Network Slicing in Heterogeneous Software-defined RANs

Qiaofeng Qin[1], Nakjung Choi[2], Muntasir Raihan Rahman[2], Marina Thottan[2], and Leandros Tassiulas[1]
[1]Department of Electrical Engineering and Institute for Network Science, Yale University, USA
[2]E2E Network and Service Automation (ENSA) Research Lab, Nokia Bell Labs, USA

***Abstract*—5G technologies promise to revolutionize mobile networks and push them to the limits of resource utilization. Besides better capacity, we also need better resource management via virtualization. End-to-end network slicing not only involves the core but also the Radio Access Network (RAN) which makes this a challenging problem. This is because multiple alternative radio access technologies exist (e. g. ,LTE, WLAN, and WiMAX), and there is no unifying abstraction to compare and compose from diverse technologies. In addition, existing work assumes that all RAN infrastructure exists under a single administrative domain. Software-Defined Radio Access Network (SD-RAN) offers programmability that facilitates a unified abstraction for resource sharing and composition across multiple providers harnessing different technology stacks. In this paper we propose a new architecture for heterogeneous RAN slicing across multiple providers. A central component in our architecture is a service orchestrator that interacts with multiple network providers and service providers to negotiate resource allocations that are jointly optimal. We propose a double auction mechanism that captures the interaction among selfish parties and guarantees convergence to optimal social welfare in finite time. We then demonstrate the feasibility of our proposed system by using open source SD-RAN systems such as EmPOWER (WiFi) and FlexRAN (LTE).**

***Index Terms*—Mechanism Design, Auctions, Network Slicing, SD-RAN**

I. Introduction

5G technologies will revolutionize mobile networks and push them to the limit. Besides the significant improvement in efficiency and capacity, the network has better support to a wide range of services with distinct requirements by virtualization. On the same physical infrastructure, multiple virtual networks are established as slices, and network resources are isolated into each slice to meet the requirement of different services.

An end-to-end virtualization involves slicing not only in the core but also in the radio access networks (RANs). One challenging problem of RAN slicing is the coexistence of heterogeneous radio access technologies (RATs) such as LTE, WLAN and WiMAX, where types of resources are not identical and cannot be allocated under a uniform mechanism. Moreover, it is common for a mobile device (such as a smartphone) to have multiple network interfaces and utilize them simultaneously. Therefore, slicing and radio resource allocation across multiple RANs is required.

This publication was supported partly by the Army Research Office under Agreement Number W911NF-18-10-378 and the National Science Foundation under Grant CNS 1815676. This work was initiated and partially finished during the Bell Labs internship program.

Aiming towards more advanced coordination of heterogeneous RANs, 3GPP has developed standards like LTE-WLAN Aggregation and LTE-WLAN Radio Level Integration. Several slicing architectures across RANs have also been proposed [1] [2]. These solutions usually enforce changes or deploy new components in the network infrastructure. They also assume RANs are owned by the same network provider or they are fully cooperative. These requirements cannot be satisfied in a more general case when multiple network providers have private infrastructures and compete in selling resources to services.

Software-Defined Radio Access Network (SD-RAN) brings new possibilities to this problem. By separating the control plane and data plane, Software-Defined Networking (SDN) provides us with a centralized and programmable management layer over the network. SDN was popularized for datacenters and wired networks, but now they are also being applied to RAN dissagregation. Radio resource allocations at physical devices (e.g., eNodeBs of LTE and Access Points of WLAN) can be achieved in a flexible manner through a central controller.

Building on the flexibility of SD-RAN, we proposed a novel architecture towards heterogeneous RAN slicing. Our architecture includes a slicing orchestrator which coordinates multiple network providers and service providers to reach a joint slicing allocation by negotiations. In our architecture, the SD-RAN controller of each network provider has an associated agent which runs as an SDN application and takes part in the resource allocation mechanism. Compared with existing approaches, our architecture has more flexible and modular support of heterogeneous RANs. As a network application, it can be dynamically deployed on SD-RAN platforms owned by either the same or different operators without any infrastructure changes and is not limited within specific RATs. At the same time, it takes advantage of the functions that already exist in the RAN such as access control, handover and resource abstractions.

Specifically, we make the following contributions:

- We design an architecture of a novel type of orchestrator which realizes network slicing across heterogeneous radio access technologies and SD-RAN platforms, taking diversity of network services, competing network owners and users' multi-connectivity into account.
- We propose theoretical models capturing interactions and competitions among different roles (e.g., network providers and service providers) during slicing configura-

tion. We guarantee optimal social welfare by an iterative double auction algorithm.

- We develop a prototype of our proposed architecture based on state-of-the-art SD-RAN open-source projects and real commercial mobile devices. We evaluate our system by taking measurements from multiple realistic use-cases and evaluating multiple performance metrics.

II. Related Works

SD-RAN. The concept of Software-Defined Networking has been applied to heterogeneous RANs with some success. [3] implements CAPWAP protocol enabling a controller to manage a collection of WLAN access points. Odin [4] proposes light virtual access point (LVAP) abstraction to provide WLANs with SDN services. With similar abstraction, [5], [6] support more applications including network slicing. For LTE networks, FlexRAN [7] is the first SD-RAN platform which separates RAN control and data planes and achieves network slicing. Then, Orion [8] proposes an improved architecture by enabling functional isolation and novel abstractions of slices.

