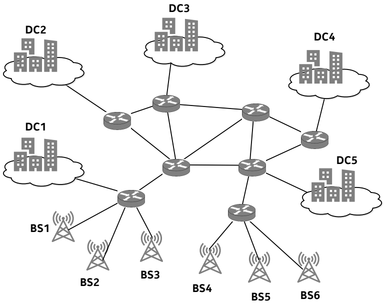
Dynamic Control and Optimization of Wireless Virtual Networks

Quang-Trung Luu

Université Paris-Saclay, CentraleSupélec, CNRS, L2S and Nokia Bell Labs

Advisors
Michel Kieffer, Alexandre Mouradian (Université Paris-Saclay)
Sylvaine Kerboeuf (Nokia Bell Labs)

CentraleSupélec, Gif-sur-Yvette, June 7, 2021



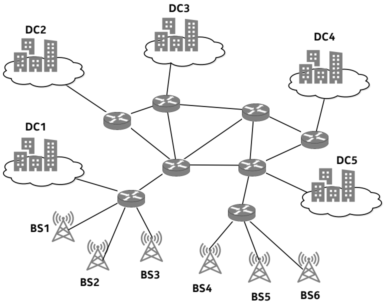
Past: physical dedicated devices: switches, routers, *etc*.

Network virtualization era:

- SDN (2008): decouples the control and data plane
- NFV (2012): virtualizes network functions → VNF

Network slicing (2015): leverages SDN and NFV [NGMN'15]

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function



Past: physical dedicated devices: switches, routers, *etc*.

Network virtualization era:

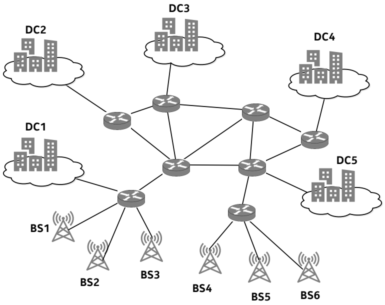
- SDN (2008): decouples the control and data plane
- NFV (2012): virtualizes network functions $\rightarrow$ VNF

Network slicing (2015): leverages SDN and NFV [NGMN'15]

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function

Network slicing | Introduction

NOKIA Bell Labs



Past: physical dedicated devices: switches, routers, *etc*.

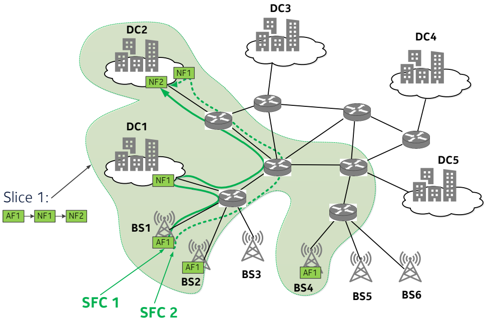
Network virtualization era:

- SDN (2008): decouples the control and data plane
- NFV (2012): virtualizes network functions $\rightarrow$ VNF

Network slicing (2015): leverages SDN and NFV [NGMN'15]

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function

NOKIA Bell Labs



Past: physical dedicated devices: switches, routers, *etc*.

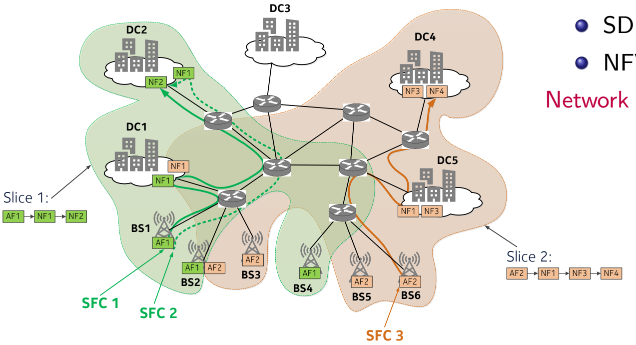
Network virtualization era:

- SDN (2008): decouples the control and data plane
- NFV (2012): virtualizes network functions $\rightarrow$ VNF

Network slicing (2015): leverages SDN and NFV [NGMN'15]

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function

NOKIA Bell Labs



Past: physical dedicated devices: switches, routers, *etc*.

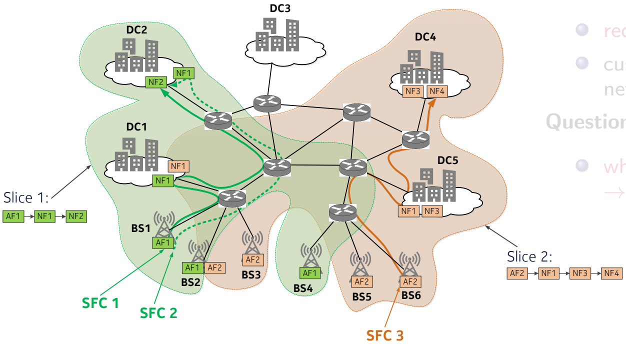
Network virtualization era:

- SDN (2008): decouples the control and data plane
- NFV (2012): virtualizes network functions → VNF

Network slicing (2015): leverages SDN and NFV [NGMN'15]

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function

NOKIA Bell Labs



Advantages:

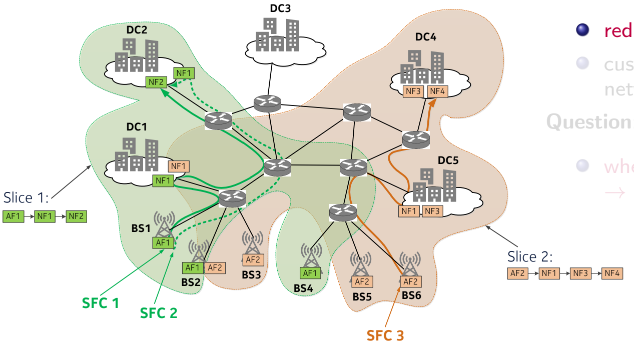
- increase flexibility in network operations [Rost'17]
- reduce overall equipment and management costs [Liang'14]
- customers can manage their own apps by exploiting built-in network slices tailored to their needs [GSM'17]

Question:

- where to place VNFs to deploy SFCs (slice)
  → embedding of slice's SFCs

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function

NOKIA Bell Labs



Advantages:

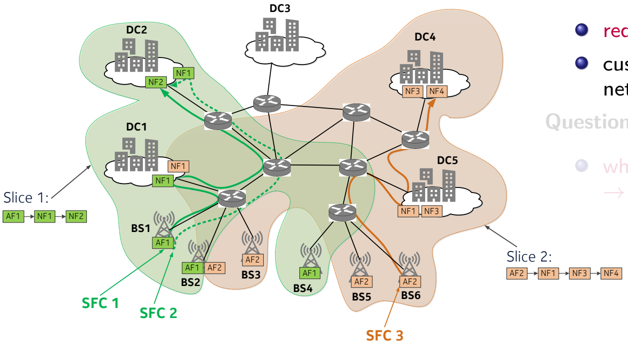
- increase flexibility in network operations [Rost'17]
- reduce overall equipment and management costs [Liang'14]
- customers can manage their own apps by exploiting built-in network slices tailored to their needs [GSM'17]

Question:

- where to place VNFs to deploy SFCs (slice) → embedding of slice's SFCs

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function

Network slicing | Introduction



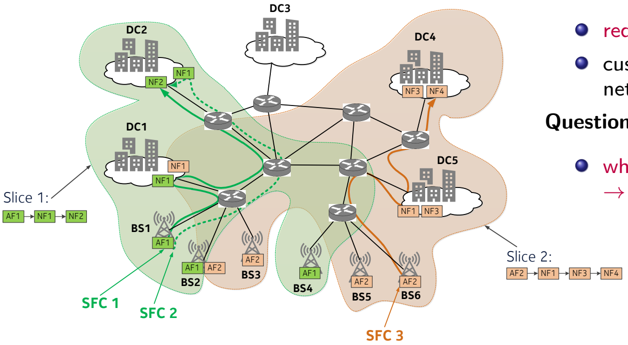
Advantages:

- increase flexibility in network operations [Rost'17]
- reduce overall equipment and management costs [Liang'14]
- customers can manage their own apps by exploiting built-in network slices tailored to their needs [GSM'17]

Question:

- where to place VNFs to deploy SFCs (slice)
  → embedding of slice's SFCs

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function



Advantages:

- increase flexibility in network operations [Rost'17]
- reduce overall equipment and management costs [Liang'14]
- customers can manage their own apps by exploiting built-in network slices tailored to their needs [GSM'17]

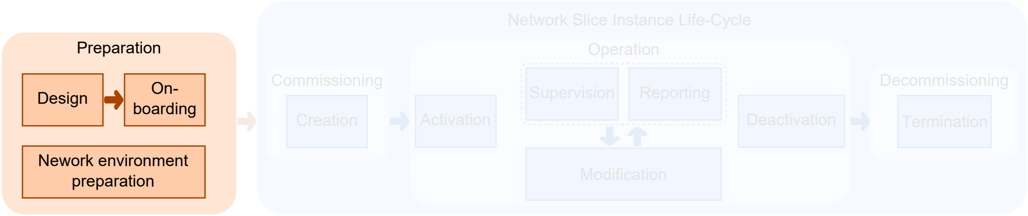
Question:

- where to place VNFs to deploy SFCs (slice)
  → embedding of slice's SFCs

SDN: Software Defined Networking NFV: Network Function Virtualization VNF: Virtual Network Function

NOKIA Bell Labs

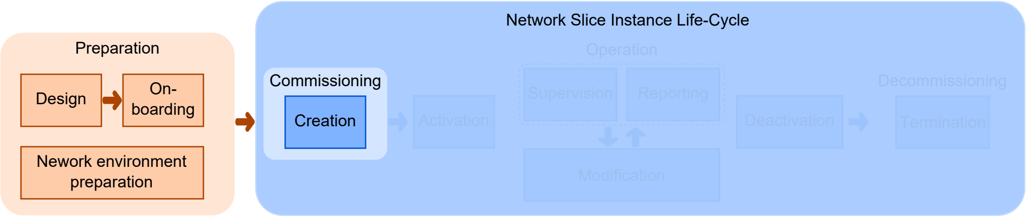
3GPP view on network slicing managements aspects



3GPP (2020). Management and Orchestration; Concepts, Use Cases and Requirements. 3GPP TS 28.530 V17.0.0.

Network slicing | Life-cycle

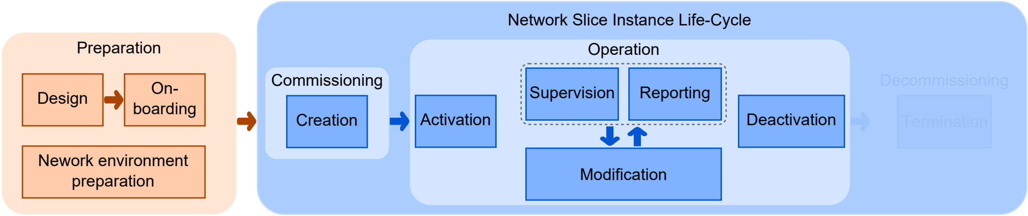
3GPP view on network slicing managements aspects



3GPP (2020). Management and Orchestration; Concepts, Use Cases and Requirements. 3GPP TS 28.530 V17.0.0.

NOKIA Bell Labs

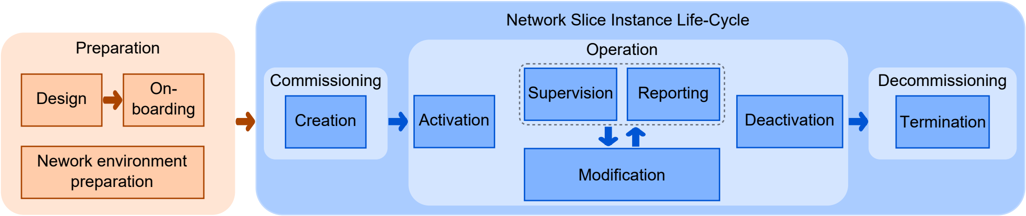
3GPP view on network slicing managements aspects



3GPP (2020). Management and Orchestration; Concepts, Use Cases and Requirements. 3GPP TS 28.530 V17.0.0.

NOKIA Bell Labs

3GPP view on network slicing managements aspects



3GPP (2020). Management and Orchestration; Concepts, Use Cases and Requirements. 3GPP TS 28.530 V17.0.0.

- SFC resource allocation (embedding): [Riggio'16], [Vizarreta'17], [Halabian'19]
- Slice resource allocation with coverage constraints: [Lee'16], [Chatterjee'18], [Teague'19]
- Uncertainty-aware slice resource allocation: [Huin'17], [Mireslami'19], [Fendt'19]
- Slice admission control: [Noroozi'19]
  - SAC with dynamic slice requests: [Bega'17], [Bega'20], [Ebrahimi'20], [Han'20]
- Slice resource provisioning
  - joint slice resource provisioning and allocation: [Xiong'19], [Sun'19] (radio resource only)

- SFC resource allocation (embedding): [Riggio'16], [Vizarreta'17], [Halabian'19]
- Slice resource allocation with coverage constraints: [Lee'16], [Chatterjee'18], [Teague'19]
- Uncertainty-aware slice resource allocation: [Huin'17], [Mireslami'19], [Fendt'19]
- Slice admission control: [Noroozi'19]
  - SAC with dynamic slice requests: [Bega'17], [Bega'20], [Ebrahimi'20], [Han'20]
- Slice resource provisioning
  - joint slice resource provisioning and allocation: [Xiong'19], [Sun'19] (radio resource only)

- SFC resource allocation (embedding): [Riggio'16], [Vizarreta'17], [Halabian'19]
- Slice resource allocation with coverage constraints: [Lee'16], [Chatterjee'18], [Teague'19]
- Uncertainty-aware slice resource allocation: [Huin'17], [Mireslami'19], [Fendt'19]
- Slice admission control: [Noroozi'19]
  - SAC with dynamic slice requests: [Bega'17], [Bega'20], [Ebrahimi'20], [Han'20]
- Slice resource provisioning
  - joint slice resource provisioning and allocation: [Xiong'19], [Sun'19] (radio resource only)

- SFC resource allocation (embedding): [Riggio'16], [Vizarreta'17], [Halabian'19]
- Slice resource allocation with coverage constraints: [Lee'16], [Chatterjee'18], [Teague'19]
- Uncertainty-aware slice resource allocation: [Huin'17], [Mireslami'19], [Fendt'19]
- Slice admission control: [Noroozi'19]
  - SAC with dynamic slice requests: [Bega'17], [Bega'20], [Ebrahimi'20], [Han'20]
- Slice resource provisioning
  - joint slice resource provisioning and allocation: [Xiong'19], [Sun'19] (radio resource only)

NOKIA Bell Labs

- SFC resource allocation (embedding): [Riggio'16], [Vizarreta'17], [Halabian'19]
- Slice resource allocation with coverage constraints: [Lee'16], [Chatterjee'18], [Teague'19]
- Uncertainty-aware slice resource allocation: [Huin'17], [Mireslami'19], [Fendt'19]
- Slice admission control: [Noroozi'19]
  - SAC with dynamic slice requests: [Bega'17], [Bega'20], [Ebrahimi'20], [Han'20]
- Slice resource provisioning
  - joint slice resource provisioning and allocation: [Xiong'19], [Sun'19] (radio resource only)

This thesis

NOKIA Bell Labs

Challenges addressed:

- Provision enough infrastructure resources to satisfy slice resource demands
- Account for coverage constraints
- Account for uncertainties
- Account for *in-advance* slice requests (anticipated provisioning)

This thesis

Challenges addressed:

- Provision enough infrastructure resources to satisfy slice resource demands
- Account for coverage constraints
- Account for uncertainties
- Account for *in-advance* slice requests (anticipated provisioning)

This thesis

Challenges addressed:

- Provision enough infrastructure resources to satisfy slice resource demands
- Account for coverage constraints
- Account for uncertainties
- Account for *in-advance* slice requests (anticipated provisioning)

This thesis

Challenges addressed:

- Provision enough infrastructure resources to satisfy slice resource demands
- Account for coverage constraints
- Account for uncertainties
- Account for *in-advance* slice requests (anticipated provisioning)

Contributions

Part I. Deterministic slice resource demands

Part II. Random slice resource demands

1

network provisioning

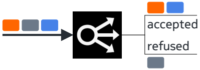
2

coverage-aware

3

uncertainty-aware

4



anticipated prov.

