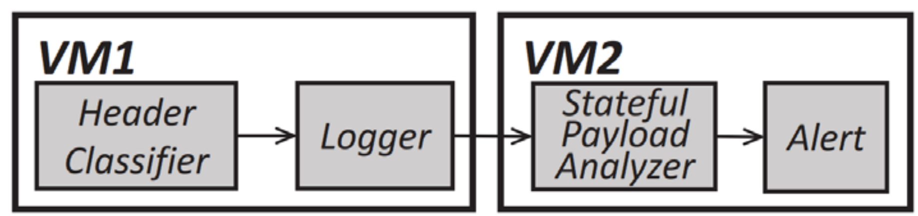
- The orchestrator may trigger the migration of a VNF from a HVS to another if necessary, to rearrange VNFs of several services, in order to optimize the use and allocation of physical resources.



(a) MSFC before scaling



(b) Scaling out with traditional method



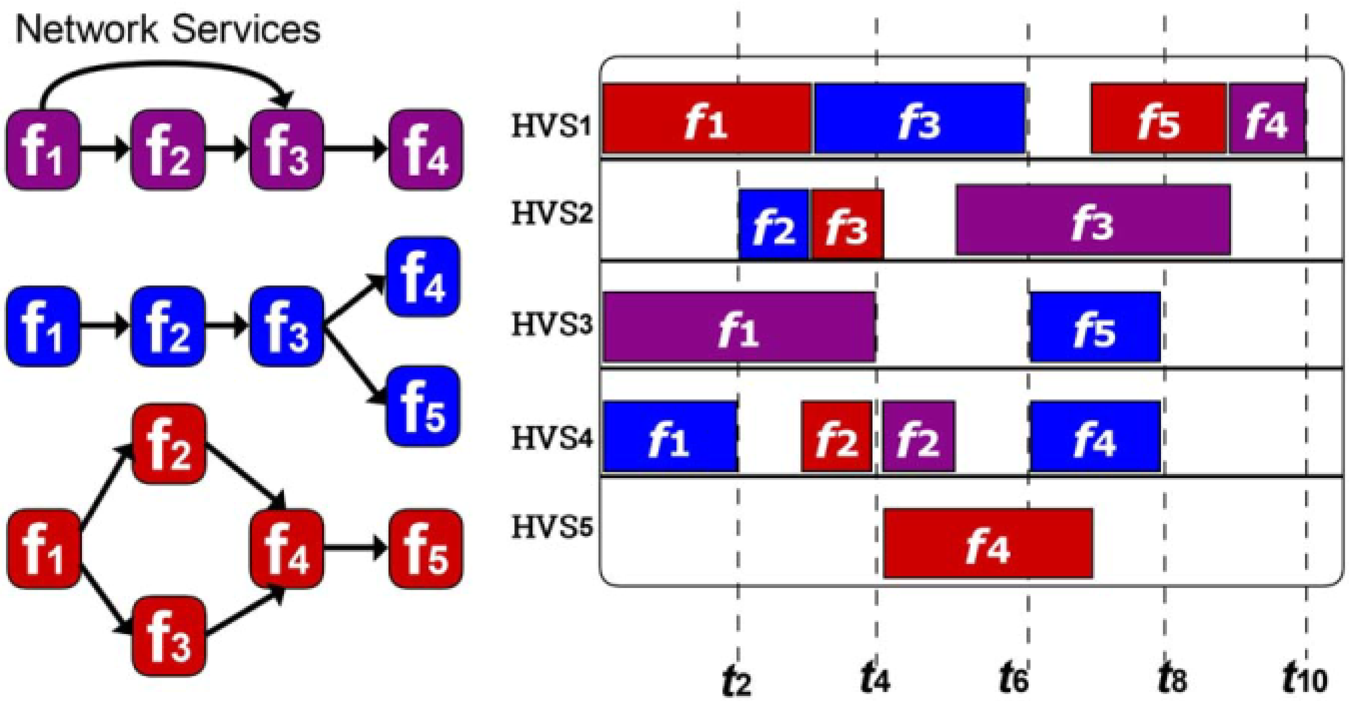
(c) Push-aside scaling up

NFV-resource allocation

- VNFs - Scheduling (VNFs-SCH)
  - The NFV infrastructure is comprised of several and different HVSs, therefore, a proper scheduling of VNFs' execution should be performed in order to minimize the total execution time of the network services, and thus obtain improved performance.