Slicing in Heterogeneous RANs. Among SD-RAN approaches above, [5] aims at having centralized control on not only WiFi access points but also LTE eNodeB. Similarly, architectures of applying control and slicing over heterogeneous RANs are discussed in [1] and [2]. More specifically, [9] and [10] address the control architecture and resource allocation problem across LTE and WLAN. [11] discusses the slicing problem in heterogeneous cellular networks. Most of these works adopt centralized management, ignoring interactions and competitions occur among RANs owned by different parties. In our work we utilize game theoretic modeling and mechanism design to deal with self-interested parties.

Game Theory in Slicing and Resource Allocation. Game theory [12] and mechanism design [13] has been widely applied to slicing and resource allocation of wireless networks [14], [15]. Congestion games and Price of Anarchy (PoA) [16] have been analyzed for network slicing [17]. The authors in [18], [19] design combinatorial auctions for efficient spectrum resource allocation. [20] proposes truthful auctions to enforce cooperation among wireless relay nodes in a network. The authors in [21] propose a share-constrained proportional allocation for network slicing games. Compared with existing works, we take more realistic factors into consideration at the same time, including heterogeneous RATs, services and multi-connectivity of users.

Multi-Connectivity. It is one promising feature of 5G, permitting a single user to make use of multiple access networks simultaneously. Currently there has been different approaches towards multi-connectivity. Multipath TCP (MPTCP) [22] is a solution extending TCP protocol to multiple paths, and has been implemented in different platforms. [23] analyzes MPTCP in mobile devices and propose a proxy-based solutions as an improvement. There are also similar solutions in both commercial software, e. g. , Speedify [24] and open-source projects such as Dispatch-proxy [25]. The authors in [26] adopt an SDN-based method and control multiple interfaces by Open vSwitch [27]. In the design and evaluation

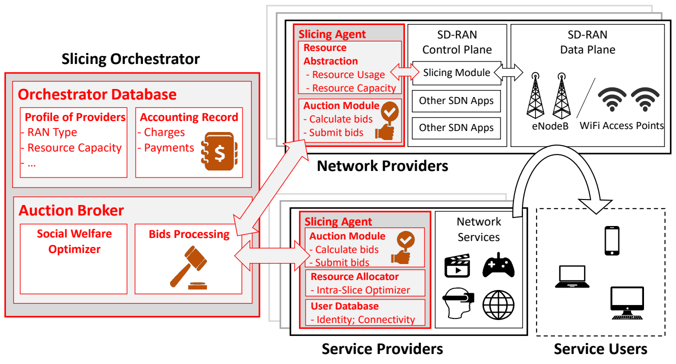


Fig. 1. Architecture of proposed system. The new components we introduce (Slicing Orchestrator and Slicing Agents) are marked in red.

of our system, we take the diversity of multi-connectivity solutions into consideration.

Mobile Data Offloading. Another possible way to make better use of multiple RATs for a cellular network operator is offloading its traffic to third-party owned WiFi access points [28]. Both centralized algorithms [29], [30] and game theory models [31]–[33] are developed towards this approach. Mobile data offloading schemes merely consider interaction between two specific RATs, with the cellular operator as the game leader. Our proposed architecture is not RAT-dependent, and demonstrates the advantages of introducing an orchestrator taking the leader role instead of one of the network providers. This ensures that any single network provider cannot monopolize the market.

III. System Model

A. Overview

In this section, we propose an architecture and system model for achieving network slicing across heterogeneous SD-RANs. In a typical slicing scenario, there will be multiple network providers of same or different radio access technologies (RAT) represented by a set $\mathcal{K} = \{1, 2, ..., K\}$, and multiple service providers represented by $\mathcal{M} = \{1, 2, ..., M\}$. Each service provider owns a slice with a certain amount of isolated resources (e. g., power, bandwidth, speed). Such resource slicing (isolation) tasks are challenging, as they should permit a slice to purchase resources from more than one RAN, and make decisions on the amount of resources to allocate that satisfy the demands of all network and service providers.

To solve this problem, a key component in our design is the *Slicing Orchestrator*. It is a centralized entity establishing connections to all network and service providers. However, it is owned and operated by a third party different from the network and service providers. Although the orchestrator does not have the full access to either control or private information of each RAN, it is capable of managing the competitions among network and service providers by holding *auctions*. The purpose of the orchestrator is to maximize the slicing efficiency, which is represented by *social welfare* maximization, while making profits for itself. Social welfare is typically defined as the sum of utilities of every agent involved in the auction. In order for heterogeneous providers to communicate with the Slicing Orchestrator through a uniform protocol, a *Slicing Agent* is deployed at each network and service provider, which is another significant component in our design. We do not require any modifications or new components in the devices of network and service users.

Figure 1 shows the overall system architecture. To demonstrate the incentive of deploying the orchestrator and the auction mechanism, we first model the problem in aspects of both service and network providers.

B. Service Provider Slicing Agent

Each slice is owned by a service provider. We assume that a Slicing Agent is deployed for requesting resources from network providers. The Slicing Agent aggregates the demands of all service users to estimate the amount of resource required.