Contributions

Part I. Deterministic slice resource demands

Part II. Random slice resource demands

1

network provisioning

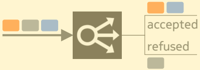
2

coverage-aware

3

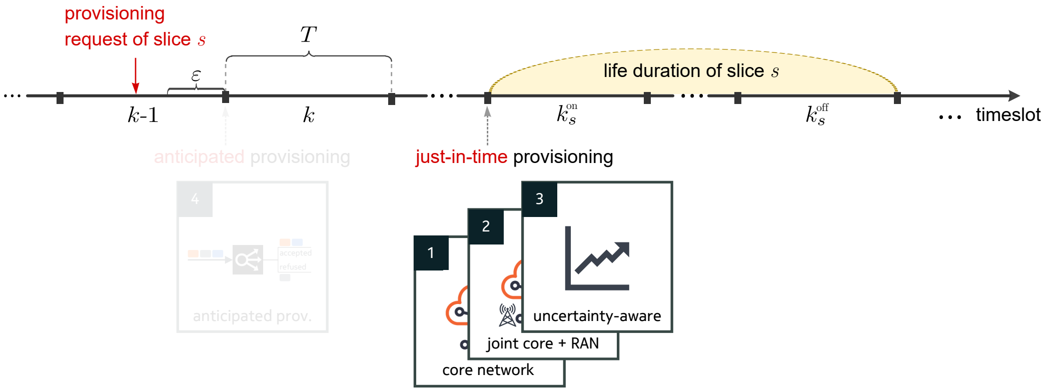
uncertainty-aware

4



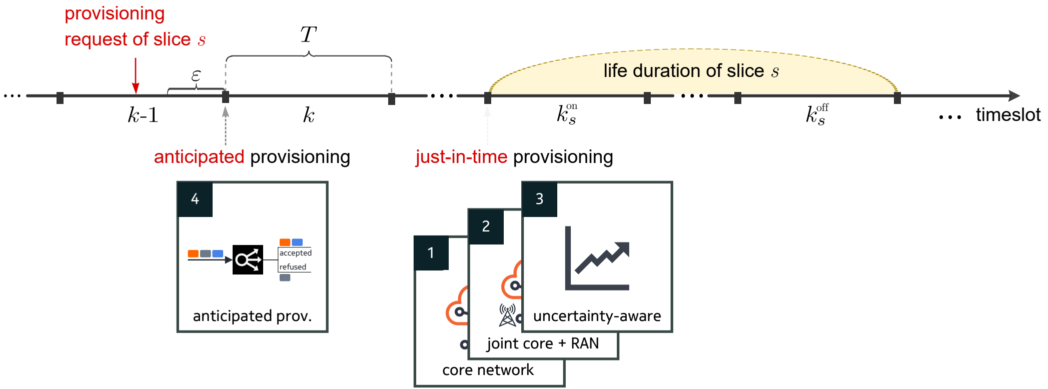
anticipated prov.

System Model | Slice provisioning requests



just-in-time: performed only for slices that need to be activated in the next time slot

NOKIA Bell Labs



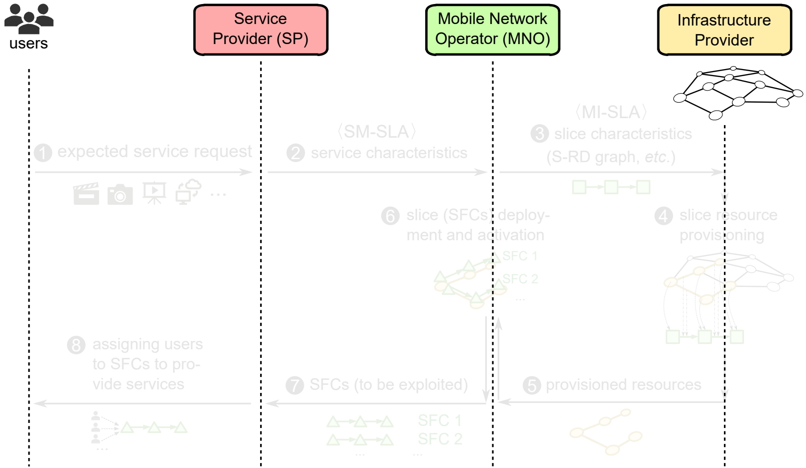
just-in-time: performed only for slices that need to be activated in the next time slot
anticipated: if not enough resources to satisfy all requests $\rightarrow$ useful to perform provisioning *in advance* to response to the request

Outline

NOKIA Bell Labs

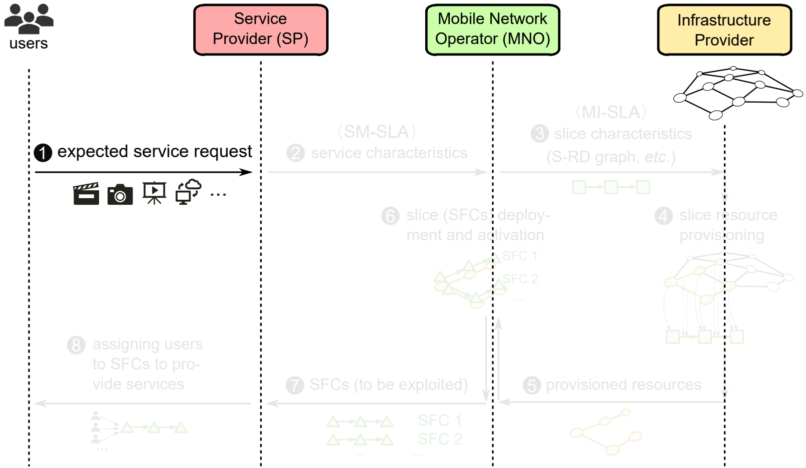
1. System model
2. Main contributions
   - Contribution 1: Network resource provisioning
   - Contribution 4: Anticipated provisioning
3. Conclusions and Perspective
   - Contribution 2: Coverage-aware resource provisioning
   - Contribution 3: Uncertainty-aware resource provisioning