  - The effectiveness, performance, and efficiency of the scheduling process can be defined in terms of:
    - i) number of available HVSs in the NFV infrastructure so they can process the functions composing the services
    - ii) the computing capacity of each server to process all the assigned functions
    - iii) the complexity of the different network services, i.e., the number of functions composing each service.

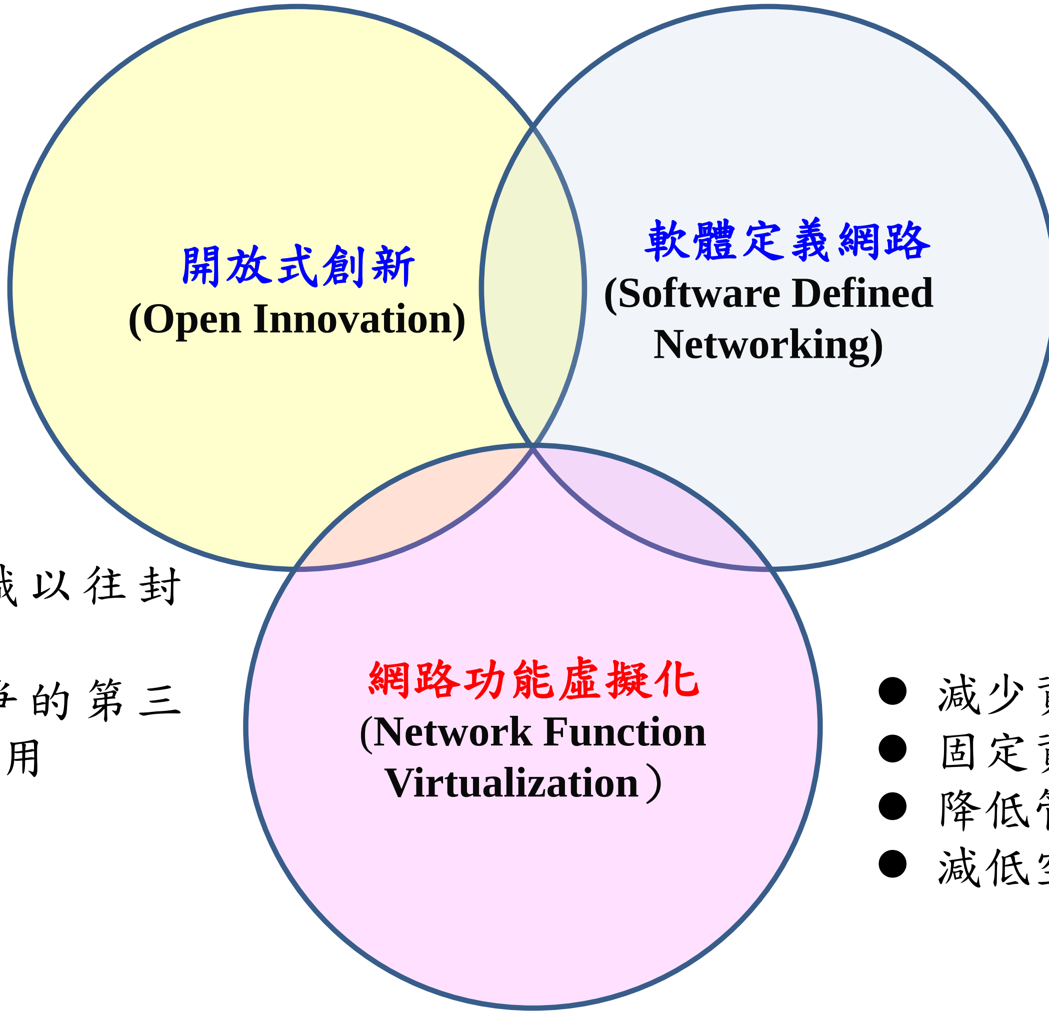
NFV-resource allocation

- VNFs - Scheduling (VNFs-SCH)