User Connectivity Profile. A service provider $m \in \mathcal{M}$ usually has multiple users to serve through its slice. The connectivity of a user $i \in \mathcal{I}_m$ can be denoted by a vector $\beta_{\mathbf{i}} = (\beta_{ki})_{k \in \mathcal{K}}$. β_{ki} is a non-negative number representing factors such as the link quality (e.g., $(0, 1]$ depending on the path loss). The values can be determined according to related indicators (e.g., Channel Quality Indicator (CQI) in LTE, Received Signal Strength Indicator (RSSI) in 802.11) reported by users.

Intra-slice Resource Allocation. The service provider should consider all its users when requesting resources for its slice, which is an optimization problem of intra-slice resource allocation. Suppose a service provider m requests a certain amount of resources from every RAN k denoted by a vector $\mathbf{x_m} = (x_{mk})_{k \in \mathcal{K}}$, then it allocates a portion of them, $\mathbf{z_i} = (z_{ki})_{k \in \mathcal{K}}$ to each user i. Depending on the resource allocated, user i has its utility $u_{mi}(\mathbf{z_i})$ (the form varies based on the type of service). In order to maximize the sum of all user utilities, an optimization problem should be solved by service provider m:

$$U_m(\mathbf{x_m}) = \max_{\mathbf{z_i}} \sum_{i=1}^{I_m} u_{mi}(\mathbf{z_i}) \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{I_m} \frac{z_{ki}}{\beta_{ki}} \leq x_{mk}, \forall k \in \mathcal{K} \tag{2}$$

$$z_{ki} \geq 0, \forall i \in \mathcal{I}_m, k \in \mathcal{K} \tag{3}$$

We can make an assumption that user utility $u_{mi}(.)$ is an increasing and concave function, which holds in most scenarios, e.g., elastic traffic [34], or services that guarantee fairness [35].

Objective during Slicing. The Slicing Agent determines $\mathbf{x_m}$, the amount of resources to request which maximizes the service provider's own interest, denoted by the utility function $U_m(\mathbf{x_m})$ of this service provider m.

C. Network Provider Slicing Agent

Similarly, a Slicing Agent exists as an application of each SD-RAN platform, determining the type and amount of resources allocated to different services.

Resource Abstraction. Although an SD-RAN may have its own abstraction of radio resources (e.g., Resource Blocks in LTE, airtime control in WLAN), it is able to quantify them as the performance level of the same network metric (e.g., downlink throughput), which will be a crucial function of the Slicing Agent. If we consider one specific network metric in this way, the resource offered by a network provider k can be denoted by a vector $\mathbf{y_k} = (y_{km})_{m \in \mathcal{M}}$. Without loss of generality[1], capacity C_k limits the amount of resources that can be offered, i.e., $\sum_{m=1}^{M} y_{km} \leq C_k$.

Objective during Slicing. Similar to the service provider, the Slicing Agent of a network provider aims at maximizing its own profit, i.e., minimizing a cost function $V_k(\mathbf{y_k})$. This cost occurs because of the operation and management overheads of the RAN, as well as the opportunity cost since the network provider cannot use these resources for other purposes. In this paper, we mainly discuss the representative cases where $V_k(\mathbf{y_k})$ function is increasing and convex. [36]

D. Slicing Orchestrator

A service provider requests resources for its slice which maximize its utility, while network providers aim at minimizing their costs. Since these goals are at conflict it is hardly possible to achieve it without negotiations through a third party. This is the job of the Slicing Orchestrator which leads to an agreement of resource requests and offers through double auctions. In other words, a legal slicing scheme requires $y_{km} = x_{mk}$ for any service provider m and network provider k. Beyond that, we consider maximizing the total utility of all providers, which can be regarded as a social welfare optimization problem:

$$\max_{\mathbf{x_m}, \mathbf{y_k}} \sum_{m=1}^{M} U_m(\mathbf{x_m}) - \sum_{k=1}^{K} V_k(\mathbf{y_k}) \tag{4}$$

$$\text{s.t. } y_{km} = x_{mk}, \forall k \in \mathcal{K}, m \in \mathcal{M} \tag{5}$$

$$\sum_{m=1}^{M} y_{km} \leq C_k, \forall k \in \mathcal{K} \tag{6}$$

$$y_{km} \geq 0, x_{mk} \geq 0, \forall k \in \mathcal{K}, m \in \mathcal{M} \tag{7}$$

Although the Slicing Orchestrator acts as a centralized component connecting to Slicing Agents of all providers, it cannot solve this problem directly. First, it is a reasonable assumption that each provider is selfish and cares about their own utility or cost, rather than the social welfare. Second, the orchestrator does not have full access to information private to network and service providers. More specifically, providers do not always have the incentive to reveal their utility and cost functions $U_m(\mathbf{x_m})$ and $V_k(\mathbf{y_k})$. Moreover, the orchestrator should be profiting during the resource allocation in order to maintain itself. In the next section, we will introduce a double auction mechanism to solve this problem, where the Slicing Orchestrator is the broker and each Slicing Agent is a bidder.

[1]Although the capacity can be dependent of k in the cases such as WiFi channel conflicts, the orchestrator introduced later is capable to notify each provider to prevent such conflicts. In addition, it is easy to extend the algorithm described in the next section to other forms of linear constraints.

IV. Methodology

In this section, we introduce the methods to solve the social welfare optimization problem described in the last section. We also analyze the benefit of proposed mechanism theoretically in comparison with other possible slicing architectures.

A. User Utility Optimization

First, we focus on the properties of the service provider utility function $U_m(\mathbf{x_m})$. It is the aggregation of each single service user's utility, where a subproblem of the intra-slice resource allocation exists, as stated in (1).