System Model | Slice request processing

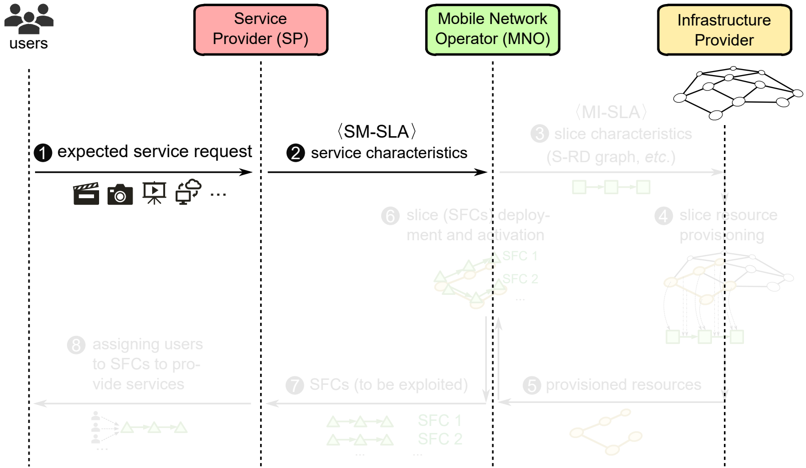


System Model | Slice request processing

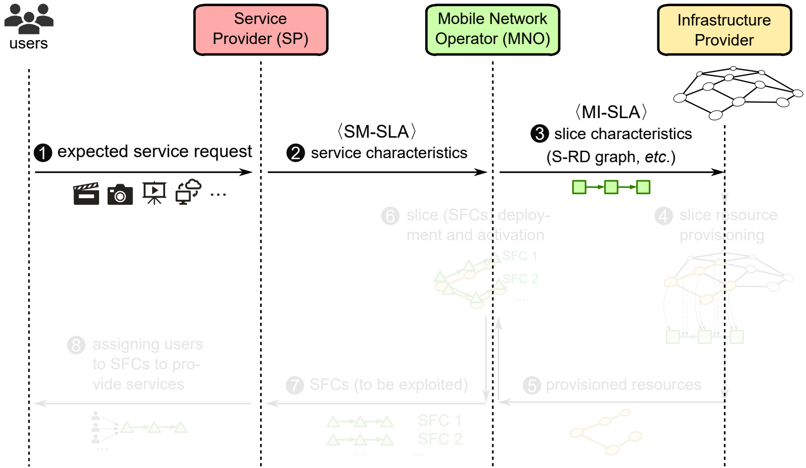
NOKIA Bell Labs



System Model | Slice request processing

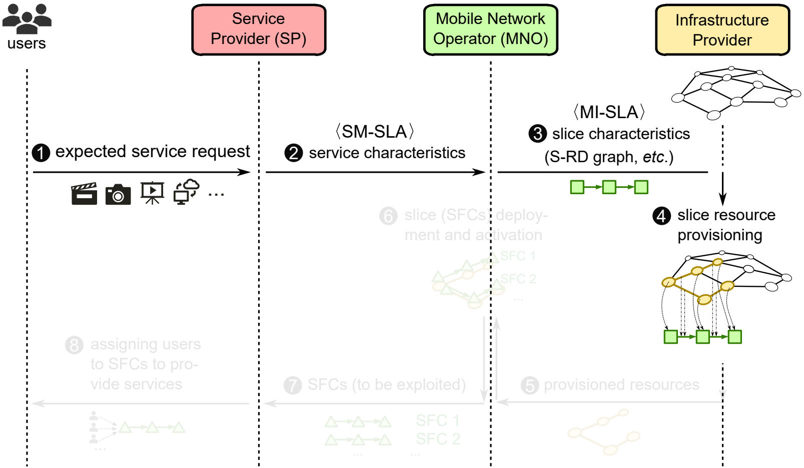


System Model | Slice request processing



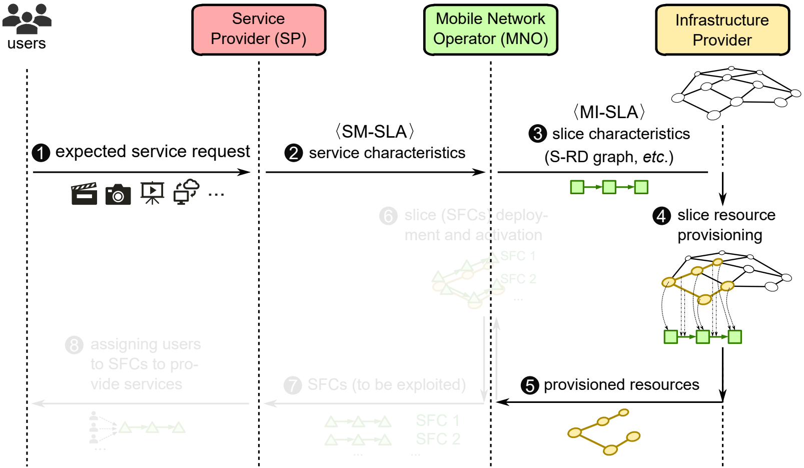
System Model | Slice request processing

NOKIA Bell Labs



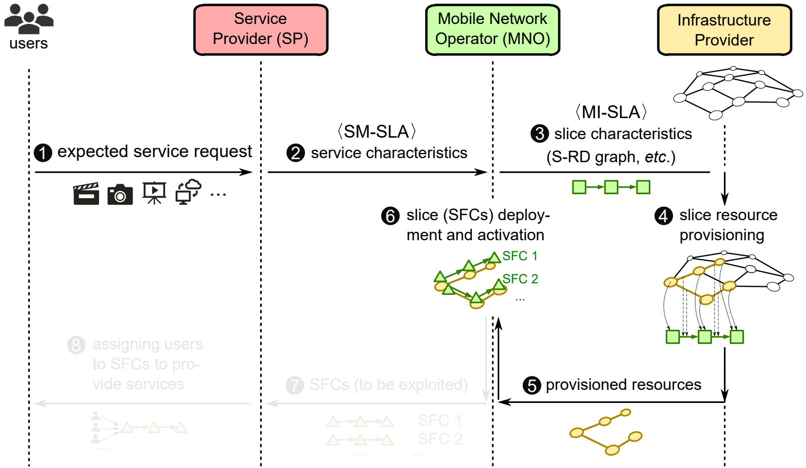
System Model | Slice request processing

NOKIA Bell Labs



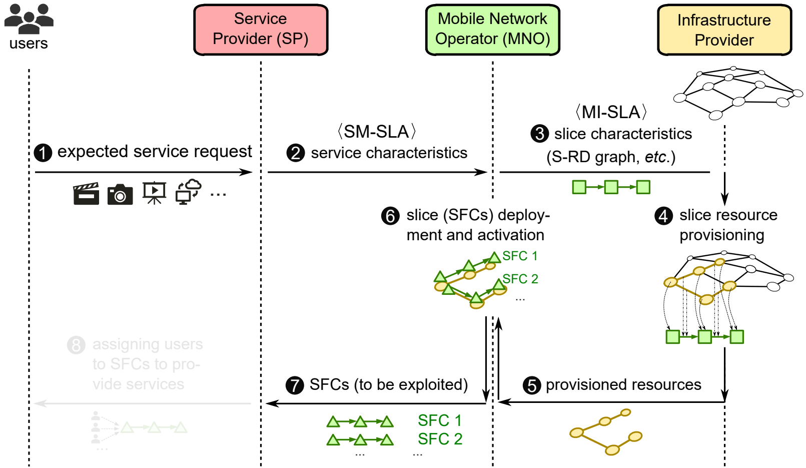
System Model | Slice request processing

NOKIA Bell Labs



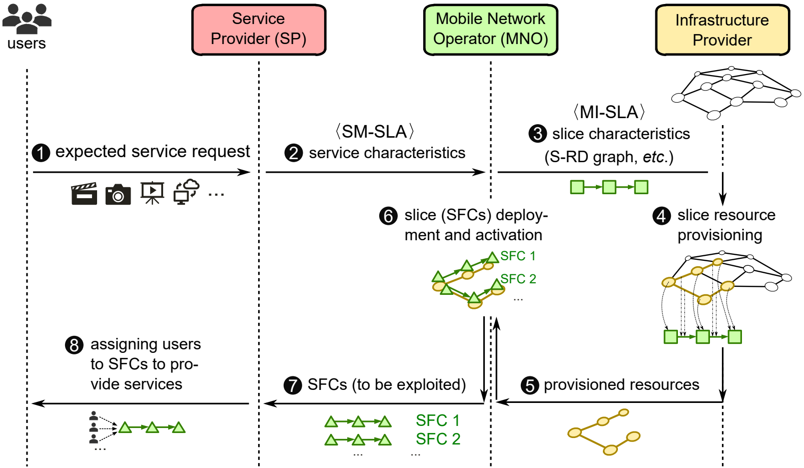
System Model | Slice request processing

NOKIA Bell Labs

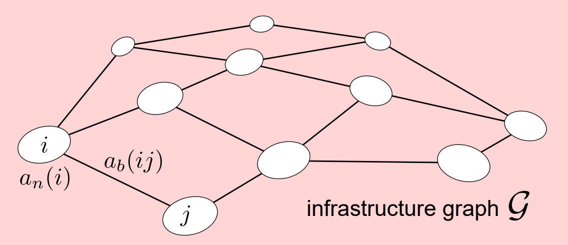


System Model | Slice request processing

NOKIA Bell Labs



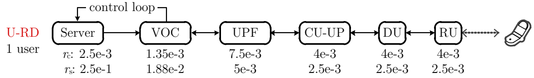
NOKIA Bell Labs



Infrastructure provides resources with associated costs

- node $i \in \mathcal{N}$: computing $a_c(i)$, memory $a_m(i)$, wireless $a_w(i)$
- link $ij \in \mathcal{E}$: bandwidth $a_b(ij)$
- costs: $c_c(i)$, $c_m(i)$, $c_w(i)$, $c_b(ij)$

Resource demand of a typical user (U-RD): $\mathbf{U}_s$



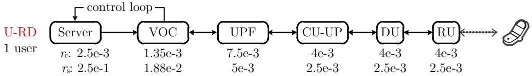
Resource demand of a slice (S-RD): $\mathbf{R}_s$

Resource demand of an SFC (SFC-RD): $\mathbf{r}_s$

System Model | Virtual graphs of resource demand

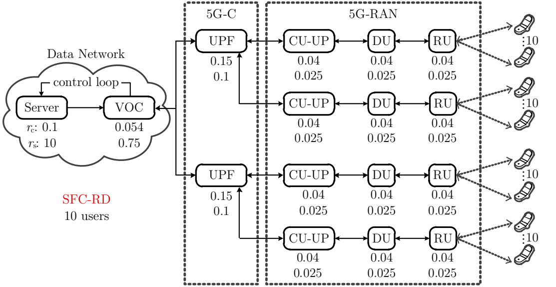
NOKIA Bell Labs

Resource demand of a typical user (U-RD): $\mathbf{U}_s$



Resource demand of a slice (S-RD): $\mathbf{R}_s$

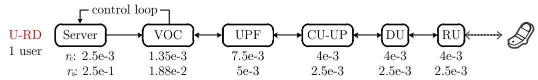
Resource demand of an SFC (SFC-RD): $\mathbf{r}_s$



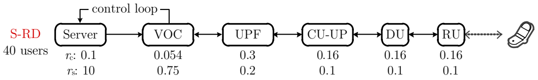
System Model | Virtual graphs of resource demand

NOKIA Bell Labs

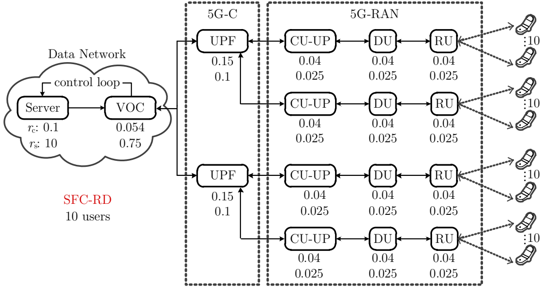
Resource demand of a typical user (U-RD): $\mathbf{U}_s$



Resource demand of a slice (S-RD): $\mathbf{R}_s$



Resource demand of an SFC (SFC-RD): $\mathbf{r}_s$



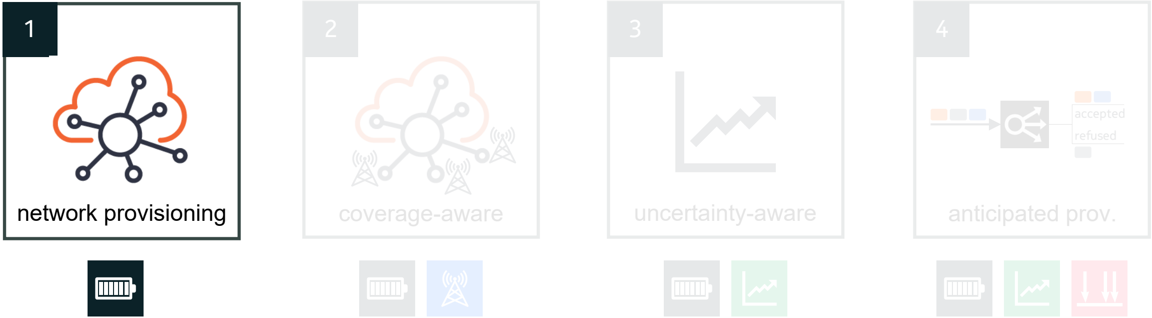
Outline

NOKIA Bell Labs

Contributions

Part I. Deterministic slice resource demands

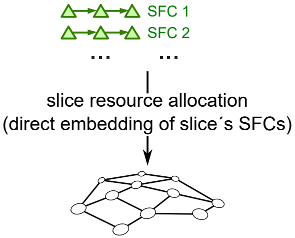
Part II. Random slice resource demands



NOKIA Bell Labs

Direct embedding vs provisioned embedding

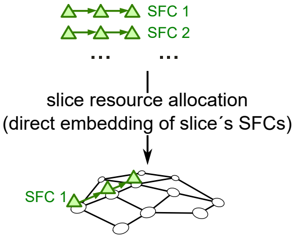
Prior works: Direct embedding (one-to-one mapping)



(1) Network provisioning | Problem statement

Direct embedding vs provisioned embedding

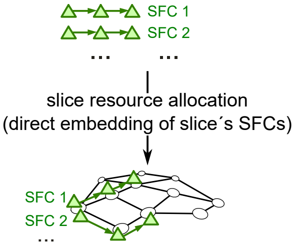
Prior works: Direct embedding (one-to-one mapping)



NOKIA Bell Labs

Direct embedding vs provisioned embedding

Prior works: Direct embedding (one-to-one mapping)



Offen formulated as (Mixed) Integer Linear Programming and solved either directly or using heuristics

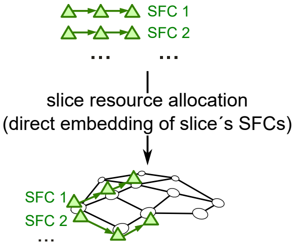
- ILP: [Cohen2015], [Riera2016], [Vizarreta2017]
- heuristics: column generation [Liu2017] , eigendecomposition [Mechtri2016]

(1) Network provisioning | Problem statement

NOKIA Bell Labs

Direct embedding vs provisioned embedding

Prior works: Direct embedding (one-to-one mapping)



Offen formulated as (Mixed) Integer Linear Programming and solved either directly or using heuristics

ILP: [Cohen2015], [Riera2016], [Vizarreta2017]

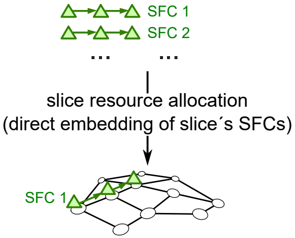
heuristics: column generation [Liu2017] , eigendecomposition [Mechtri2016]

Slice resource provisioning: (many-to-one mapping)

(1) Network provisioning | Problem statement

Direct embedding vs provisioned embedding

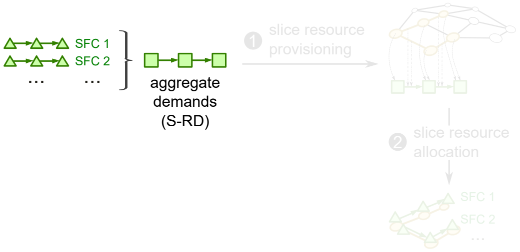
Prior works: Direct embedding (one-to-one mapping)



Offen formulated as (Mixed) Integer Linear Programming and solved either directly or using heuristics

- ILP: [Cohen2015], [Riera2016], [Vizarreta2017]
- heuristics: column generation [Liu2017] , eigendecomposition [Mechtri2016]

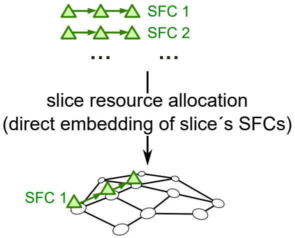
Slice resource provisioning: (many-to-one mapping)



NOKIA Bell Labs

Direct embedding vs provisioned embedding

Prior works: Direct embedding (one-to-one mapping)

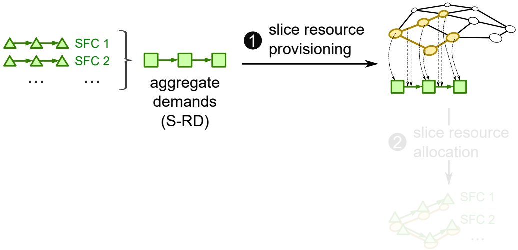


Offen formulated as (Mixed) Integer Linear Programming and solved either directly or using heuristics

ILP: [Cohen2015], [Riera2016], [Vizarreta2017]

heuristics: column generation [Liu2017] , eigendecomposition [Mechtri2016]

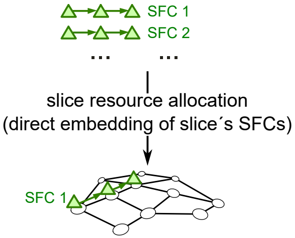
Slice resource provisioning: (many-to-one mapping)



(1) Network provisioning | Problem statement

Direct embedding vs provisioned embedding

Prior works: Direct embedding (one-to-one mapping)

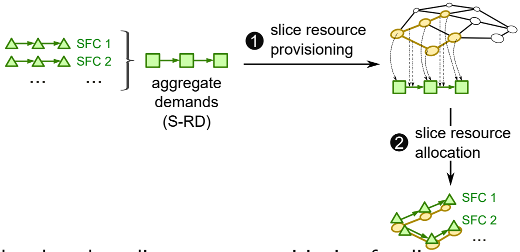


Offen formulated as (Mixed) Integer Linear Programming and solved either directly or using heuristics

ILP: [Cohen2015], [Riera2016], [Vizarreta2017]

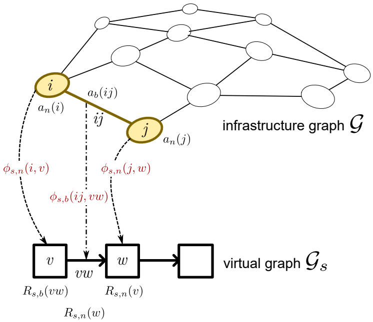
heuristics: column generation [Liu2017] , eigendecomposition [Mechtri2016]

Slice resource provisioning: (many-to-one mapping)



Related work: radio resource provisioning for slices: [Sun2019], [Xiong2019]

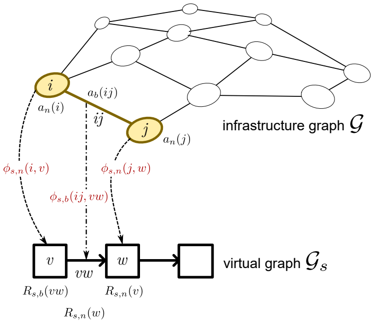
NOKIA Bell Labs



- $\phi_{s,n}(i,v) \in [0,1]$: Proportion of provisioned node resources
- $\phi_{s,b}(ij,vw) \in [0,1]$: Proportion of provisioned link resources
- $\widetilde{\phi}_{s,n}(i,v) \in \{0,1\}$: Node mapping indicator

Objective function

$$
\begin{aligned}
c_{\text{total}}(\boldsymbol{\Phi}) &= \sum_{s} c_{\text{total}}(\boldsymbol{\Phi}_s) \\
&= \sum_{s,i,v,n} a_n(i)\,\phi_{s,n}(i,v) c_n(i) && \text{(node resource leasing)} \\
&+ \sum_{s,ij,vw} a_b(ij)\,\phi_{s,b}(ij,vw) c_b(ij) && \text{(link resource leasing)} \\
&+ \sum_{s,i,v} \widetilde{\phi}_{s,n}(i,v)\, c_d(i) && \text{(VNF deployment)}
\end{aligned}
$$



- $\phi_{s,n}(i,v) \in [0,1]$: Proportion of provisioned node resources
- $\phi_{s,\mathrm{b}}(ij,vw) \in [0,1]$: Proportion of provisioned link resources
- $\widetilde{\phi}_{s,n}(i,v) \in \{0,1\}$: Node mapping indicator

Objective function

$$
\begin{aligned}
c_{\text{total}}(\boldsymbol{\Phi}) &= \sum_{s} c_{\text{total}}(\boldsymbol{\Phi}_s) \\
&= \sum_{s,i,v,n} a_n(i)\,\phi_{s,n}(i,v) c_n(i) && \text{(node resource leasing)} \\
&+ \sum_{s,ij,vw} a_\mathrm{b}(ij)\,\phi_{s,\mathrm{b}}(ij,vw) c_\mathrm{b}(ij) && \text{(link resource leasing)} \\
&+ \sum_{s,i,v} \widetilde{\phi}_{s,n}(i,v) c_\mathrm{d}(i) && \text{(VNF deployment)}
\end{aligned}
$$