8. VNFs Scheduling.

NFV V.S. SDN



- 網路虛擬化
- 加速創新

- 突破組織以往封閉的疆界
- 充滿競爭的第三方創新應用

- 減少資金
- 固定資產的投入
- 降低管理支出
- 減低空間及能源消耗

NFV V.S. SDN

	SDN	NFV
作法	管理網路的控制與傳送封包的轉送分開，讓資訊技術對於網路行為與效能擁有更大的控制能力。	將那些傳統上與特殊化專屬硬體相關聯的網路程序轉換成可以在標準商用硬體執行的虛擬化軟體平台。
效用	集中化控制及可程式設計能解決快速成長和分散化網路在管理上所涉及的複雜性。	網路功能可依需求在網路內移動向上或向下擴展，不會因安裝新硬體裝置，而造成延誤與成本負擔。
好處	允許在軟體和硬體方面做出個別的採購決策。	促成快速且低成本的功能升級，達到快速的服務創新。

Joint Optimization of Virtual Function Migration and Rule Update in Software Defined NFV Networks

J. Zhang, D. Zeng, L. Gu, H. Yao and M. Xiong, GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-5.

Outline

- Introduction
- System model and problem formulation
- Heuristic algorithm design
- Performance evaluations
- Conclusion

Introduction

- To simplify the network management and make efficient use of network resources,
  - Network function virtualization (NFV)
    - run the network functions as software instances on commodity servers
  - Software defined networks (SDNs)
    - decouples the network control plane from the data plane
- During network operation, network updates need to be implemented frequently
  - dynamically adjust the network configuration to meet the predefined QoS
  - Virtual network functions (VNF) may need to be migrated from one server to another to adapt to the network changes.

Introduction

- Study [7], [17], [18] have focused the problem of VNF placement and migration
  - to preserve consistency of NFV state so as to ensure the network service correctness during migration
  - the forwarding rules of flows shall also be updated accordingly to reserve the correctness of network service semantics
- In SDN, the forwarding is determined by per-flow rule.
- During the process of network update, the controller instructs switches to add, change or remove some rules.

[7] J. Liu, Z. Jiang, N. Kato, O. Akashi, and A. Takahara, "Reliability Evaluation for NFV Deployment of Future Mobile Broadband Networks," IEEE Wireless Communications Magazine, vol. 23, no.3, pp. 90-96, Jun. 2016.
[17] A. Gember-Jacobson, R. Viswanathan, C. Prakash, R. Grandl, J. Khalid, S. Das, and A. Akella, "OpenNF: Enabling Innovation in Network Function Control," in ACM SIGCOMM CCR, vol. 44, no. 4, pp. 163–174, 2015.
[18] S. Rajagopalan, D. Williams, H. Jamjoom, and A. Warfield, "Split/Merge: System Support for Elastic Execution in Virtual Middle-boxes.," in Proc. of NSDI, vol. 13, pp. 227–240, 2013.

Introduction

- To avoid service disruption
  - the update strategy must be robust with respect to various factors
    - such as non-deterministic processing time on the switch to install or modify rules
    - transmission latency between the controller and the switches
- All existing studies assume a given goal network.
- In NFV-enabled environment, it is possible to shape the goal network by flexibly migrating the VNFs according to the network conditions towards various objectives.

Introduction

- In this paper, we are motivated to jointly investigate the VNF migration and rule update in software defined NFV networks.
- Migrating a VNF from one server to another migration cost is non-ignorable.
  - Different migration decisions
    - different migration cost
    - different goal network topologies
    - influence the forwarding rule update decisions and the update delay

Introduction

- Our main contributions are summarized as follows:
  - Address the VNF migration and forwarding rule update to balance the migration cost and network update delay.
  - Describe the problem into a mixed integer non-linear optimization programming (MINLP) problem.
    - polynomial-time two-stage heuristic algorithm
  - Through extensive simulations, the high efficiency of our heuristic algorithm performs much close to the optimal solution.