If $\mathbf{x_m}$ has been determined by the orchestrator, the service provider can directly solve this subproblem by itself. For example, the unique optimal can be efficiently found by applying Karush-Kuhn-Tucker (KKT) conditions [37]. What is more, $U_m(\mathbf{x_m})$ has following important property:

Lemma 1. *$U_m(\mathbf{x_m})$ is an increasing and concave function.*

Proof. For the monotonicity, with an increased x_{mk}, allocating the marginal value to any arbitrary user i improves $u_{mi}(\mathbf{z_i})$ and $\sum_{i=1}^{I_m} u_{mi}(\mathbf{z_i})$. Therefore, the optimal allocation $U_m(\mathbf{x_m})$ is increasing as well.

For the concavity, define $\mathbf{z} = (z_{ki})_{k\in\mathcal{K}, i\in\mathcal{I}_m}$, and

$$f(\mathbf{z}, \mathbf{x_m}) = \begin{cases} \sum_{i=1}^{I_m} u_{mi}(\sum_{k=1}^{K} z_{ki}) & \text{if constraint (2) holds} \\ -\infty & \text{otherwise} \end{cases}$$

$f(\mathbf{z}, \mathbf{x_m})$ is a concave function of both $\mathbf{z}$ and $\mathbf{x_m}$. According to [37], its partial maximization (i.e., $U_m(\mathbf{x_m})$) preserves concavity. □

B. Iterative Double Auction

The result of the subproblem above implies the concavity of the social welfare function, making it possible for us to adopt a double auction mechanism similar to [33] optimizing the resource allocation during slicing.

By applying KKT conditions and introducing Lagrange multipliers, the problem (4) has a unique optimal solution because of the concavity. However, it cannot be acquired without information of $U_m(\mathbf{x_m})$, $V_k(\mathbf{y_k})$. Instead, we consider the following alternative optimization problem:

$$\begin{aligned} L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = & \sum_{m=1}^{M}\sum_{k=1}^{K}(p_{mk}\log x_{mk} - \frac{a_{km}}{2}y_{km}^2) \\ & - \sum_{k=1}^{K}\lambda_k(\sum_{m=1}^{M} y_{km} - C_k) \\ & - \sum_{k=1}^{K}\sum_{m=1}^{M}\mu_{mk}(x_{mk} - y_{km}) \end{aligned} \tag{8}$$

where $\boldsymbol{\lambda} = (\lambda_k \geq 0)_{k\in\mathcal{K}}$ and $\boldsymbol{\mu} = (\mu_{mk} \geq 0)_{k\in\mathcal{K}, m\in\mathcal{M}}$ are Lagrange multipliers. There are undetermined parameters $\mathbf{a_k} = (a_{km} \geq 0)_{k\in\mathcal{K}, m\in\mathcal{M}}$ in this alternative problem, which are the bids that the broker expects each network provider k to submit. Similarly, $\mathbf{p_m} = (p_{km} \geq 0)_{k\in\mathcal{K}}$ are bids from each service provider m. Two sets of rules are required for the auctions. First, we need *allocation rules* to solve this alternative problem. Second, *payment rules* will guide the providers to submit bids determining the parameters of this alternative problem, which should lead the optimal solution coincide with the original problem.

Allocation Rules. Optimal results $\mathbf{x}^*$ and $\mathbf{y}^*$ of this alternative problem can be calculated by KKT conditions of $L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu})$:

$$x_{mk}^* = \frac{p_{mk}}{\mu_{mk}^*}, \qquad y_{km}^* = \frac{\mu_{mk}^* - \lambda_k^*}{a_{km}} \tag{9}$$

$$\mu_{mk}^*(x_{mk}^* - y_{km}^*) = 0, \qquad \lambda_k^*(\sum_{m=1}^{M} y_{km}^* - C_k) = 0 \tag{10}$$

Equation (9) shows the allocation rules of the double auction, revealing how the orchestrator determine the amount of resources allocated to each slice based on bids received.

Payment Rules. By comparing the alternative problem (8) with the original one (4), we notice that they have the same optimal solution $\mathbf{x}^*$ and $\mathbf{y}^*$ only when:

$$p_{mk} = x_{mk}^*\frac{\partial U_m(\mathbf{x_m^*})}{\partial x_{mk}}, \; a_{km} = \frac{1}{y_{km}^*}\frac{\partial V_k(\mathbf{y_k^*})}{\partial y_{km}} \tag{11}$$

The orchestrator applies payment rules to induce bidders submitting the above values. More specifically, the broker charges $g_m(\mathbf{p_m})$ to each service provider m for the resource it bids to request, and pays $h_k(\mathbf{a_k})$ to each network provider k for the resource it bids to offer.

In this case, each service provider m determines its bid that maximizes their payoff:

$$\mathbf{p_m^*} = \arg\max_{\mathbf{p_m}}(U_m(\mathbf{x_m}) - g_m(\mathbf{p_m})) \tag{12}$$

Similarly, each network provider k makes decisions according to:

$$\mathbf{a_k^*} = \arg\max_{\mathbf{a_k}}(-V_k(\mathbf{y_k}) + h_k(\mathbf{a_k})) \tag{13}$$

Payment rules should make these results coincide with Equation (11). It can be calculated by combining the allocation rules (9) with bids expressions (11)(12) and (13). As a result, the payments and charges are proportional to the resources demanded/offered:

$$g_m(\mathbf{p_m}) = \sum_{k=1}^{K} p_{mk}, \qquad h_n(\mathbf{a_k}) = \sum_{m=1}^{M}\frac{(\mu_{mk} - \lambda_k)^2}{a_{km}} \tag{14}$$

Iterative Algorithm. The allocation and payment rules above are parameterized by the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, which can be calculated in a gradient descent manner by running auctions of multiple rounds. The procedure of the iterative double auctions is listed in Algorithm 1.