NOKIA Bell Labs

Constraint 1: Slice resource requirements

$$\sum_{i} \underbrace{a_n(i)\,\phi_{s,n}(i,v)}_{\text{prov. rsrc from } i \text{ to } v} \geq \underbrace{R_n(v)}_{\text{S-RD at } v}, \forall s \in \mathcal{S}, v \in \mathcal{N}_s, n \in \Upsilon$$

$$\sum_{ij} \underbrace{a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,vw)}_{\text{prov. rsrc from } ij \text{ to } vw} \geq \underbrace{R_{\mathrm{b}}(vw)}_{\text{S-RD at } v}, \forall s \in \mathcal{S}, vw \in \mathcal{E}_s$$

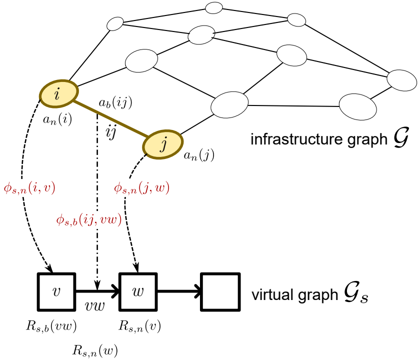
Constraint 2: Infrastructure resource limit

$$\sum_{s,v} \phi_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_{s,vw} \phi_{s,\mathrm{b}}(ij,vw) \leq 1, \forall ij \in \mathcal{E}$$

Other constraints:

- proportionality of provisioned node resources (computing-memory-wireless)
- flow conservation: ensures flow continuity



Constraint 1: Slice resource requirements

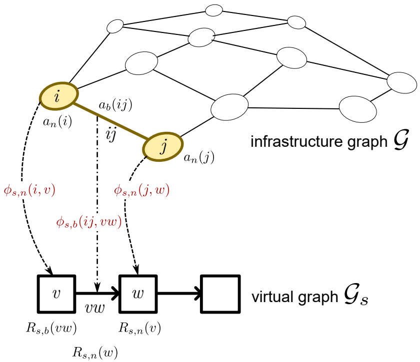
$$\sum_i \underbrace{a_n(i)\,\phi_{s,n}(i,v)}_{\text{prov. rsrc from } i \text{ to } v} \geq \underbrace{R_n(v)}_{\text{S-RD at } v}, \forall s \in \mathcal{S}, v \in \mathcal{N}_s, n \in \Upsilon$$

$$\sum_{ij} \underbrace{a_b(ij)\,\phi_{s,b}(ij,vw)}_{\text{prov. rsrc from } ij \text{ to } vw} \geq \underbrace{R_b(vw)}_{\text{S-RD at } v}, \forall s \in \mathcal{S}, vw \in \mathcal{E}_s$$

Constraint 2: Infrastructure resource limit

$$\sum_{s,v} \phi_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_{s,vw} \phi_{s,b}(ij,vw) \leq 1, \forall ij \in \mathcal{E}$$



Other constraints:

- proportionality of provisioned node resources (computing-memory-wireless)
- flow conservation: ensures flow continuity

Constraint 1: Slice resource requirements

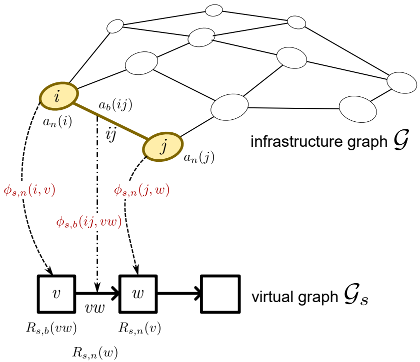
$$\sum_{i} \underbrace{a_n(i)\,\phi_{s,n}(i,v)}_{\text{prov. rsrc from } i \text{ to } v} \geq \underbrace{R_n(v)}_{\text{S-RD at } v}, \forall s \in \mathcal{S}, v \in \mathcal{N}_s, n \in \Upsilon$$

$$\sum_{ij} \underbrace{a_{\mathrm{b}}(ij)\,\phi_{s,\mathrm{b}}(ij,vw)}_{\text{prov. rsrc from } ij \text{ to } vw} \geq \underbrace{R_{\mathrm{b}}(vw)}_{\text{S-RD at } v}, \forall s \in \mathcal{S}, vw \in \mathcal{E}_s$$

Constraint 2: Infrastructure resource limit

$$\sum_{s,v} \phi_{s,n}(i,v) \leq 1, \forall i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_{s,vw} \phi_{s,\mathrm{b}}(ij,vw) \leq 1, \forall ij \in \mathcal{E}$$



Other constraints:

- proportionality of provisioned node resources (computing-memory-wireless)
- flow conservation: ensures flow continuity

NOKIA Bell Labs

MILP network resource provisioning

minimize $c_{\text{total}}(\mathbf{\Phi}) = \sum_{s \in \mathcal{S}} c_{\text{total}}(\mathbf{\Phi}_s)$,

subject to

(1) slice resource requirements

(2) infrastructure resource limits

(3) resource proportionality

(4) flow conservation

Provisioning approaches:

(1) Sequential approach (SP): Resources are provisioned slice-by-slice

(2) Joint approach (JP): Provisioning performed for all slices simultaneously

(1) Network provisioning | Main constraints

MILP network resource provisioning

minimize $c_{\text{total}}(\mathbf{\Phi}) = \sum_{s \in \mathcal{S}} c_{\text{total}}(\mathbf{\Phi}_s)$,

subject to

(1) slice resource requirements

(2) infrastructure resource limits

(3) resource proportionality

(4) flow conservation

Provisioning approaches:

(1) Sequential approach (SP): Resources are provisioned slice-by-slice

(2) Joint approach (JP): Provisioning performed for all slices simultaneously

NOKIA Bell Labs

MILP network resource provisioning

minimize $c_{\text{total}}(\mathbf{\Phi}) = \sum_{s \in \mathcal{S}} c_{\text{total}}(\mathbf{\Phi}_s)$,

subject to

(1) slice resource requirements

(2) infrastructure resource limits

(3) resource proportionality

(4) flow conservation

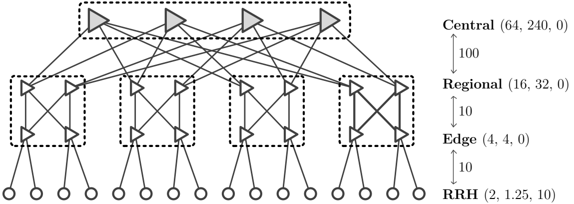
Provisioning approaches:

(1) Sequential approach (SP): Resources are provisioned slice-by-slice

(2) Joint approach (JP): Provisioning performed for all slices simultaneously

k-ary fat-tree topology

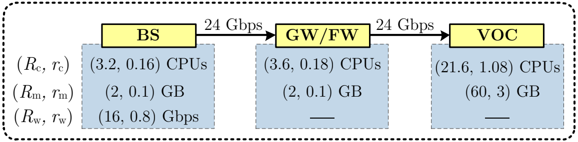
- Per-unit resource cost $= 1$ (whatever the type of resource)
- Deployment cost $c_{\mathrm{d}} = 50$



NOKIA Bell Labs

Slices: HD video streaming with *deterministic* S-RD

- Data rate at 4 Mbps for $N_s = 4000$ users.



Simulation schemes

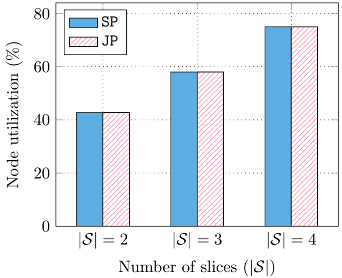
Comparison of 2 resource provisioning approaches:

- Sequential approach (SP)
- Joint approach (JP)

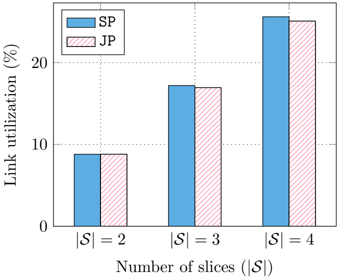
Both schemes are evaluated using CPLEX MILP solver interfaced with MATLAB.

NOKIA Bell Labs

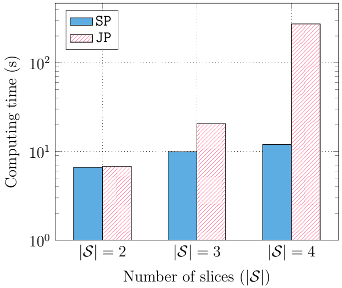
Sequential (SP) vs Joint (JP) approaches, with different $|\mathcal{S}|$



node utilization



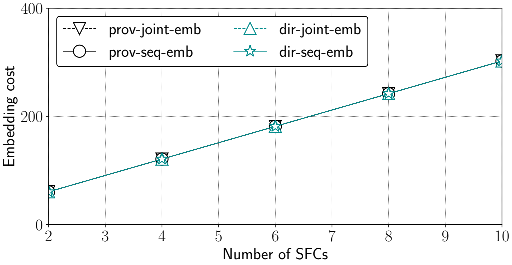
link utilization



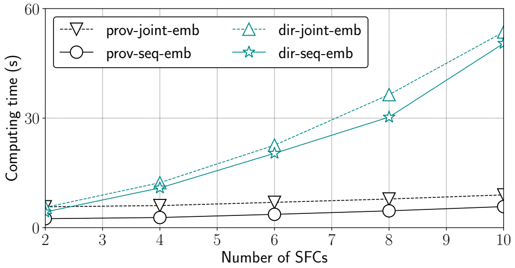
computing time

NOKIA Bell Labs

Provisioned embedding vs Direct embedding (single slice)



Embedding cost



Computing time

Provisioned embedding: embedding is performed with only a subset of the infrastructure graph

- S-RD graph: aggregation of resource demands
- Slice resource provisioning: mapping of infrastructure graph to S-RD graph
- Two slice resource provisioning approaches:
  - Sequential approach (`SP`): more efficient in terms of computing time
  - Nevertheless: degraded node and link utilization compared to Joint approach (`JP`)
- Once provisioning completed, SFC embedding is faster

Contributions

Part I. Deterministic slice resource demands

1

network provisioning

2

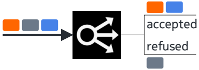
coverage-aware

Part II. Random slice resource demands

3

uncertainty-aware

4



anticipated prov.

NOKIA Bell Labs

Related works

- dynamic slice/SFC deployment
  - [Liu'17]: deployment of newly arrived SFCs and readjustment of in-service SFCs
  - [Wang'19]: slice reconfiguration at two levels: SFCs within a slice at small time intervals, or multiple slices at larger time interval
- slice admission control:
  - [Bega'17], [Bega'20]: online SAC for radio resources at base stations
  - [Han'20]: SAC with dynamic requests, but not accounting for uncertainties

(4) Anticipated provisioning | Problem statement

Related works

- dynamic slice/SFC deployment
  - [Liu'17]: deployment of newly arrived SFCs and readjustment of in-service SFCs
  - [Wang'19]: slice reconfiguration at two levels: SFCs within a slice at small time intervals, or multiple slices at larger time interval
- slice admission control:
  - [Bega'17], [Bega'20]: online SAC for radio resources at base stations
  - [Han'20]: SAC with dynamic requests, but not accounting for uncertainties

(4) Anticipated provisioning | Problem statement

NOKIA Bell Labs

This contribution:

Multiple slice requests from tenants to MNO

Uncertainties

- slice resource demands
- available infrastructure resource

Dynamic nature of slice provisioning requests

- arrive at anytime
- various activation times
- various lifetime
- resource needs vary during slice operation

This contribution:

Multiple slice requests from tenants to MNO

Uncertainties

- slice resource demands
- available infrastructure resource

Dynamic nature of slice provisioning requests

- arrive at anytime
- various activation times
- various lifetime
- resource needs vary during slice operation

(4) Anticipated provisioning | Problem statement

NOKIA Bell Labs

This contribution:
Multiple slice requests from tenants to MNO
Uncertainties

- slice resource demands
- available infrastructure resource

Dynamic nature of slice provisioning requests

- arrive at anytime
- various activation times
- various lifetime
- resource needs vary during slice operation

NOKIA Bell Labs

This contribution:

Multiple slice requests from tenants to MNO

Uncertainties

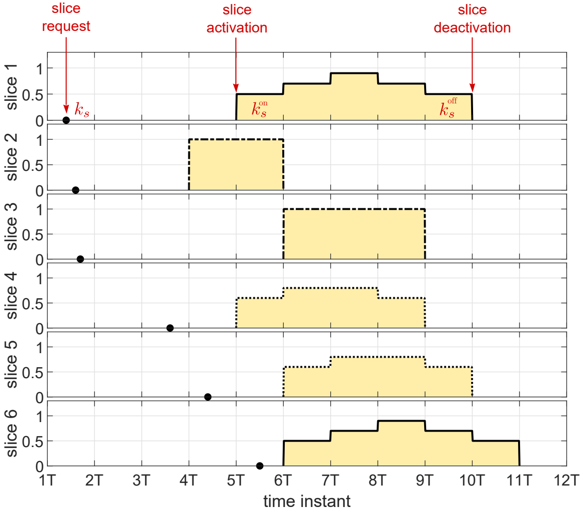
- slice resource demands
- available infrastructure resource

Dynamic nature of slice provisioning requests

- arrive at anytime
- various activation times
- various lifetime
- resource needs vary during slice operation

(4) Anticipated provisioning | System model

Slice provisioning requests



NOKIA Bell Labs

Slice classes: Premium and Standard
Initial priority assignment

$$P_{s,k} = \begin{cases} P_{\max}, & \text{for Premium slices,} \\ 0 & \text{for Standard slice, if } k_s^{\text{on}} > k+1, \\ P_{\max} - 1 & \text{for Standard slice, if } k_s^{\text{on}} = k+1. \end{cases}$$

Only slices with $P_{s,k} \geqslant P_{\text{thres}}$ are processed

$$P_{\text{thres}} = \alpha \left(P_{\max} - 1 \right), \text{ with } \alpha \in [0, 1] \, .$$

Slice classes: Premium and Standard
Initial priority assignment

$$P_{s,k} = \begin{cases} P_{\max}, & \text{for Premium slices,} \\ 0 & \text{for Standard slice, if } k_s^{\text{on}} > k+1, \\ P_{\max} - 1 & \text{for Standard slice, if } k_s^{\text{on}} = k+1. \end{cases}$$

Only slices with $P_{s,k} \geqslant P_{\text{thres}}$ are processed

$$P_{\text{thres}} = \alpha \left(P_{\max} - 1\right), \text{ with } \alpha \in [0,1].$$