System model and problem formulation

A. System Model

- consider a software defined NFV networks as a graph $G = (V, E)$
  - V is the set of servers and switches
  - E is the set of links between servers
  - Some VNFs have already been deployed on some servers and we denote the set of VNFs placed in server $v \in V$ as F_v.
- We assume that a set F of VNFs need to be migrated out of their current host server.
  - $K = |F|$: the number of to-be-migrated VNFs.
  - B_f : traffic demand of flow through function $f \in F$
  - R_f : the number of flow entries that need to be installed per server to route flow on function f
  - P_f : the old routing path of flow on function f, the new path is denoted by P'_f.

System model and problem formulation

- A path from a source server v_{sf} to a destination server v_{df} is defined as the list of traversed servers $\{v_{sf} = v_0, v_1, v_2, \ldots, v_k = v_{df}\}$
- The paths are loop-free and each server has only one next hop.
  - c_e : the capacity of link $e \in E$
  - q_v : the flow table size of switch $v \in V$
- C_v : the resource capacity on server $v \in V$
- S_f : the resource demand of function $f \in F$
- When a VNF $f \in F$ is migrated to server $v \in V$, certain migration cost d_f^v will be incurred.
  - the migration cost is proportional to the distance between servers

System model and problem formulation

B. Problem Formulation

- 1) Migration Cost:
  - As there are K to-be-migrated VNFs
    - at most K steps to update the forwarding rules
    - without violating the flow table size constraints
  - x_{fv}^{k} to indicate whether f is migrated to server v in step $k \in [1, K]$.
  - For each to-be-migrated VNF $f \in F$, it must be migrated to one server $v \in V$ with enough resource to accommodate it eventually.

$$\sum_{k=1}^{K} \sum_{v \in V} x_{fv}^{k} = 1, \forall f \in F,$$

$$\sum_{f \in F} x_{fv}^{k} \cdot S_f + \sum_{f' \in F_v} S_{f'} \leq C_v, \forall v \in V, k \in [1, K].$$

System model and problem formulation

- 2) Update Delay:
- To avoid congestion
  - part of flows will be migrated to new path during each update step
  - $0 \leq y_f^k \leq 1$: fraction of flows of VNF $f \in F$ updated from the old path P_f to the new one P'_f in step k
  - n_k, $k \in [1, K]$: whether the rule update is finished at step k.
- The value of n_k is determined by the value of y_f^k
  - $n_k = 1$ if and only if there exists a y_f^k, $\forall f \in F$ with value larger than 0
  - otherwise, $n_k = 0$

$$n_k \geq \sum_{f \in F} \frac{y_f^k}{K}, \forall k \in [1, K].$$

- As long as the update is finished in step k, there will be no more update operations in the subsequent steps.

$$n_k \geq n_{k+1}, \forall k \in [1, K-1].$$

System model and problem formulation

- Analyze the link capacity constraints and flow table size constraints of the problem.
- Let l_e^k and r_v^k represent the traffic load on link e and the number of rules on switch v in step k, respectively

$$l_e^0 = \sum_{f \in F} B_f \alpha_{fe}$$

α_{fe}= 1 if link $e \in P_f$; otherwise, α_{fe}= 0

- During rule update process, the old path and new path may coexist.
  - always guarantee that the total bandwidth requirement does not exceed the link capacity

$$l_e^k = l_e^{k-1} + \sum_{f \in F} B_f y_f^k \alpha'_{fe} - \sum_{f \in F} B_f y_f^k \alpha_{fe}$$