Defining a Lyapunov function summing the quadratic drifts of $\boldsymbol{\lambda}^{(t)}$ and $\boldsymbol{\mu}^{(t)}$, the convergence can be proved [33] under the assumption that bidders are price-takers, where they passively accept the price raised by the broker, rather than strategically exert impact on it. It is true in perfect competition market, which is reasonable in our architecture because we are considering multiple network and service providers with limited information of each other.

Algorithm 1 Iterative Double Auction

```
1: $t \leftarrow 0$
2: Initialize $\mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \boldsymbol{\lambda}^{(0)}, \boldsymbol{\mu}^{(0)}$
3: IsConverged$\leftarrow$ False
4: while IsConverged is False do
5:     The broker announces $\boldsymbol{\lambda}^t, \boldsymbol{\mu}^t$
6:     Each service provider $m$ submits bids $\mathbf{p_m}^{(t+1)}$ by (12); each network provider $n$ submits bids $\mathbf{a_n}^{(t+1)}$ by (13)
7:     The broker calculates $\mathbf{x}^{(t+1)}$ and $\mathbf{y}^{(t+1)}$ by (9)
8:     The broker calculates $\boldsymbol{\lambda}^{(t+1)}$ and $\boldsymbol{\mu}^{(t+1)}$ by:
       $\mu_{mk}^{(t+1)} = (\mu^{(t)} + s^{(t)} \cdot (x_{mk}^{(t)} - y_{km}^{(t)}))^+$
       $\lambda_k^{(t+1)} = (\lambda_k^{(t)} + s^{(t)} \cdot (\sum_{m=1}^M y_{km}^{(t)} - C_k))^+$
       $\forall k \in \mathcal{K}, m \in \mathcal{M}$, and $s^{(t)} > 0$ is the step size of gradient descent.
9:     if $|\frac{p_{mk}^{(t+1)} - p_{mk}^{(t)}}{p_{mk}^{(t)}}| < \epsilon_1$ and $|\frac{a_{km}^{(t+1)} - a_{km}^{(t)}}{a_{km}^{(t)}}| < \epsilon_2$, $\forall k \in \mathcal{K}, m \in \mathcal{M}$ then
10:        IsConverged $\leftarrow$ True
11:    end if
12:    $t \leftarrow t + 1$
13: end while
14: Output $\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\mu}^{(t)}$
```

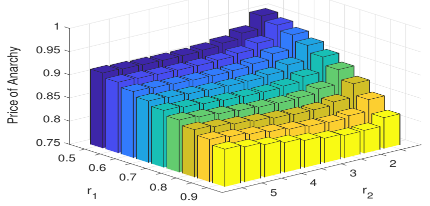


Fig. 2. PoA with different utilities and costs when disabling the orchestrator.

According to (12)(13), the algorithm is efficient and individually rational, i.e., optimal social welfare is reached when every bidder maximizes their own payoffs. This conclusion can easily be extended to the case where a service provider owns multiple slices or a network provider owns multiple sets of RAN infrastructures. The provider can simply make decisions for each of its slice/infrastructure independently.

The algorithm is scalable in aspects of both computing (concave minimization with linear constraints) and synchronization overheads ($\mathcal{O}(M \cdot K)$ messages in each round). In addition, it is straightforward to demonstrate that the profit $\sum_{m=1}^M g_m(\mathbf{p_m^*}) - \sum_{k=1}^K h_k(\mathbf{a_k^*})$ is always non-negative. Therefore, the orchestrator faces no problem of maintaining itself (it never runs a loss) and has the incentive to hold auctions.

C. Social Welfare Improvement

The optimal social welfare achieved by introducing the Slicing Orchestrator is nontrivial. In particular, we demonstrate that while our method can guarantee optimality with Price of Anarchy (PoA) = 1, alternative distributed architectures without centralized control (where providers directly negotiate with each other as described below) can lead to sub-optimal performance with PoA < 1. Thus the central orchestrator of our architecture is an essential component for RAN slicing.

The Stackelberg game is a typical model to depict such distributed architectures and widely discussed in related literature [38] [39], in which a leader and followers take actions sequentially. The interaction of each service provider m and network providers in this alternative architecture can therefore be captured by a two-stage Stackelberg game:

- Stage 1: The service provider announces P_m, the price it is willing to pay for every unit resource offered.
- Stage 2: Each network provider k submits $x_{mk} = y_{km}$.

To analyze the price of anarchy (PoA) for the Stackelberg game model, we need to make a few additional assumptions. First, we assume network providers are able to interact with every service provider independently, by assuming $V_k(\mathbf{y}_k) = \sum_{m=1}^M V_{km}(y_{km})$. Secondly, the capacity of resources is no longer a constraint. We also assume all information are public. These assumptions actually weaken the practicality of such models. A strength of our proposed design is that the difficulties due to these assumptions are avoided. Moreover, we demonstrate that the Stackelberg game has inferior social welfare even if all these extra assumptions are satisfied.