NOKIA Bell Labs

Slice classes: Premium and Standard
Initial priority assignment

$$P_{s,k} = \begin{cases} P_{\max}, & \text{for Premium slices,} \\ 0 & \text{for Standard slice, if } k_s^{\text{on}} > k+1, \\ P_{\max} - 1 & \text{for Standard slice, if } k_s^{\text{on}} = k+1. \end{cases}$$

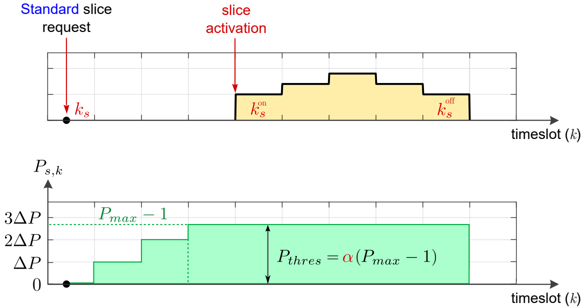
Only slices with $P_{s,k} \geqslant P_{\text{thres}}$ are processed

$$P_{\text{thres}} = \alpha \left(P_{\max} - 1 \right), \text{ with } \alpha \in [0,1],$$

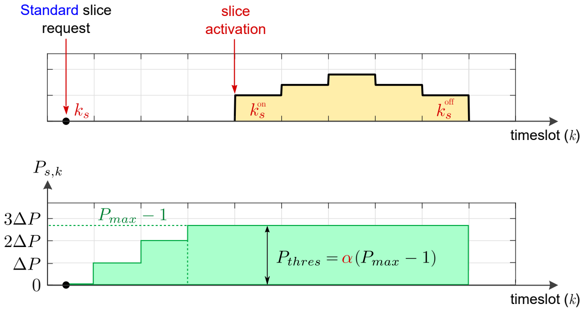
Priority update for unprocessed Standard slices in next time slots:

$$P_{s,k} = \begin{cases} \min\left\{ P_{s,k-1} + \Delta P, P_{\max} - 1 \right\} & \text{if } k_s^{\text{on}} > k+1, \\ P_{\max} - 1 & \text{if } k_s^{\text{on}} = k+1, \end{cases}$$

$\Delta P \geqslant 0$: priority increment.



(4) Anticipated provisioning | Priority assignment



Parameterized priority level with α and ΔP

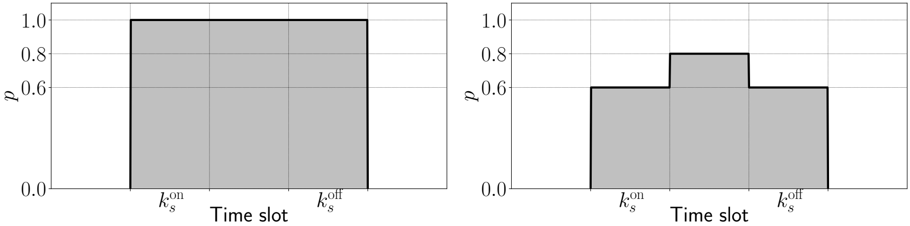
- $\alpha = 0$: Premium = Standard, $\forall \Delta P \rightarrow$ min benefit for Premium slices
- $\alpha = 1$, $\Delta P = 0$, Standard slices are delayed until the time slot preceding their activation $\rightarrow$ max benefit for Premium slices

(4) Anticipated provisioning | S-RD

NOKIA Bell Labs

Slice resource demands:

- Number of users of slice s follows binomial distribution $\mathcal{B}\left(p_{s,k}\right)$
- Temporal evolution of $p_{s,k}$



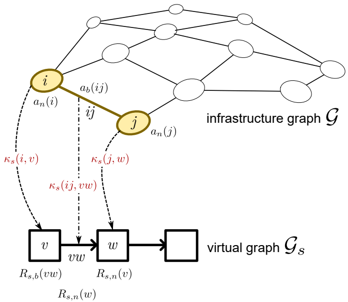
constant over a time interval | piece-wise constant

Service usage pattern

(4) Anticipated provisioning | Variables

Resource provisioning for slice s defined by mapping between $\mathcal{G}$ and $\mathcal{G}_s$, for time slot ℓ

- $d_s \in \{0, 1\}$ indicates whether slice s is accepted or not
- $\kappa_{s,\ell}(i, v) \in \mathbb{N}$ determines number of VNFs that node i can provide resources for virtual node v
- $\kappa_{s,\ell}(ij, vw) \in \mathbb{N}$ determines bandwidth provisioned by link ij to support the traffic between virtual nodes of type v and w



Note:
Contributions 1 and 2 use $\mathbf{\Phi}_s$ as main decision variables
Here: $\boldsymbol{\kappa}_s$

NOKIA Bell Labs

Assignment $\boldsymbol{\kappa}_{s,\ell}$ for slice s over time slot ℓ must satisfy

- infrastructure node and link capacity limits, w.r.t
  - background traffic
- service satisfaction probability, w.r.t
  - usage uncertainties
  - resource availability
- flow conservation constraint

(4) Anticipated provisioning | SSP constraint

NOKIA Bell Labs

Service Satisfaction Probability (SSP) of slice s at time slot k for a given assignment $\boldsymbol{\kappa}_{s,\ell}$

$$p_{s,\ell}\left(\boldsymbol{\kappa}_{s,\ell}, d_s\right) = \Pr\left\{\begin{array}{l} \underbrace{\sum_i \kappa_{s,\ell}(i,v)\, r_{s,n}(v)}_{\text{prov. rsrc for } v} \geqslant d_s \underbrace{R_{s,n,\ell}(v)}_{\text{S-RD of } v}, \forall v, n, \\ \underbrace{\sum_{ij} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw)}_{\text{prov. rsrc for } vw} \geqslant d_s \underbrace{R_{s,\mathrm{b},\ell}(vw)}_{\text{S-RD of } vw}, \forall vw \end{array}\right\} \geqslant \underbrace{\underline{p}_s^{\mathrm{sp}}}_{\text{required min SSP}}, \forall s \in \mathcal{R}_k, \ell \in \mathcal{K}_s$$

Relaxed constraints:

$$\sum_i \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \geqslant d_s \underbrace{\left[\overline{R}_{s,n,\ell}(v) + \gamma_{s,\ell} \widetilde{R}_{s,n,\ell}(v)\right]}_{\text{S-RD of } v}, \forall \ell > k, v \in \mathcal{N}_s, \forall n \in \Upsilon,$$

$$\sum_{ij} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant d_s \underbrace{\left[\overline{R}_{s,\mathrm{b},\ell}(vw) + \gamma_{s,\ell} \widetilde{R}_{s,\mathrm{b},\ell}(vw)\right]}_{\text{S-RD of } vw}, \forall \ell > k, vw \in \mathcal{E}_s.$$

$\mathcal{R}_k$: set of slices processed at time slot k
$\overline{R}$, $\widetilde{R}$: mean and standard deviation of S-RD
$\gamma_{s,\ell}$: safety factor

(4) Anticipated provisioning | SSP constraint

NOKIA Bell Labs

Service Satisfaction Probability (SSP) of slice s at time slot k for a given assignment $\boldsymbol{\kappa}_{s,\ell}$

$$p_{s,\ell}\left(\boldsymbol{\kappa}_{s,\ell}, d_s\right) = \Pr\left\{\begin{array}{l} \underbrace{\sum_i \kappa_{s,\ell}(i,v)\, r_{s,n}(v)}_{\text{prov. rsrc for } v} \geqslant d_s \underbrace{R_{s,n,\ell}(v)}_{\text{S-RD of } v}, \forall v, n, \\ \underbrace{\sum_{ij} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw)}_{\text{prov. rsrc for } vw} \geqslant d_s \underbrace{R_{s,\mathrm{b},\ell}(vw)}_{\text{S-RD of } vw}, \forall vw \end{array}\right\} \geqslant \underbrace{p_s^{\mathrm{sp}}}_{\text{required min SSP}}, \forall s \in \mathcal{R}_k, \ell \in \mathcal{K}_s$$

Relaxed constraints:

$$\sum_i \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \geqslant d_s \underbrace{\left[\overline{R}_{s,n,\ell}(v) + \gamma_{s,\ell}\widetilde{R}_{s,n,\ell}(v)\right]}_{\text{S-RD of } v}, \forall \ell > k, v \in \mathcal{N}_s, \forall n \in \Upsilon,$$

$$\sum_{ij} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant d_s \underbrace{\left[\overline{R}_{s,\mathrm{b},\ell}(vw) + \gamma_{s,\ell}\widetilde{R}_{s,\mathrm{b},\ell}(vw)\right]}_{\text{S-RD of } vw}, \forall \ell > k, vw \in \mathcal{E}_s.$$

$\mathcal{R}_k$: set of slices processed at time slot k
$\overline{R}$, $\widetilde{R}$: mean and standard deviation of S-RD
$\gamma_{s,\ell}$: safety factor

NOKIA Bell Labs

Impact Probability (ImP) constraints on background services at infrastructure node i and link ij:

$$p^{\mathrm{im}}_{n,\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, i\right) = \Pr\left\{ \underbrace{\sum_s \sum_v \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right)}_{\text{resources prov. by } i} \geqslant a_n\left(i\right) - \underbrace{B_{n,\ell}\left(i\right)}_{\text{bkgd serv. at } i} \right\} \leqslant \underbrace{\overline{p}^{\mathrm{im}}}_{\text{required max ImP}},$$

$$p^{\mathrm{im}}_{\mathrm{b},\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, ij\right) = \Pr\left\{ \underbrace{\sum_s \sum_{vw} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right)}_{\text{resources prov. by } ij} \geqslant a_{\mathrm{b}}\left(ij\right) - \underbrace{B_{\mathrm{b},\ell}\left(ij\right)}_{\text{bkgd serv. at } ij} \right\} \leqslant \underbrace{\overline{p}^{\mathrm{im}}}_{\text{required max ImP}}.$$

Relaxed constraints:

$$\sum_s \sum_v \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right) \leqslant a_n\left(i\right) - \left[\overline{B}_{n,\ell}\left(i\right) + \gamma_{\mathrm{B},\ell}\widetilde{B}_{n,\ell}\left(i\right)\right], \forall \ell > k, i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_s \sum_{vw} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \leqslant a_{\mathrm{b}}\left(ij\right) - \underbrace{\left[\overline{B}_{\mathrm{b},\ell}\left(ij\right) + \gamma_{\mathrm{B},\ell}\widetilde{B}_{\mathrm{b},\ell}\left(ij\right)\right]}_{\text{background services}}, \forall \ell > k, ij \in \mathcal{E}.$$

$\overline{B}$, $\widetilde{B}$: mean and standard deviation of resource consumption of background services

$\gamma_{\mathrm{B},\ell}$: safety factor

(4) Anticipated provisioning | ImP constraint

Impact Probability (ImP) constraints on background services at infrastructure node i and link ij:

$$p^{\mathrm{im}}_{n,\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, i\right) = \Pr\left\{ \underbrace{\sum_s \sum_v \kappa_{s,\ell}(i,v)\, r_{s,n}(v)}_{\text{resources prov. by } i} \geqslant a_n(i) - \underbrace{B_{n,\ell}(i)}_{\text{bkgd serv. at } i} \right\} \leqslant \underbrace{\overline{p}^{\mathrm{im}}}_{\text{required max ImP}},$$

$$p^{\mathrm{im}}_{\mathrm{b},\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, ij\right) = \Pr\left\{ \underbrace{\sum_s \sum_{vw} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw)}_{\text{resources prov. by } ij} \geqslant a_{\mathrm{b}}(ij) - \underbrace{B_{\mathrm{b},\ell}(ij)}_{\text{bkgd serv. at } ij} \right\} \leqslant \underbrace{\overline{p}^{\mathrm{im}}}_{\text{required max ImP}}.$$

Relaxed constraints:

$$\sum_s \sum_v \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \left[\overline{B}_{n,\ell}(i) + \gamma_{\mathrm{B},\ell}\widetilde{B}_{n,\ell}(i)\right], \forall \ell > k, i \in \mathcal{N}, n \in \Upsilon,$$

$$\sum_s \sum_{vw} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \underbrace{\left[\overline{B}_{\mathrm{b},\ell}(ij) + \gamma_{\mathrm{B},\ell}\widetilde{B}_{\mathrm{b},\ell}(ij)\right]}_{\text{background services}}, \forall \ell > k, ij \in \mathcal{E}.$$

$\overline{B}$, $\widetilde{B}$: mean and standard deviation of resource consumption of background services
$\gamma_{\mathrm{B},\ell}$: safety factor

NOKIA Bell Labs

Earnings of InP = costs charged to MNOs

Max-min problem: InP tries to

- maximize the nb of accepted slices
- minimize the cost for MNOs

Max-min anticipated provisioning problem

$$\max_{\mathrm{d}_{\mathcal{R}_k}} \min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} \quad E_k^{\mathrm{InP}}\left(\mathrm{d}_{\mathcal{R}_k}, \boldsymbol{\kappa}_{\mathcal{R}_k}\right) = \sum_{s \in \mathcal{R}_k} \sum_{\ell \in \mathcal{K}_s} \left(\underbrace{C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right)}_{\text{resource}} + \underbrace{C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right)}_{\text{node disposal}} + \underbrace{C_{\mathrm{a}}\left(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell-1}\right)}_{\text{adaptation}} \right)$$

subject to service satisfaction probability constraints

impact on background services constraints

flow conservation constraint

NOKIA Bell Labs

Approaches

Sequential: solve problem for each slice s with initial $d_s = 1$

- $\min_{\boldsymbol{\kappa}_{s_1}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{s_1}, 1\right), \min_{\boldsymbol{\kappa}_{s_2}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{s_2}, 1\right), \ldots$

Joint: start with $\mathrm{d}_{\mathcal{R}_k} = 1$ $(d_s = 1, \forall s)$, then refuse progressively low-prioritized slices if no solution found

- iteration 1: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 1, 1\}\right)$
- iteration 2: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 1, 0\}\right)$
- iteration 3: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 0, 0\}\right)$
- ... until a feasible solution found

NOKIA Bell Labs

Approaches

Sequential: solve problem for each slice s with initial $d_s = 1$

- $\min_{\boldsymbol{\kappa}_{s_1}} E_k^{\mathrm{InP}}(\boldsymbol{\kappa}_{s_1}, 1)$, $\min_{\boldsymbol{\kappa}_{s_2}} E_k^{\mathrm{InP}}(\boldsymbol{\kappa}_{s_2}, 1)$, ...

