DPoS: Decentralized, Privacy-Preserving, and Low-Complexity Online Slicing for Multi-Tenant Networks

Hailiang ZHAO @ ZJU-CS

http://hliangzhao.me

April 19, 2021

This slide is a report on my paper *DPoS*, accepted by **IEEE Trans. on Mobile Computing**.

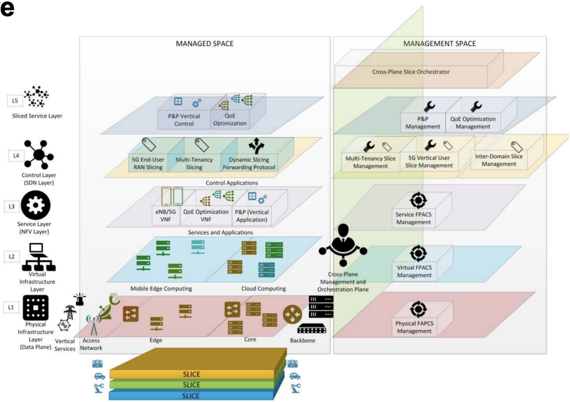
Outline

Outline

The End-to-End Network Slicing (E2E-NS) Architecture

Detailed Architecture

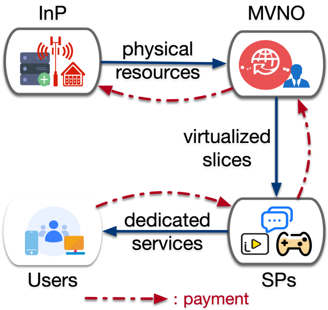
- L5 – Sliced Service Layer
- L4 – SDN Control Layer
- L3 – NFV Data Service Layer
- L2 - Virtual Infrastructure Layer
- L1 - Physical Infrastructure Layer



The picture is from *End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks*, the SliceNet Consortium of 5GPPP.

Business Models

Players involved: infrastructure providers (InPs), mobile network operators (MNOs), cloud providers (one kind of InPs actually), edge & cloud service providers (a.k.a. tenants), service subscribers (i.e. users), mobile virtual network operators (MVNOs), ...



Efficient Resource Allocation for VNFs

The key problem underlying network slicing is that

How to allocate different kind of resources to each slice, on top of the physical infrastructure, to maximize the utility of involved players?

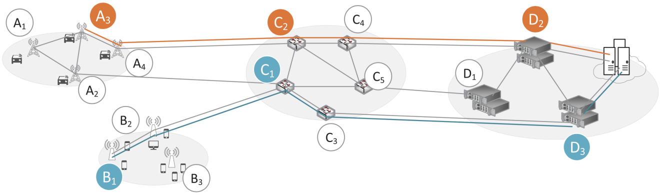
Existing approaches ...

- *Insufferable complexity*. Fine-tuned heuristics or deep machine learning models such as deep Q-network (DQN)
- *Privacy leakage*. The centralized algorithms are generally built on the com- plete knowledge regarding all preferences of involved busi- ness players, including the monetary budget of tenants, the number and purchasing-power of service subscribers, etc
- *Offline settings*. All tenants sit together to bid. The MVNO knows the willingness to bid of tenants and many other private information of all tenants during each bidding round

Outline

Network Infrastructure as a Graph

We consider the scenario where multiple network slices are built upon an SDN/NFV-enabled 5G network infrastructure.



physical resources $\longmapsto$ weighted connected graph

- $\mathcal{C} \triangleq \{1, \ldots, C\}$: the set of resources
- $\mathcal{N} \triangleq \{1, \ldots, N\}$: the set of tenants (each requires one slice)
- $\{d_n^c\}_{\forall c \in \mathcal{C}}$: the requirements of the n-th tenant

$$d_n^c \begin{cases} > 0 & \text{if } c \in \mathcal{C}_n \\ = 0 & \text{otherwise} \end{cases}$$

The picture is from the paper *A Service-Oriented Deployment Policy of End-to-End Network Slicing Based on Complex Network Theory*.