The game has an equilibrium. At Stage 2, given the price P_m, a network provider responds maximizing its payoff:

$$y_{mk}^*(P_m) = \arg\max_{y_{km}} (P_m \cdot y_{km} - V_{km}(y_{km})) \tag{15}$$

Anticipating the response, the service provider will determine the price in Stage 1 as:

$$P_m^* = \arg\max_{P_m} (U_m(\mathbf{y}_m^*(P_m)) - P_m \cdot \sum_{k=1}^K y_{mk}^*(P_m)) \tag{16}$$

Correspondingly, the social welfare under the equilibrium is $SW_{equil} = \sum_{m=1}^M U_m(\mathbf{y}_m^*(P_m^*)) - \sum_{k=1}^K V_k(\mathbf{y}_k^*(\mathbf{P}_m^*))$.

We introduce Price of Anarchy (PoA), the ratio of social welfare between the worst equilibrium and the centralized optimal solution, as a metric to demonstrate the benefits gained by setting up an orchestrator. For instance, we consider following power functions as utility and cost, i.e., $U_m(\mathbf{x}_m) = A \cdot (\sum_{k=1}^K x_{mk})^{r_1}$ and $V_k(y_{km}) = B_k \cdot \sum_{m=1}^M y_{km}^{r_2}$, $\forall k \in \mathcal{K}$, $r_1 \in (0,1), r_2 \in (1, \infty)$. Then the equilibrium social welfare SW_{equil} can be calculated following (15) and (16). It is also trivial to acquire the unique optimal solution SW_{opt} by making $x_{mk} = y_{km}$ and taking derivatives of $\sum_{m=1}^M U_m(\mathbf{x}_m) - \sum_{k=1}^K V_k(\mathbf{y}_k)$. Finally the PoA has following expression:

$$PoA = \frac{SW_{equil}}{SW_{opt}} = \frac{r_2^{\frac{2r_1}{r_1 - r_2}} - r_2^{\frac{2r_2}{r_1 - r_2}} \cdot r_1}{r_2^{\frac{r_1}{r_1 - r_2}} - r_2^{\frac{r_2}{r_1 - r_2}} \cdot r_1} \tag{17}$$

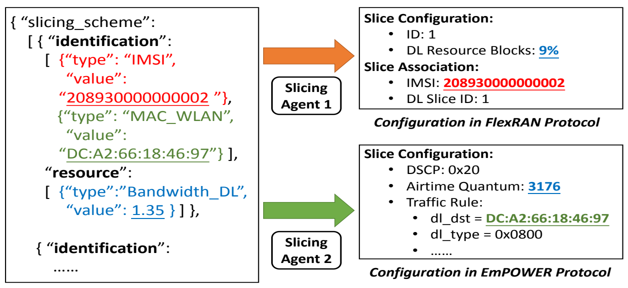
The PoA is impacted by the extent of concavity/convexity of utility/cost functions. In Figure 2 we calculate PoA in different combinations of r_1 and r_2. The results are lower than 0.8 in worst cases, indicating that our proposed architecture is able to improve the social welfare by more than 25% in specific scenarios. Later in the evaluation section, we will also demonstrate this improvement in realistic settings.

V. Implementation

We develop a prototype of proposed Slicing Agents and Slicing Orchestrator and test them over several SD-RAN controllers. In this section, we introduce some details of our implementation.

We define two protocols in the system. First, RESTful APIs are exposed to operators for the database update and lookup. The orchestrator is open for network and service providers to register in the system, update their information or quit the system. It also keeps an account recording the history charges and compensations caused by auctions to each provider. Similarly, the agent has APIs for a service provider to update the profiles of its users, including the list of RANs a user is connecting to, the signal quality it receives and its identification in each RAN. The double auction is also initiated at the side of service provider, by specifying the type of resource to request, the agent will send an auction message to the Slicing Orchestrator.

Once the auction message is accepted by the orchestrator, it will broadcast to all agents to start an auction through the second protocol. The auction proceeds automatically by the communications between the orchestrator and agents, following the steps described in Algorithm 1. The auction protocol defines several types of messages, representing actions during a double auction such as bidding and parameter updating. When the orchestrator ensures the convergence of the algorithm, it broadcasts a message to end the auction, and sends a summary to each bidder, containing the final slicing scheme and a bill of charges/compensations. The auction module of our prototype is implemented in Python 3, with bidding decisions calculated by Scipy [40] optimizer.



```
{ "slicing_scheme":
   [ { "identification":
      [  {"type": "IMSI",
          "value":
         "208930000000002 "},
         {"type": "MAC_WLAN",
          "value":
         "DC:A2:66:18:46:97"} ],
     "resource":
      [  {"type":"Bandwidth_DL",
          "value": 1.35 } ] },

    { "identification":
       ......
```

Slice Configuration:
- ID: 1
- DL Resource Blocks: 9%

Slice Association:
- IMSI: 208930000000002
- DL Slice ID: 1

Configuration in FlexRAN Protocol

Slice Configuration:
- DSCP: 0x20
- Airtime Quantum: 3176
- Traffic Rule:
  - dl_dst = DC:A2:66:18:46:97
  - dl_type = 0x0800
  - ......