Joint: start with $\mathrm{d}_{\mathcal{R}_k} = 1$ $(d_s = 1, \forall s)$, then refuse progressively low-prioritized slices if no solution found

- iteration 1: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 1, 1\}\right)$
- iteration 2: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 1, 0\}\right)$
- iteration 3: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 0, 0\}\right)$
- ... until a feasible solution found

NOKIA Bell Labs

Approaches

Sequential: solve problem for each slice s with initial $d_s = 1$

- $\min_{\boldsymbol{\kappa}_{s_1}} E_k^{\text{InP}}(\boldsymbol{\kappa}_{s_1}, 1)$, $\min_{\boldsymbol{\kappa}_{s_2}} E_k^{\text{InP}}(\boldsymbol{\kappa}_{s_2}, 1)$, ...

Joint: start with $\mathrm{d}_{\mathcal{R}_k} = 1$ ($d_s = 1$, $\forall s$), then refuse progressively low-prioritized slices if no solution found

- iteration 1: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\text{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 1, 1\}\right)$
- iteration 2: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\text{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 1, 0\}\right)$
- iteration 3: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\text{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 0, 0\}\right)$
- ... until a feasible solution found

Approaches

Sequential: solve problem for each slice s with initial $d_s = 1$

- $\min_{\boldsymbol{\kappa}_{s_1}} E_k^{\text{InP}}(\boldsymbol{\kappa}_{s_1}, 1)$, $\min_{\boldsymbol{\kappa}_{s_2}} E_k^{\text{InP}}(\boldsymbol{\kappa}_{s_2}, 1)$, ...

Joint: start with $\mathrm{d}_{\mathcal{R}_k} = 1$ ($d_s = 1$, $\forall s$), then refuse progressively low-prioritized slices if no solution found

- iteration 1: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\text{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 1, 1\}\right)$
- iteration 2: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\text{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 1, 0\}\right)$
- iteration 3: $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\text{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \{1, 1, \cdots, 0, 0\}\right)$
- ... until a feasible solution found

NOKIA Bell Labs

Slice model:

- Three types of slice
  - HD videos streaming, SSP constraint: $\underline{p}_s^{\mathsf{sp}} = 0.99$
  - SD video streaming, SSP constraint: $\underline{p}_s^{\mathsf{sp}} = 0.95$
  - video surveillance and traffic monitoring, SSP constraint: $\underline{p}_s^{\mathsf{sp}} = 0.9$
- Slice type and demand pattern chosen uniformly at random.
- Number of provisioning request arrivals/time slot $\mathrm{Pois}(1)$
- Arrival time uniformly distributed within each time slot
- Activation delay $\mathcal{U}(1,6)$, lifetime $\mathcal{U}(1,3)$

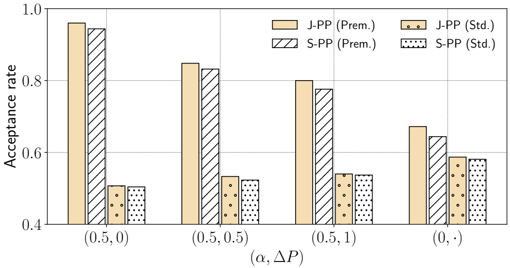
Background services:

- consume on average 20% of available infrastructure resources (with 5% std. dev.)
- ImP constraint: $\bar{p}^{\mathsf{im}} = 0.1$

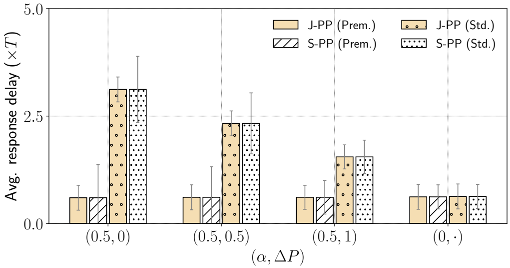
NOKIA Bell Labs

Multiple slice provisioning

J-PP vs. S-PP with different processing strategies $(\alpha, \Delta P)$



acceptance rate

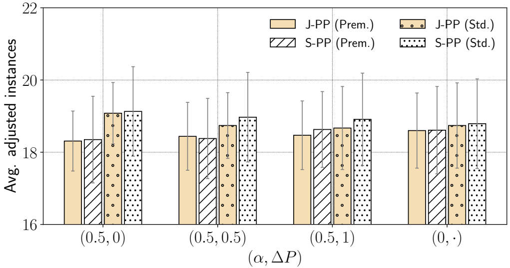


average response delay

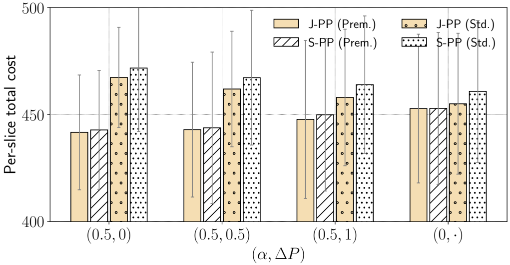
NOKIA Bell Labs

Multiple slice provisioning

J-PP vs. S-PP with different processing strategies $(\alpha, \Delta P)$



average adjusted instances

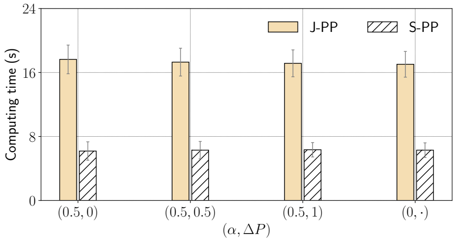


average cost per slice

Objective: to minimize cost and nb of adjusted instances

Multiple slice provisioning

J-PP vs. S-PP with different processing strategies $(\alpha, \Delta P)$



computing time

Two prioritized slice resource provisioning (PP) variants to

- maximize acceptance rate,
- maximize earnings of the InP + minimize costs charged to MNO

Advantages

- adapt to dynamic nature of slice provisioning requests;
- robustness to uncertainties.

Outline

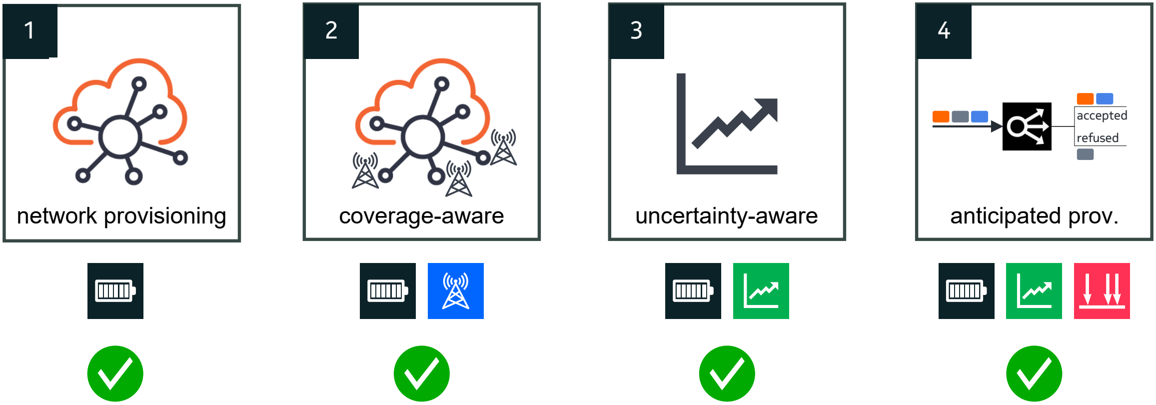
NOKIA Bell Labs

Conclusions and Perspective

Contributions

Part I. Deterministic slice resource demands

Part II. Random slice resource demands



Conclusions and Perspective

NOKIA Bell Labs

Perspectives

- Additional constraints, *e.g.*, latency, end-to-end error rate probability
- Efficient heuristics to scale up to more realistic setups
  - *eigen-decompostion* [Mechtri'16]
  - *column generation* [Nemhauser'12]
- Multiple domains network slicing (*e.g.*, multi-InPs)
  - each InP finds its own resource provisioning solution
  - afterwards: coordination between InPs to obtain final provisioning solution
- Slot-by-slot provisioning: instead of considering all active time slots simultaneously, provision resources for the slice slot-by-slot $\rightarrow$ reduce time complexity

Conclusions and Perspective

NOKIA Bell Labs

Perspectives

- Additional constraints, *e.g.*, latency, end-to-end error rate probability
- Efficient heuristics to scale up to more realistic setups
  - *eigen-decompostion* [Mechtri'16]
  - *column generation* [Nemhauser'12]
- Multiple domains network slicing (*e.g.*, multi-InPs)
  - each InP finds its own resource provisioning solution
  - afterwards: coordination between InPs to obtain final provisioning solution
- Slot-by-slot provisioning: instead of considering all active time slots simultaneously, provision resources for the slice slot-by-slot → reduce time complexity

Conclusions and Perspective

Perspectives

- Additional constraints, *e.g*., latency, end-to-end error rate probability
- Efficient heuristics to scale up to more realistic setups
  - *eigen-decompostion* [Mechtri'16]
  - *column generation* [Nemhauser'12]
- Multiple domains network slicing (*e.g*., multi-InPs)
  - each InP finds its own resource provisioning solution
  - afterwards: coordination between InPs to obtain final provisioning solution
- Slot-by-slot provisioning: instead of considering all active time slots simultaneously, provision resources for the slice slot-by-slot → reduce time complexity

Publications

Journal papers

(J1) Admission Control and Resource Provisioning for Prioritized Slice Requests with Uncertainties (IEEE TNSM 2021, submitted)

(J2) Uncertainty-Aware Resource Provisioning for Network Slicing (IEEE TNSM 2021, published)

(J3) Coverage-Aware Resource Provisioning Method for Network Slicing (IEEE/ACM TNET 2020, published)

Conference/Workshop Papers

(C1) Foresighted Resource Provisioning for Network Slicing (IEEE HPSR 2021, to appear).

(C2) Radio Resource Provisioning for Network Slicing with Coverage Constraints (IEEE ICC 2020, published)

(C3) Aggregated Resource Provisioning for Network Slices (IEEE GLOBECOM 2018, published)

Project Deliverable

(D1) Virtual Network Orchestration Framework and Algorithms (ANR MAESTRO-5G Project Deliverable D3.1, 2020)

Patents

(P1) Method and Apparatus for Mapping Network Slices Onto Network Infrastructures With SLA Guarantee. WIPO Patent No. WO2020114608A1 (filed on Dec. 07, 2018 by Nokia Solutions and Networks)

BACK UP

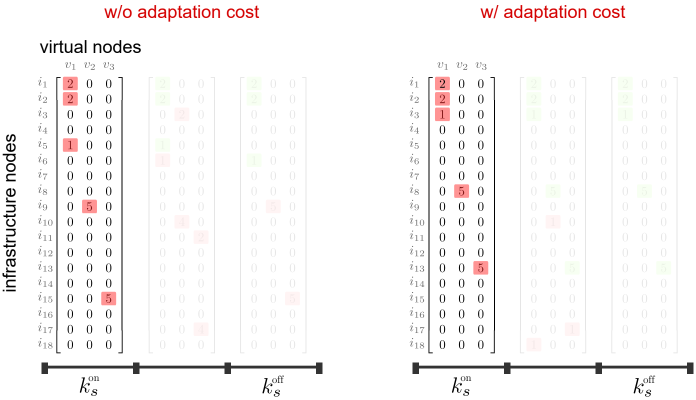
(4) Anticipated provisioning | Evaluation

Single slice provisioning

$$E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \mathrm{d}_{\mathcal{R}_k}\right) = \sum_{s\in\mathcal{R}_k}\sum_{\ell\in\mathcal{K}_s}\Big(\underbrace{C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right)}_{\text{resource}} + \underbrace{C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right)}_{\text{node disposal}} + \underbrace{C_{\mathrm{a}}\left(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell-1}\right)}_{\text{adaptation}}\Big)$$

Changes of $\kappa_{s,\ell}(i,v)$ along with ℓ:

- red: increase of $\kappa_{s,\ell}(i,v)$
- green: $\kappa_{s,\ell}(i,v)$ remains the same



(4) Anticipated provisioning | Evaluation

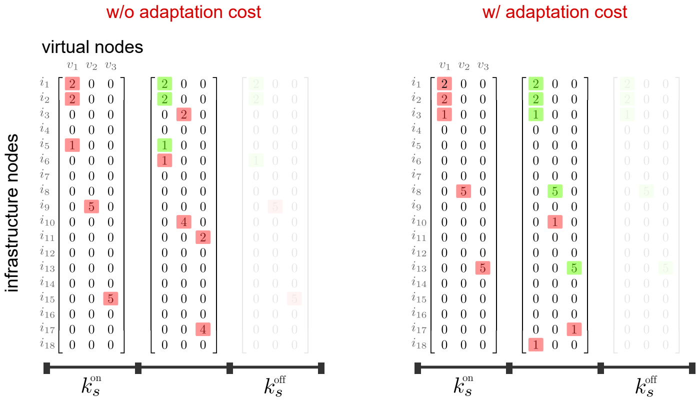
NOKIA Bell Labs

Single slice provisioning

$$E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \mathrm{d}_{\mathcal{R}_k}\right) = \sum_{s\in\mathcal{R}_k}\sum_{\ell\in\mathcal{K}_s}\left(\underbrace{C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right)}_{\text{resource}} + \underbrace{C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right)}_{\text{node disposal}} + \underbrace{C_{\mathrm{a}}\left(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell-1}\right)}_{\text{adaptation}}\right)$$

Changes of $\kappa_{s,\ell}(i,v)$ along with ℓ:

- red: increase of $\kappa_{s,\ell}(i,v)$
- green: $\kappa_{s,\ell}(i,v)$ remains the same

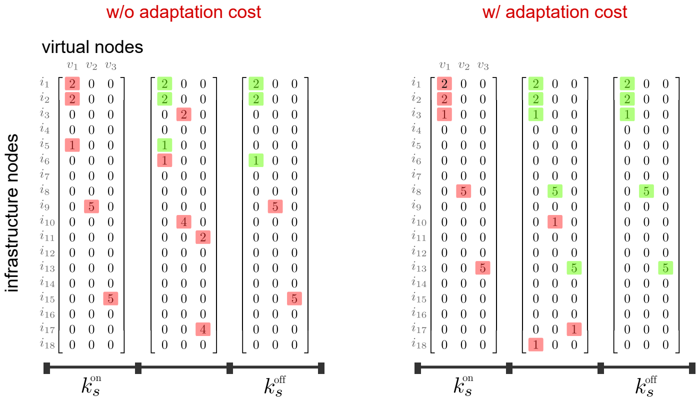


Single slice provisioning

$$E_k^{\mathsf{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, \mathsf{d}_{\mathcal{R}_k}\right) = \sum_{s\in\mathcal{R}_k}\sum_{\ell\in\mathcal{K}_s}\left(\underbrace{C_{\mathsf{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right)}_{\text{resource}} + \underbrace{C_{\mathsf{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right)}_{\text{node disposal}} + \underbrace{C_{\mathsf{a}}\left(\boldsymbol{\kappa}_{s,\ell}, \boldsymbol{\kappa}_{s,\ell-1}\right)}_{\text{adaptation}}\right)$$