System model and problem formulation

- 3) Joint Optimization:
- As migrating f to server v incurs cost d_f^v, the overall migration cost therefore can be calculated as $\sum_{k=1}^{K} \sum_{v \in V} \sum_{f \in F} x_{fv}^k \cdot d_f^v$
- As n_k indicates whether there is any rule update in step k, we can calculate the total number of steps,
- Taking all the constraints discussed above, we formulate the joint optimization problem into MINLP form as:

$$
\begin{aligned}
\text{min}: \quad & \beta \sum_{k=1}^{K} \sum_{v \in V} \sum_{f \in F} x_{fv}^k \cdot d_f^v + \sum_{k=1}^{K} n_k \\
\text{s.t.} \quad & l_e^k \le c_e, \forall e \in E, k \in [1, K], \\
& r_v^k \le q_v, \forall v \in V, k \in [1, K], \\
& (1), (2), (3) - (6).
\end{aligned}
$$

$\beta \in (0, 1)$ as a bias factor to balance the two issues.

Heuristic algorithm design

Algorithm 1 Relaxation-based VNF migration and rule update

Phase 1: Solve the Relaxed Problem (12).
Relax the integer variable x^k_{fv} and n_k
Obtain a fraction solution $\widetilde{x}$ and $\widetilde{n}$
Phase 2: Round to 0-1 solution
$F^u = F$
$\forall x^k_{fv} \leftarrow 0$
while $F^u \neq \emptyset$ **do**
    Choose a maximum x^k_{fv}, $f \in F^u$, set $x^k_{fv} = 1$
    Check the constraint (2)
    if the currently selected x^k_{fv} is infeasible **then**
        Set $x^k_{fv} = 0$
        Choose next x^k_{fv} with smaller value
    else
        $F^u = F^u - f$
        Choose a maximum n_k, set $n_k = 1$
        Regard $n_k = 1$ as known, resolve the problem (12).
        Obtain fractional solutions $\widetilde{x}$ and $\widetilde{n}$
    end if
end while

the fractional solutions do not reflect the actual VNF migration and rule update decisions

the one with higher value shall be with higher priority to be converted into 1

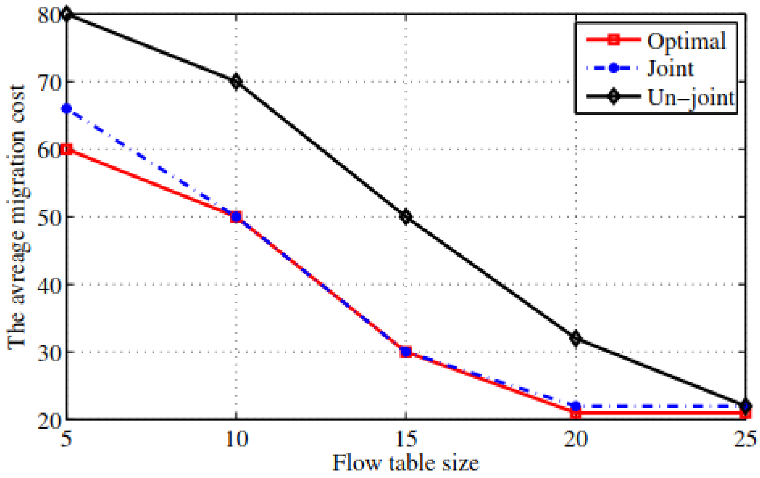
$$\min: \quad \beta \sum_{k=1}^{K} \sum_{v \in V} \sum_{f \in F} x^k_{fv} \cdot d^v_f + \sum_{k=1}^{K} n_k$$

$$\text{s.t.} \quad l^k_e \leq c_e, \forall e \in E, k \in [1, K],$$
$$r^k_v \leq q_v, \forall v \in V, k \in [1, K],$$
$$(1), (2), (3) - (6).$$