Configuration in EmPOWER Protocol

Fig. 3. Abstract of a slicing scheme as the result of an auction, from which agents of network providers extract information and convert it into a readable format for heterogeneous SD-RANs.

Receiving the slicing scheme, the agent of network provider calls its SD-RAN controller to actually execute slicing. The Slicing Agent works as a bridge enabling the interaction between Slicing Orchestrator and SD-RAN controller. It establishes southbound communications with the SD-RAN following the controller's protocols and interfaces (which may be heterogeneous). As an instance, we implement the downlink bandwidth auction with FlexRAN LTE controller and EmPOWER WLAN controller. FlexRAN identifies a user by its International Mobile Subscriber Identity (IMSI), and realizes bandwidth slicing by allocating specific number of Resource Blocks (RBs), the smallest resource unit of an LTE frame.

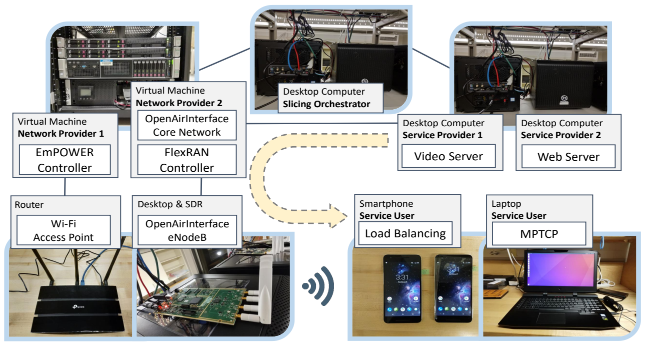


Fig. 4. Testbed setup and experimentation scenario of two services (video streaming and web browsing) and two RANs (WLAN and LTE).

On the other hand, EmPOWER marks flows classified by OpenFlow [41] rules with a Differentiated Services Code Point (DSCP) header, and applies the Airtime Deficit Round Robin (ADRR) packet scheduling policy [42] for downlink bandwidth slicing. Correspondingly, the Slicing Agent has two major tasks. First, for each user it picks the proper identification (e.g., IMSI for FlexRAN and OpenFlow fields for EmPOWER) from the a multiple ones provided by the service provider. Then, it will translate the amount of bandwidth requested into the unit which the SD-RAN controller adopts, e.g., number of Resource Blocks for FlexRAN, and airtime portion for EmPOWER. Figure 3 shows the details of above example about the resource and user identity abstraction. Although protocol-dependent, development of such a module is not a bottleneck when a new SD-RAN joins the coordination of Slicing Orchestrator. In our prototype, lines of this module's codes account for less than 10% in total. And remaining codes are identical for all providers. This enables the modular and fast deployment of proposed architecture in heterogeneous RANs with no infrastructure modification and minor code development required.

VI. Evaluation

In this section, we evaluate our proposed architecture and algorithm with both our implementation in real devices and numerical simulations in large-scale network topologies.

A. Testbed Setup

To quantitatively evaluate the performance of realistic scenarios, we build a testbed containing heterogeneous SD-RANs, multiple network services and different types of user devices, as shown in Figure 4.

We set up two network providers, one of LTE and another of WLAN. In the LTE network, a desktop computer (3.6 GHz, 16 GB of RAM) with USRP B210 deploying OpenAirInterface [43] eNodeB works as the data plane on LTE band 7. We also deploy virtual machines in a server (HP ProLiant DL360) running components of LTE core network (HSS, MME and SPGW) and the FlexRAN control plane. The WLAN network has similar setting, while deploying EmPOWER control plane and using a router (TP-Link AC1750) as data plane on 802.11g, channel 11 instead. Control planes of both RANs run our Network Provider Agent.

We also set up two service providers (one web server and one video streaming server) in another desktop computer, both deploying our Service Provider Agent. The Slicing Orchestrator exists in third desktop computer. 1Gb/s Ethernet links are set between RAN data planes and RAN control planes, as well as the agents and orchestrator.

Two types of user devices, one HP Omen laptop and two Nexus 6P Android smartphones are deployed for experiments. The laptop connects to LTE network with a Huawei E3372 LTE USB modem. Besides, we consider multiple approaches enabling multi-connectivity and deploy different solutions in user devices.

B. Experimentation

Video Streaming Service. First, we consider a scenario where a single service provider requests downlink bandwidths from both LTE and WLAN providers, in order to support HTTP video streaming to a laptop through MPTCP v0.93 [22]. We assume that network providers have cost functions $V_k(\mathbf{y_k}) = w_k^n \cdot \sum_{m=1}^{M} y_{km}^2$. Here we have $k = 1$ for LTE and $k = 2$ for WLAN. According to the actual performance of infrastructure, we set $C_1 = 15Mbps$ and $C_2 = 9Mbps$. On the user side, we choose the quality of received video as its utility. Although it would not be easy to deduct its exact relationship with downlink bandwidth, the network provider can estimate it using an elastic utility function $u_{mi}(\mathbf{z_i}) = w_m^s \cdot \sum_{k=1}^{K}(1 - e^{-\alpha \cdot z_{ki}})$, $m = 1, i = 1$, which is general to cover various services [34]. Value of α can be estimated depending on the video bitrate. We use a 1080P video (around 7000 kbps) for experimentation, and assume $\alpha = 1.6$ correspondingly.