Resource Demand and Estimated Revenue

- The traffic demand on the n-th slice is denoted by $\{f_s(\gamma, \tau)\}_{\forall s \in \mathcal{S}_n}$
- The *estimated* revenue of each tenant n is from the payment of its service subscribers:

$$v_n \triangleq \sum_{s \in \mathcal{S}_n} \varrho_s \cdot \sigma_n\Big(f_s(\gamma, \tau)\Big)$$

(v_n *can be interpreted as the willingness-to-pay of tenant* n)

- The bound for estimated revenue:

$$\forall c \in \mathcal{C} : \begin{cases} \underline{p_c} \leq \min_{\forall n \in \mathcal{N}: d_n^c \neq 0} e_n^c \longrightarrow \textit{true preferences} \\ \overline{p_c} \geq \max_{\forall n \in \mathcal{N}: d_n^c \neq 0} e_n^c \longrightarrow \textit{eliminate mock auctions} \end{cases}$$

Social Welfare Maximization

- Define a non-decreasing zero-startup cost function of resource c: $\forall c \in \mathcal{C}, f_c : [0,1] \to \mathbb{R}$
- Utility of the n-th tenant: $U_n \triangleq (v_n - \pi_n) \cdot x_n$
- Utility of the MVNO: $U_o \triangleq \sum_{n\in\mathcal{N}} \pi_n \cdot x_n - \sum_{c\in\mathcal{C}} f_c\Big(\sum_{n\in\mathcal{N}} d_n^c x_n \Big)$

The global *offline* social welfare maximization problem:

$$\mathcal{P}_1 : \max_{\{x_n\}_{\forall n\in\mathcal{N}}} \sum_{n\in\mathcal{N}} x_n \sum_{s\in\mathcal{S}_n} \varrho_s \cdot \sigma_n\Big(f_s(\gamma,\tau)\Big) - \sum_{c\in\mathcal{C}} f_c\Big(\sum_{n\in\mathcal{N}} d_n^c x_n\Big)$$

$$s.t. \quad \sum_{n\in\mathcal{N}} d_n^c x_n \le 1, \forall c \in \mathcal{C},$$

$$x_n \in \{0,1\}, \forall n \in \mathcal{N}.$$

Privacy Concerns

What the MVNO should know:

- the setup information $\{f_c, \underline{p_c}, \overline{p_c}\}_{\forall c \in \mathcal{C}}$
- the attributes defined in the GSTs $\{\varrho_s\}_{\forall s \in \mathcal{S}_n, \forall n \in \mathcal{N}}$

What the MVNO should ***NOT*** *know:*

- the private tuple

$$\boldsymbol{\theta} \triangleq \Big(\{\sigma_n\}_{\forall n \in \mathcal{N}}, \big\{ \{f_s(\gamma, \tau)\}_{\forall s \in \mathcal{S}_n} \rightarrow \mathcal{S}_n \big\}_{\forall n \in \mathcal{N}} \Big)$$

- the arrival sequence of tenants

Outline

The Primal-Dual Approach

The *Relaxed Primal Problem* $\mathcal{P}_2$:

$$\mathcal{P}_2 : \max_{\boldsymbol{x},\boldsymbol{y}} \sum_{n\in\mathcal{N}} x_n \sum_{s\in\mathcal{S}_n} \varrho_s \cdot \sigma_n\Big(f_s(\gamma,\tau)\Big) - \sum_{c\in\mathcal{C}} \tilde{f}_c(y_c)$$

$$s.t. \quad \sum_{n\in\mathcal{N}} d_n^c x_n \leq y_c, \forall c \in \mathcal{C},$$

$$\boldsymbol{x} \leq \mathbf{1}, \boldsymbol{x} \geq \mathbf{0}, \boldsymbol{y} \geq \mathbf{0},$$

where $\tilde{f}_c(\cdot)$ is the *extended cost functions*, defined as follows:

$$\tilde{f}_c(y) \triangleq \begin{cases} f_c(y) & \text{if } y \in [0,1] \\ +\infty & \text{if } y \in (1,+\infty). \end{cases}$$

Proposition: $\mathcal{P}_2$ is equivalent to $\mathcal{P}_1$ except the relaxation of $\{x_n\}_{\forall n\in\mathcal{N}}$.