Changes of $\kappa_{s,\ell}(i,v)$ along with ℓ:

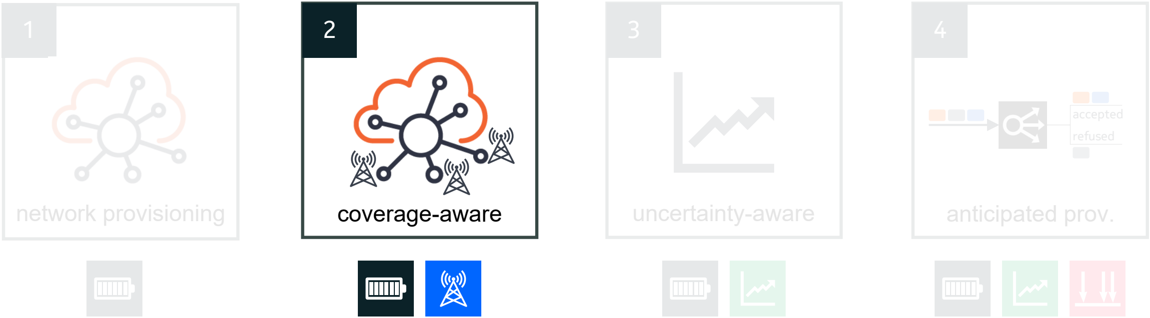
- red: increase of $\kappa_{s,\ell}(i,v)$
- green: $\kappa_{s,\ell}(i,v)$ remains the same



Contributions

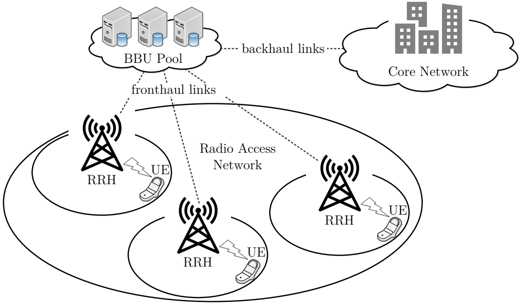
Part I. Deterministic slice resource demands

Part II. Random slice resource demands



NOKIA Bell Labs

- **RP**: **R**adio resource **P**rovisioning: radio
- **NP**: (other) **N**etwork resource **P**rovisioning: computing, storage, and bandwidth



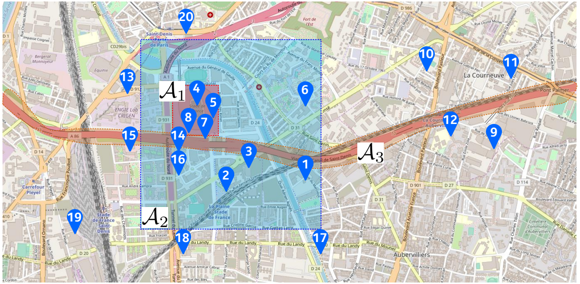
Cloud Radio Access Network (CRAN) general architecture

- RRH: Remote Radio Head
- BBU: Base Band Unit

(2) Coverage-aware | Radio provisioning (RP)

Problem statement

- SM-SLA involves:
  - coverage constraints: geographical distribution slice users. This distribution is described by the user density function $\rho_s(x)$, with $x \in \mathcal{A}$.
  - supported service type
  - targeted QoS such as a minimum average data rate $U_{s,\mathrm{u}}$ and $U_{s,\mathrm{d}}$ for the wireless uplink and downlink traffic for each user



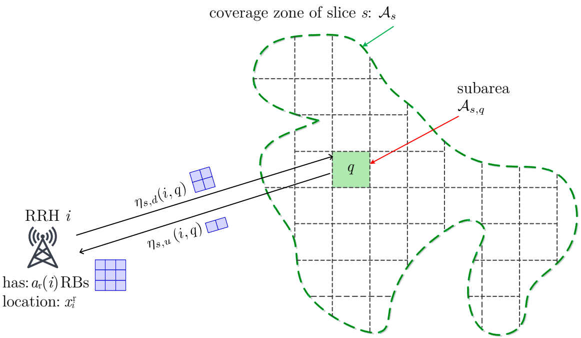
$\mathcal{A}$: the geographical area under study,

$\mathcal{A}_s$: the subarea over which service s has to be made available

NOKIA Bell Labs

Proposed approach for RP

- For slice s, InP has to provide a minimum data rate $(U_{s,\mathrm{u}}, U_{s,\mathrm{d}})$ to each user spread over $\mathcal{A}_s$ with density $\rho_s(x)$
- Each RRH has a fixed amount $a_\mathrm{r}(i)$ of resource blocks (RB) per time unit



- $\mathcal{A}_s$ is partitioned into Q_s convex subareas $\mathcal{A}_{s,q}$, $q \in \mathcal{Q}_s = \{1, \ldots, Q_s\}$
- Instead of allocating RBs to users, RRH nodes allocate RBs to subareas

(2) Coverage-aware | Radio provisioning (RP)

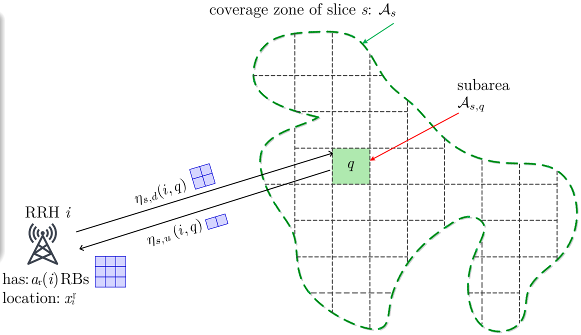
NOKIA Bell Labs

RP cost function

Cost function of RP problem

$$c_{\mathrm{rr}}(\boldsymbol{\eta}) = \sum_{\sigma \in \mathcal{S}} \sum_{i \in \mathcal{N}_{\mathrm{r}}} c_{\mathrm{f}}(i)\, \widetilde{\eta}_s(i)$$
$$+ \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{N}_{\mathrm{r}}} \sum_{q \in \mathcal{Q}_s} \left[c_{\mathrm{r}}(i) - \lambda b_{\mathrm{u}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)\right] a_{\mathrm{r}}(i)\, \eta_{s,\mathrm{u}}(i,q)$$
$$+ \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{N}_{\mathrm{r}}} \sum_{q \in \mathcal{Q}_s} \left[c_{\mathrm{r}}(i) - \lambda b_{\mathrm{d}}\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right)\right] a_{\mathrm{r}}(i)\, \eta_{s,\mathrm{d}}(i,q)$$

- $\eta_s(i,q) \in [0,1]$: proportion of RBs that RRH i provisions to subarea q
- $\lambda b\left(x_i^{\mathrm{r}}, \mathcal{A}_{s,q}\right) a_{\mathrm{r}}(i) \eta_s(i,q)$: rate-related discount to penalty the use of far RRHs



(2) Coverage-aware | Radio provisioning (RP)

RP main constraints

(1) RRH capacity limit: $\forall i \in \mathcal{N}_\mathrm{r}$,

$$\sum_{s \in \mathcal{S}} \sum_{q \in \mathcal{Q}_s} \left(\eta_{s,\mathrm{u}}(i,q) + \eta_{s,\mathrm{d}}(i,q) \right) \leqslant 1,$$

(2) Rate requirement per user: $\forall s \in \mathcal{S}, q \in \mathcal{Q}_s$,

$$\sum_{i \in \mathcal{N}_\mathrm{r}} \eta_{s,\mathrm{u}}(i,q)\, a_\mathrm{r}(i)\, b_\mathrm{u}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right) \geqslant U_{s,\mathrm{u}} \int_{\mathcal{A}_{s,q}} \rho_s(x)\,\mathrm{d}x,$$

$$\sum_{i \in \mathcal{N}_\mathrm{r}} \eta_{s,\mathrm{d}}(i,q)\, a_\mathrm{r}(i)\, b_\mathrm{d}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right) \geqslant U_{s,\mathrm{d}} \int_{\mathcal{A}_{s,q}} \rho_s(x)\,\mathrm{d}x,$$

(3) Rate requirement per slice: $\forall s \in \mathcal{S}$,

$$\sum_{q \in \mathcal{Q}_s} \sum_{i \in \mathcal{N}_\mathrm{r}} \eta_{s,\mathrm{u}}(i,q)\, a_\mathrm{r}(i)\, b_\mathrm{u}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right) \geqslant R_{s,\mathrm{u}}(v_\mathrm{r}),$$

$$\sum_{q \in \mathcal{Q}_s} \sum_{i \in \mathcal{N}_\mathrm{r}} \eta_{s,\mathrm{d}}(i,q)\, a_\mathrm{r}(i)\, b_\mathrm{d}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right) \geqslant R_{s,\mathrm{d}}(v_\mathrm{r}),$$

(4) UL-DL proportionality: $\forall q \in \mathcal{Q}_s$,

$$\frac{\eta_{s,\mathrm{u}}(i,q)\, a_\mathrm{r}(i)\, b_\mathrm{u}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right)}{R_{s,\mathrm{u}}(v_\mathrm{r})} = \frac{\eta_{s,\mathrm{d}}(i,q)\, a_\mathrm{r}(i)\, b_\mathrm{d}\left(x_i^\mathrm{r}, \mathcal{A}_{s,q}\right)}{R_{s,\mathrm{d}}(v_\mathrm{r})}.$$

(2) Coverage-aware | Single-step vs two-step provisioning

NOKIA Bell Labs

- Global provisioning problem (single-step):

$$\begin{aligned} \min \quad & c_{\text{tot}}(\boldsymbol{\eta}, \boldsymbol{\Phi}) = c_{\text{rr}}(\boldsymbol{\eta}) + c_{\text{wr}}(\boldsymbol{\Phi}) \qquad (1) \\ \text{s.t.} \quad & \text{NP and RP constraints} \end{aligned}$$

- When the size of $\boldsymbol{\Phi}$ and $\boldsymbol{\eta}$ increases, the problem may become intractable

→ Two-step provisioning algorithm: CARP (Coverage-Aware Resource Provisioning), which minimizes both terms of (1) separately.

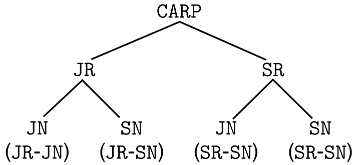


Figure: Four variants of CARP.

NOKIA Bell Labs

Infrastructure model:

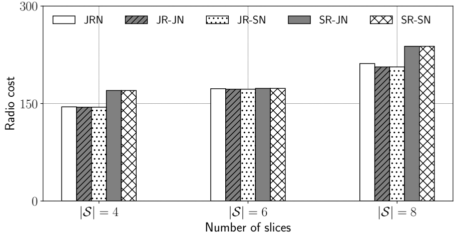
- k-ary fat tree topology
  - Leasing cost for each resource unit $= 1$
  - For coverage constraints, we consider a $1.43\,\mathrm{km} \times 4.95\,\mathrm{km}$ area around the Stade de France in Seine-Saint-Denis

Slices: three types:

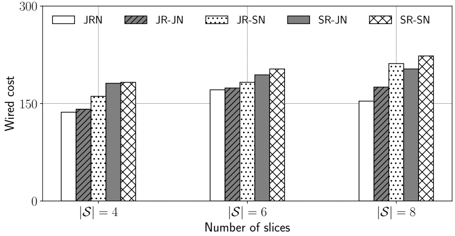
- HD video streaming
- SD video streaming
- Video surveillance and traffic monitoring

(2) Coverage-aware | Evaluation

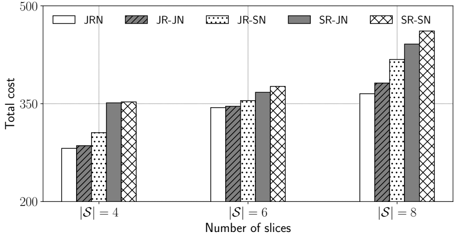
Performance of JRN vs 4 CARP variants.



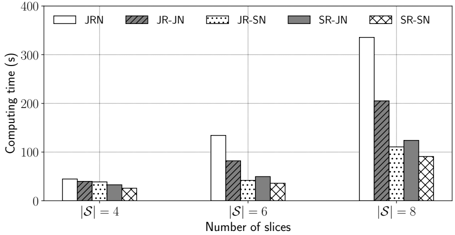
(a) Radio cost



(b) Wired cost



(c) Total cost



(d) Computing time

- Extension to Contribution 1: Slice resource provisioning accounting coverage constraints:
  - provide a minimum data rate for users in the geographical areas covered by slices
- Two-step approach to decompose the global optimization problem (`JRN`)
- `JRN` vs 4 variants of `CARP`:
  - least provisioning cost
  - but requires a much larger time complexity

Contributions

Part I. Deterministic slice resource demands

Part II. Random slice resource demands

1 network provisioning

2 coverage-aware

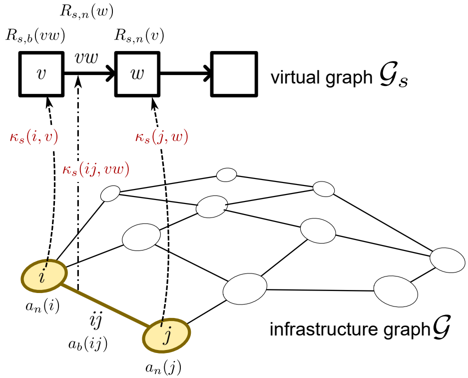
3 uncertainty-aware

4 anticipated prov.

- Uncertainties may lead to slice QoS $<$ expected level
  - S-RDs: traffic dynamics, *e.g.*, flow arrival/departure
  - available infrastructure resources
- This contribution:
  - provisions infrastructure resources for network slices, while being robust to a partly unknown number of users with a random usage of the slice resources.
  - guarantees a predefined service satisfaction probability (SSP)
  - moreover, provisioning is performed so as to limit its impact on background services

Resource provisioning for slice s

- $\kappa_s(i, v) \in \mathbb{N}$, $\kappa_s(ij, vw) \in \mathbb{N}$
- $\widetilde{\kappa}_s(i, v) \in \{0, 1\}$ node mapping indicator function.
- $\widetilde{\kappa}(i) \in \{0, 1\}$ indicator function for use of infrastructure node i.