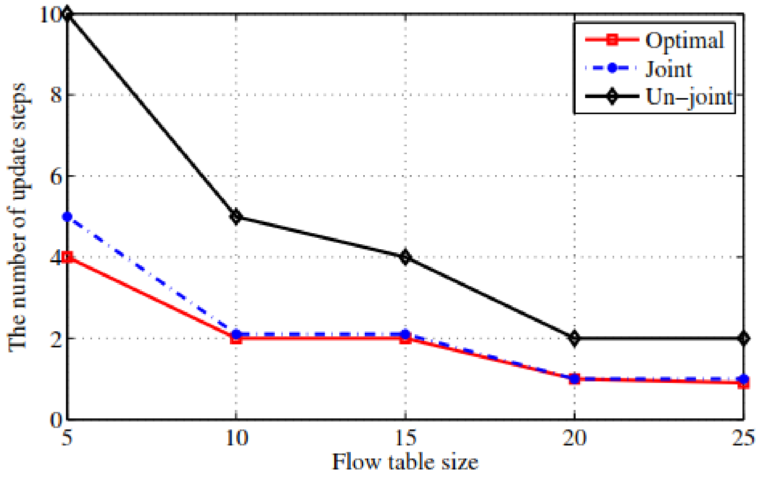
until every virtual function in F is migrated and all corresponding rules are updated

Performance evaluation

- On the effect of flow table size



(a) Migration cost

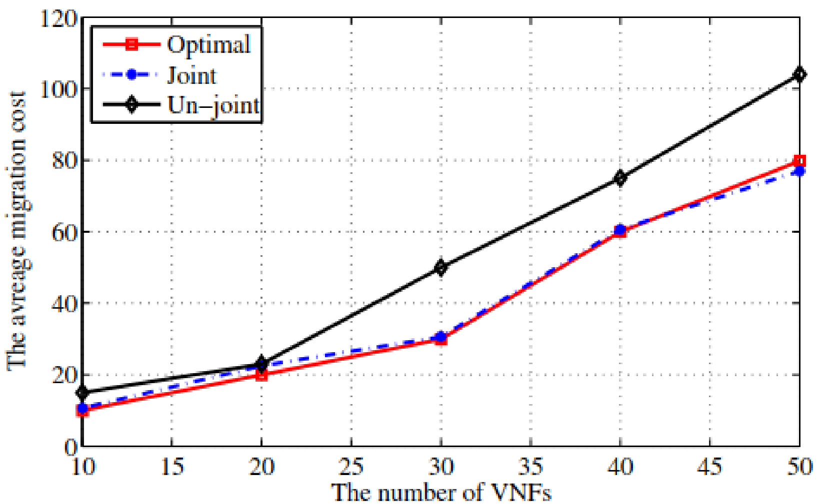


(b) Rule update delay

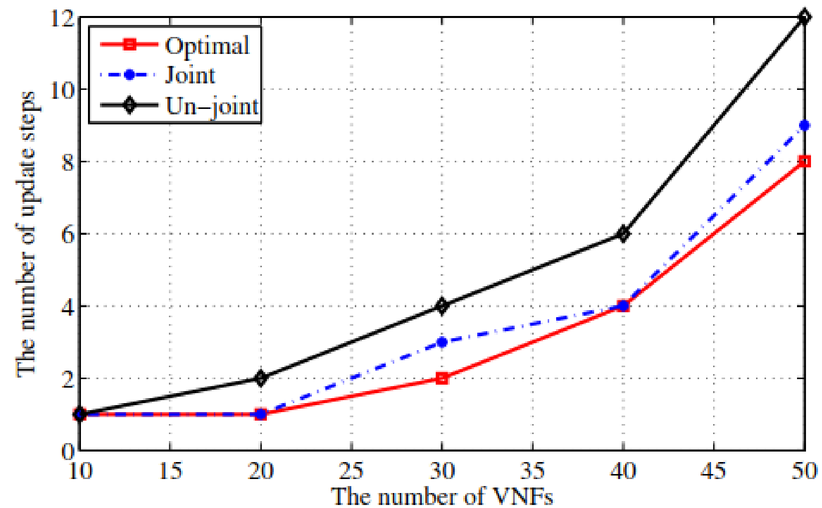
- There are 20 service chains to update.
- When the flow table size is small, the number of forwarding rules that can be stored in the flow table is limited.
- With the increase of flow table size, more rules can be stored in the flow table and hence more locations become available to migrate and it is easy to update quickly.