The Primal-Dual Approach

The *Dual Problem* of $\mathcal{P}_2$:

$$\mathcal{P}_3 : \min_{\boldsymbol{p},\boldsymbol{\psi}} \sum_{n\in\mathcal{N}} \psi_n + \sum_{c\in\mathcal{C}} h_c(p_c)$$

$$s.t. \quad \psi_n \geq \sum_{s\in\mathcal{S}_n} \varrho_s \cdot \sigma_n\Big(f_s(\gamma,\tau)\Big) - \sum_{c\in\mathcal{C}} p_c d_n^c, \forall n \in \mathcal{N},$$

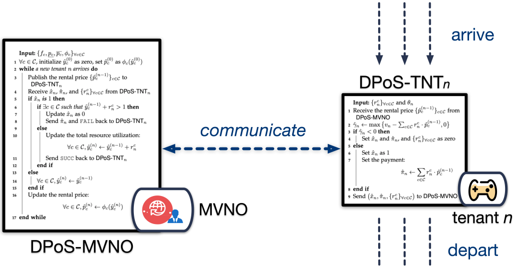
$$\boldsymbol{\psi} \geq \mathbf{0}, \boldsymbol{p} \geq \mathbf{0},$$

where $\boldsymbol{\psi} = [\psi_n]_{n\in\mathcal{N}} \in \mathbb{R}^N$ and $\boldsymbol{p} = [p_c]_{c\in\mathcal{C}} \in \mathbb{R}^C$ are the dual variables corresponding to $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Interpreting variables $\Longrightarrow$

- $\boldsymbol{x}$: the rent or out decision of each tenant
- $\boldsymbol{y}$: the quantity rent out of each resource type
- $\boldsymbol{\psi}$: the utility of the n-th tenant
- $\boldsymbol{p}$: the price of each resource type

The DPoS Algorithm

DPoS: the algorithm based on *the alternating update of primal and dual variables* in $\mathcal{P}_2$ and $\mathcal{P}_3$, respectively.



Communication Details

1. DPoS-MVNO publishes the prices
2. DPoS-TNT*n* decides to rent or not, and sends the decision and payment (if `True`) to DPoS-MVNO
3. DPoS-MVNO checks the resource surplus, and sends `SUCC` or `FAIL` back to DPoS-TNT*n*

The MVNO Side

The pseudo code of DPoS-MVNO (with $O(|\mathcal{N}| \cdot |\mathcal{C}|)$-complexity):

Algorithm 1: DPoS-MVNO

Input: $\{f_c, \underline{p_c}, \overline{p_c}, \phi_c\}_{\forall c \in \mathcal{C}}$

$\forall c \in \mathcal{C}$, initialize $\hat{y}_c^{(0)}$ as zero, set $\hat{p}_c^{(0)}$ as $\phi_c(\hat{y}_c^{(0)})$

while *a new tenant n arrives* **do**

- Publish the rental price $\{\hat{p}_c^{(n-1)}\}_{c \in \mathcal{C}}$ to DPoS-TNT$_n$
- Receive $\hat{x}_n$, $\hat{\pi}_n$, and $\{d_n^c\}_{\forall c \in \mathcal{C}}$ from DPoS-TNT$_n$
- **if** $\hat{x}_n$ *is* 1 **then**
  - **if** $\exists c \in \mathcal{C}$ *such that* $\hat{y}_c^{(n-1)} + d_n^c > 1$ **then**
    - Update $\hat{x}_n$ as 0
    - Send $\hat{\pi}_n$ and `FAIL` back to DPoS-TNT$_n$
  - **else**
    - Update the total resource utilization:

      $$\forall c \in \mathcal{C}, \hat{y}_c^{(n)} \leftarrow \hat{y}_c^{(n-1)} + d_n^c$$