NOKIA Bell Labs

Slice resource demand constraints

Static constraints: S-RD $\mathrm{R}_s = \left(R_{s,n}(v), R_{s,\mathrm{b}}(vw)\right)^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s}$ is deterministic

$$\sum_{i\in\mathcal{N}} \kappa_s(i,v)\, r_{s,n}(v) \geqslant R_{s,n}(v), \forall s\in\mathcal{S}, v\in\mathcal{N}_s, n\in\Upsilon$$

$$\sum_{ij\in\mathcal{E}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant R_{s,\mathrm{b}}(vw), \forall s\in\mathcal{S}, vw\in\mathcal{E}_s$$

Uncertainty-aware constraints: S-RD R_s is random $\rightarrow$ Service Satisfaction Probability (SSP) constraint:

$$p_s(\boldsymbol{\kappa}_s) = \Pr\left\{\begin{array}{l} \sum_{i\in\mathcal{N}} \kappa_s(i,v)\, r_{s,n}(v) \geqslant R_{s,n}(v), \forall v, n, \\ \sum_{ij\in\mathcal{E}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant R_{s,\mathrm{b}}(vw), \forall vw \end{array}\right\} \geqslant \underline{p}_s, \forall s\in\mathcal{S}.$$

$\underline{p}_s$: minimum SSP for slice s.

NOKIA Bell Labs

Slice resource demand constraints

Static constraints: S-RD $\mathrm{R}_s = \left(R_{s,n}(v), R_{s,\mathrm{b}}(vw)\right)^{\top}_{n \in \Upsilon, (v,vw) \in \mathcal{G}_s}$ is deterministic

$$\sum_{i \in \mathcal{N}} \kappa_s(i,v)\, r_{s,n}(v) \geqslant R_{s,n}(v), \forall s \in \mathcal{S}, v \in \mathcal{N}_s, n \in \Upsilon$$

$$\sum_{ij \in \mathcal{E}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant R_{s,\mathrm{b}}(vw), \forall s \in \mathcal{S}, vw \in \mathcal{E}_s$$

Uncertainty-aware constraints: S-RD R_s is random $\rightarrow$ Service Satisfaction Probability (SSP) constraint:

$$p_s(\boldsymbol{\kappa}_s) = \Pr\left\{ \begin{array}{l} \sum_{i \in \mathcal{N}} \kappa_s(i,v)\, r_{s,n}(v) \geqslant R_{s,n}(v), \forall v, n, \\ \sum_{ij \in \mathcal{E}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant R_{s,\mathrm{b}}(vw), \forall vw \end{array} \right\} \geqslant \underline{p}_s, \forall s \in \mathcal{S}.$$

$\underline{p}_s$: minimum SSP for slice s.

NOKIA Bell Labs

Slice resource demand constraints

Relaxed SSP constraints:

$$\sum_{i\in\mathcal{N}} \kappa_s(i,v)\, r_{s,n}(v) \geqslant \overline{R}_{s,n}(v) + \gamma_s \widetilde{R}_{s,n}(v), \forall s \in \mathcal{S}, v \in \mathcal{N}_s, n \in \Upsilon$$

$$\sum_{ij\in\mathcal{E}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant \overline{R}_{s,\mathrm{b}}(vw) + \gamma_s \widetilde{R}_{s,\mathrm{b}}(vw), \forall s \in \mathcal{S}, vw \in \mathcal{E}_s$$

The SSP becomes

$$p_s(\gamma_s) = \Pr\left\{ \begin{array}{l} \widehat{R}_{s,n}(v,\gamma_s) \geqslant R_{s,n}(v), \forall s \in \mathcal{S}, v \in \mathcal{N}_s, n \in \Upsilon \\ \widehat{R}_{s,\mathrm{b}}(vw,\gamma_s) \geqslant R_{s,\mathrm{b}}(vw), \forall s \in \mathcal{S}, vw \in \mathcal{E}_s \end{array} \right\}$$

A sufficiently large γ_s would satisfy the SSP constraint.

NOKIA Bell Labs

Slice resource demand constraints

Finding γ_s: The smallest smallest value of γ_s such that $p_s(\gamma_s) \geqslant \underline{p}_s$

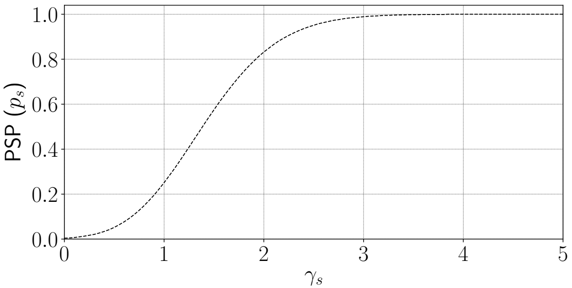


Figure: Evolution of p_s as function of γ_s.

γ_s can then be found by applying a bisection search[1].

[1] R. L. Burden and J. Douglas Faires, *Numerical Analysis*, 9th ed. Brooks/Cole, Cengage Learning, 2011.

NOKIA Bell Labs

Impact on background service (ImP) constraints

Static constraints: Infrastructure resources $(a_n(i), a_\mathrm{b}(ij))$ are deterministic

$$\sum_{s\in S}\sum_{v\in\mathcal{N}_s}\kappa_s(i,v)\,r_{s,n}(v)\leqslant a_n(i),\forall i\in\mathcal{N},n\in\Upsilon,$$

$$\sum_{s\in S}\sum_{vw\in\mathcal{E}_s}\kappa_s(ij,vw)\,r_{s,\mathrm{b}}(vw)\leqslant a_\mathrm{b}(ij),\forall ij\in\mathcal{E},$$

Uncertainty-aware constraints: $(a_n(i), a_\mathrm{b}(ij))$ are random $\rightarrow$ Impact Probabilities (ImPs) constraints:

$$p_n^{\mathrm{im}}(i)\leqslant\overline{p}^{\mathrm{im}},\forall i\in\mathcal{N},n\in\Upsilon,$$

$$p_\mathrm{b}^{\mathrm{im}}(ij)\leqslant\overline{p}^{\mathrm{im}},\forall ij\in\mathcal{E},$$

$\overline{p}^{\mathrm{im}}$: minimum ImP on background services.

Relaxed ImP constraints:

$$\sum_{s,v}\kappa_s(i,v)\,r_{s,n}(v)\leqslant a_n(i)-\left[\overline{B}_n(i)+\gamma_\mathrm{B}\widetilde{B}_n(i)\right],\forall n\in\Upsilon,i\in\mathcal{N},$$

$$\sum_{s,vw}\kappa_s(ij,vw)\,r_{s,\mathrm{b}}(vw)\leqslant a_\mathrm{b}(ij)-\left[\overline{B}_\mathrm{b}(ij)+\gamma_\mathrm{B}\widetilde{B}_\mathrm{b}(ij)\right],\forall ij\in\mathcal{E},$$

$\rightarrow$ Optimality: $\gamma_\mathrm{B}=\Phi^{-1}\left(1-\overline{p}^{\mathrm{im}}\right)$ guarantees $p_n^{\mathrm{im}}(i)\leqslant\overline{p}^{\mathrm{im}}$, $\forall i\in\mathcal{N},n\in\Upsilon$, and $p_\mathrm{b}^{\mathrm{im}}(ij)\leqslant\overline{p}^{\mathrm{im}}$, $\forall ij\in\mathcal{E}$.

NOKIA Bell Labs

Uncertainty-aware provisioning for multiple slices

$$\underset{\{\boldsymbol{d},\boldsymbol{\kappa}\}=\{d_s,\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}}{\text{maximize}} \quad \sum_{s\in\mathcal{S}} \left(l_s d_s - C_s\left(\boldsymbol{\kappa}_s\right)\right),$$

$$\text{subject to} \sum_{i\in\mathcal{N}} \kappa_s\left(i,v\right) r_{s,n}\left(v\right) \geqslant \widehat{R}_{s,n}\left(v,\gamma_s\right) d_s, \forall s\in\mathcal{S}, v\in\mathcal{N}_s, n\in\Upsilon$$

$$\sum_{ij\in\mathcal{E}} \kappa_s\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \geqslant \widehat{R}_{s,\mathrm{b}}\left(vw,\gamma_s\right) d_s, \forall s\in\mathcal{S}, vw\in\mathcal{E}_s,$$

$$\sum_{s\in\mathcal{S}}\sum_{v\in\mathcal{N}_s} \kappa_s\left(i,v\right) r_{s,n}\left(v\right) \leqslant a_n\left(i\right) - \widehat{B}_n\left(i,\gamma_{\mathrm{B}}\right), \forall i\in\mathcal{N}, n\in\Upsilon,$$

$$\sum_{s\in\mathcal{S}}\sum_{vw\in\mathcal{E}_s} \kappa_s\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \leqslant a_{\mathrm{b}}\left(ij\right) - \widehat{B}_{\mathrm{b}}\left(ij,\gamma_{\mathrm{B}}\right), \forall ij\in\mathcal{E}.$$

flow conservation constraint

Table: Provisioning approaches.

	Jointly	Sequentially
background services not taken into account	JP	SP
background services accounted	JP-B	SP-B

- Background services: consume on average 20% of available infrastructure resources (5% std. dev.)
- Types of slices:
  (1) HD video streaming at 4 Mbps. $N_s \sim \mathcal{B}(300, 0.9)$, $l_s = 900$, $\underline{p}_s = 0.99$
  (2) SD video streaming at 2 Mbps. $N_s \sim \mathcal{B}(1000, 0.8)$, $l_s = 1000$, $\underline{p}_s = 0.95$
  (3) Video surveillance and traffic monitoring at 1 Mbps. $N_s = 50$, $l_s = 800$, $\underline{p}_s = 0.9$

NOKIA Bell Labs

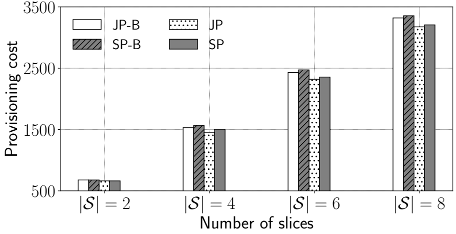
- Provisioning of a single slice of Type 1, with $\underline{p}_s = 0.99$ and $\overline{p}^{\text{im}} = 0.1$
- SP-B vs and SP (w/ and w/o background traffic):
  - SP has a lower provisioning cost, and yields a higher earning, but has a higher impact on background services

Table: SP-B vs and SP (w/ and w/o background traffic).

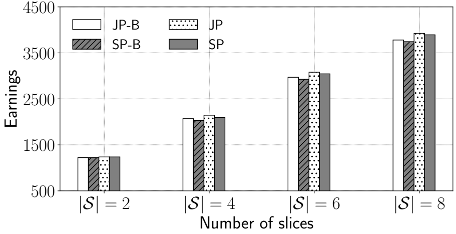
Criteria	SP-B	SP
Maximal p^{im}	1.26e-4	0.58
Provisioning cost	332	326
Total earnings	568	574
#impacted nodes	0	1
#impacted links	0	0

(3) Uncertainty-aware | Evaluation

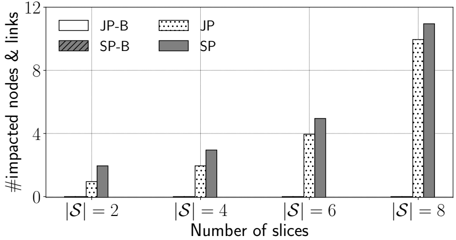
Provisioning of several slices of different types



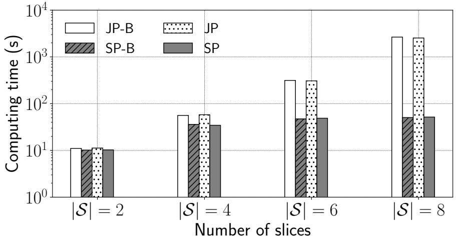
(a) Provisioning cost



(b) Earnings



(c) #impacted nodes & links



(d) Computing time

(3) Uncertainty-aware | Evaluation

Benefits of the uncertainty-aware slice resource provisioning

- UPE: uncertainty-aware provisioning and embedding solution
- DPE: deterministic provisioning and embedding solution
- S-RD: nb of users associated to the slice follows a binomial distribution $\mathcal{B}(m, p)$
  - m is fixed to 300
  - p varies from 0.4 to 0.9

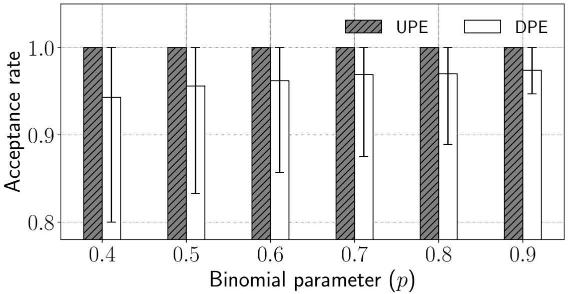


Figure: UPE vs DPE in terms of SFC acceptance rate.

- Resource provisioning method for network slicing that maximizes
  - acceptance rate,
  - earnings of the InP
- Advantages
  - robustness to uncertainties
  - keep the impact on the background services under a threshold imposed by the InP.

References

Cohen, R., Lewin-Eytan, L., Naor, J. S., and Raz, D. (2015).
Near Optimal Placement of Virtual Network Functions.
In *Proc. IEEE INFOCOM*, pages 1346–1354.

GSM Alliance (2017).
An Introduction to Network Slicing.
White Paper.

Liang, C. and Yu, F. R. (2014).
Wireless Network Virtualization: A Survey, Some Research Issues and Challenges.
IEEE Commun. Surveys Tuts., pages 1–24.

Liu, J., Lu, W., Zhou, F., Lu, P., and Zhu, Z. (2017).
On Dynamic Service Function Chain Deployment and Readjustment.
IEEE Trans. Netw. Service Manag., 14(3):543–553.

Mechtri, M., Ghribi, C., and Zeghlache, D. (2016).
A Scalable Algorithm for the Placement of Service Function Chains.
IEEE Trans. Netw. Service Manag., 13(3):533–546.

Nemhauser, G. (2012).
Column Generation for Linear and Integer Programming.
Optimization Stories, I:65–73.

Riera, J. F., Batalle, J., Bonnet, J., Dias, M., McGrath, M., Petralia, G., Liberati, F., Giuseppi, A., Pietrabissa, A., Ceselli, A., Petrini, A., Trubian, M., Papadimitrou, P., Dietrich, D., Ramos, A., Melian, J., Xilouris, G., Kourtis, A., Kourtis, T., and Markakis, E. K. (2016).
TeNOR: Steps towards an orchestration platform for multi-PoP NFV deployment.
In *IEEE NetSoft Conference and Workshops: Software-Defined Infrastructure for Networks, Clouds, IoT and Services*, pages 243–250.

Riggio, R., Bradai, A., Harutyunyan, D., Rasheed, T., and Ahmed, T. (2016).