Performance evaluation

- On the effect of the number of VNFs



(a) Migration cost

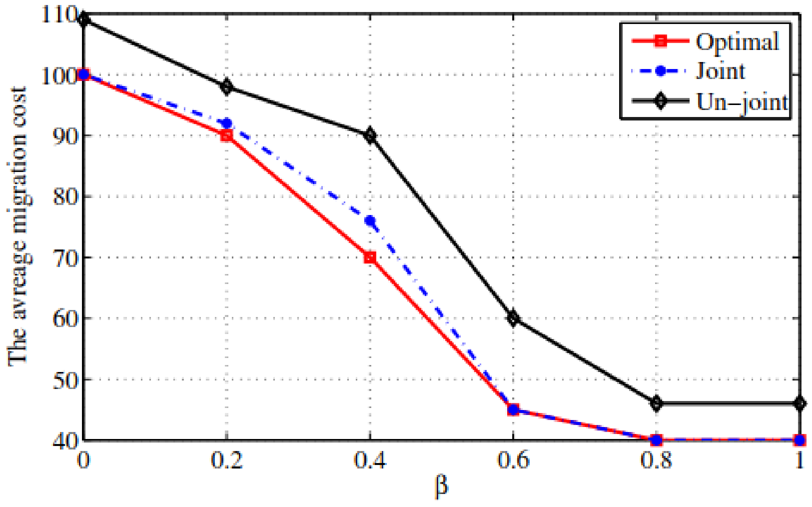


(b) Rule update delay

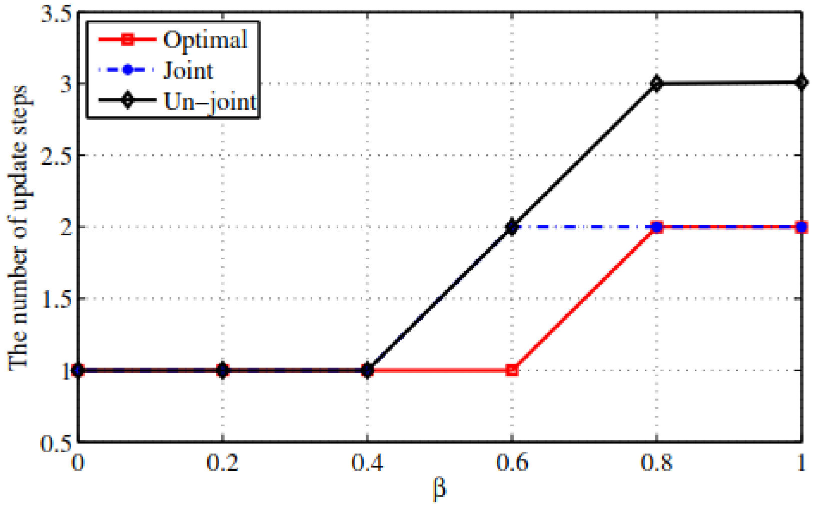
- The number of to-be-migrated VNFs varied from 10 to 50
- For each switch, the flow table size is fixed as 15.

Performance evaluation

- On the Effect of Bias Factor β



(a) Migration cost



(b) Rule update delay

- Vary the value of β from 0 to 1.
- Consider 20 VNFs to migrate and the flow table size is 15 on each switch.
- When β is small, we emphasize more on the update delay.
- When β is close to 1, we almost fully focus on the migration cost and do not care about the rule update delay.

Conclusion

- In this paper, we investigate the problem of VNF migration and forwarding rule update
  - aiming at minimizing the migration cost and rule update delay
- We discover that the VNF migration has a deep influence on the rule update decision and therefore they two shall be considered in a joint manner.
- We formulate the joint optimization problem into an MINLP form.
  - To address the computation complexity, we further propose a polynomial-time heuristic algorithm based on linearization and relaxation.
- By extensive simulations, we prove the correctness of our joint design philosophy and the high efficiency of our algorithm.

The end.