    - Send `SUCC` back to DPoS-TNT$_n$
  - **end if**
- **else**
  - $\forall c \in \mathcal{C}, \hat{y}_c^{(n)} \leftarrow \hat{y}_c^{(n-1)}$
- **end if**
- Update the rental price:

  $$\forall c \in \mathcal{C}, \hat{p}_c^{(n)} \leftarrow \phi_c(\hat{y}_c^{(n)})$$

end while

The Tenant Side

The pseudo code of DPoS-TNT$_n$ (with $O(|\mathcal{C}|)$-complexity):

Algorithm 2: DPoS-TNT$_n$

Input: $\{d_n^c\}_{\forall c \in \mathcal{C}}$ and θ_n

Receive the rental price $\{\hat{p}_c^{(n-1)}\}_{c \in \mathcal{C}}$ from DPoS-MVNO

$\hat{\psi}_n \leftarrow \max\left\{v_n - \sum_{c \in \mathcal{C}} d_n^c \cdot \hat{p}_c^{(n-1)}, 0\right\}$

if $\hat{\psi}_n < 0$ **then**

  Set $\hat{x}_n$ and $\hat{\pi}_n$, and $\{d_n^c\}_{\forall c \in \mathcal{C}}$ as zero

else

  Set $\hat{x}_n$ as 1

  Set the payment:

$$\hat{\pi}_n \leftarrow \sum_{c \in \mathcal{C}} d_n^c \cdot \hat{p}_c^{(n-1)}$$

end if

Send $\left(\hat{x}_n, \hat{\pi}_n, \{d_n^c\}_{\forall c \in \mathcal{C}}\right)$ to DPoS-MVNO

DPoS-MVNO discloses the rental prices $\{\hat{p}_c^{(n-1)}\}_{c \in \mathcal{C}}$ to tenant n. Then, tenant n judges whether it has positive utility if it decides to rent. If yes, DPoS-TNT$_n$ sets the payment $\hat{\pi}_n$ as $\sum_{c \in \mathcal{C}} d_n^c \cdot \hat{p}_c^{(n-1)}$. Otherwise, both $\hat{x}_n$ and $\hat{\pi}_n$ are set as zero.

The Pricing Functions

If $\forall c \in \mathcal{C}$, the cost function has the form

$$f_c(y) = q_c y,$$

where $0 < q_c < \underline{p_c}$. Then, in DPoS, the pricing function ϕ_c is set as follows[1]:

$$\phi_c(y) = \begin{cases} \underline{p_c} & y \in [0, w_c) \\ q_c + (\underline{p_c} - q_c) \cdot e^{y/w_c - 1} & y \in [w_c, 1] \\ +\infty & y \in (1, +\infty), \end{cases}$$

where

$$w_c = \left(1 + \ln \frac{\sum_{c' \in \mathcal{C}} (\overline{p_{c'}} - q_{c'})}{\underline{p_c} - q_c}\right)^{-1}$$

is a threshold.

[1] The result is based on the paper *Mechanism design for online resource allocation: A unified approach*.

Outline

Theoretical and Experimental Verification

Theorem: When the cost functions $\{f_c\}_{\forall c \in \mathcal{C}}$ are linear and $\{0 < \underline{\varsigma_c} < \underline{p_c}\}_{\forall c \in \mathcal{C}}$ holds, the competitive ratio α DPoS achieves is the optimal one over all possible online algorithms. Further, its value is

$$\alpha = \max_{\forall c \in \mathcal{C}} \alpha_c = \max_{\forall c \in \mathcal{C}} \frac{1}{w_c}.$$