We fix $w_1^n = 0.2$, $w_2^n = 0.1$ and run the system with different w_1^s values, which represent the willingness of the service provider to purchase resources for its slice. Figure 5(a) shows the results of proposed double auction algorithm, in which the service provider requests a share of bandwidth from both RANs. With a larger w_1^s, the service provider acquires more resources, indicating the capability of proposed system to balance the offer and request with different utilities/costs of providers. The result is concave in w_1^s, which suits the video streaming service because redundant bandwidth beyond the video bitrate adds little value. The orchestrator receives payments from the service provider and make compensations to network providers. In this scenario, these two amounts are balanced. And it is intuitive that the service provider pays more for larger requests.

We also measure the actual performance of video streaming. After the whole video is streamed, we quantify the received video quality by measuring its peak signal-to-noise ratio (PSNR). A larger PSNR value indicates a smaller quality loss during streaming. Figure 5(b) shows how the PSNR values increase with larger bandwidth allocated.

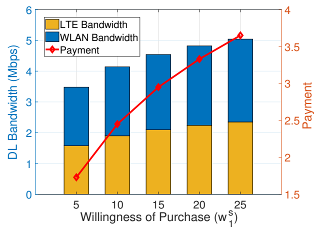
Another crucial performance metric of proposed system is the time spent on finishing the slice configuration. Figure 5(c) shows the procedure of the iterative double auction. With gradient descent step size 0.1, the offers and requests quickly converge in 11 bidding rounds. We also measure the actual time spent on finishing an auction as Figure 5(d), in which we hold auctions for 100 times and plot the cumulative distribution function (CDF). With all participants placed in the same room with wired connections, it always takes less than 0.1 second. Then we add a simulated delay on all outgoing traffic from the Slicing Orchestrator. The auctions can still be finished quickly within 1 and 2 seconds, when the delay is set to $10ms$ and $20ms$.

Furthermore, to verify above conclusions from another aspect, we monitor the traffic of video streaming through both RANs in real time. As depicted in Figure 6, initially the throughput from each RAN is consistent with the slicing scheme in Figure 5(a). Then, we assume that the service provider changes its willingness w_1^s from 5 to 25 and therefore starts a new auction at the 15^{th} second. As a result, the throughput starts to increase within 1 second, and becomes stable again within 5 seconds, indicating the whole procedure of slicing has been finished. In the figure, the WLAN throughput shows fluctuations, because we set the temporal interval of the curve as 0.1 second to better indicate the system's dynamic response. The queueing-based slicing mechanism of EmPOWER cannot achieve the same level of fine-grained control as the Resource Block allocation of LTE. However, it is able to follow the auction result correctly on a larger time scale, e.g., when measuring the average bandwidth of every 1 second. Therefore, we assert that the auction and slicing mechanisms work smoothly as designed and is flexible enough to adapt dynamic changes of user demands.

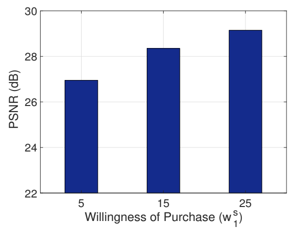
Web Browsing Service. We then consider another scenario of two Android smartphones surfing the Internet to investigate different performance metrics. Instead of MPTCP, we consider another multi-connectivity case, the load balancing assignment of flows to different network interfaces. With each smartphone, we send the same amount of HTTP requests to download a large HTML page (around 1.2 MB) through both LTE and WLAN connections. And we measure the average page load time as the performance metric. Corresponding to this metric, the service provider may choose a different utility function in the form of minimum potential delay fairness, $u_{mi}(\mathbf{z_i}) = w_m^s \cdot \sum_{k=1}^{K}(-1/z_{ki})$, $m = 2, i = 1, 2$. We assume the setting of network providers are the same as in the last scenario.

Figure 7(a) shows that the auction algorithm also succeeds in balancing the requests and offers with this new utility function. We repeat the download for 100 times using both network interfaces and take the average value of page load time for each slicing scheme. As Figure 7(b) indicates, the delay decreases if the service provider requests more bandwidth for its users.

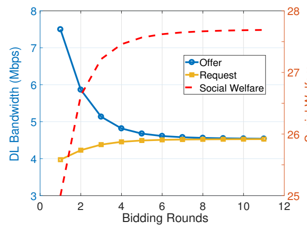
We also investigate how the proposed system is able to tackle with device mobility. Keeping $w_2^s = 5$, we change the location of the second smartphone so that its LTE signal strength falls to around -115 dB from -95 dB. (The WLAN signal strength is less impacted.) In this case, the user can report its signal strength to the service provider, reflecting in an updated β_{12} parameter in the formulation. Then the service provider can call a new auction to adjust its bids. Figure 7(c) compares the auction results before (Case 1) and after (Case 2) the movement, considering which the service



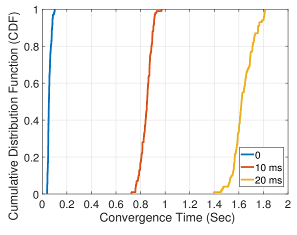
(a) Converged resource allocation.



(b) Performance of video streaming.



(c) Convergence of an auction.



(d) Time to finish auctions.

Fig. 5. The (a) resource allocation and payment schemes determined by double auction with different weights of service provider utilities. (b) Actual performance of video streaming by measuring PSNR. (c) Number of auction rounds and (d) actual time required to finish the algorithm.

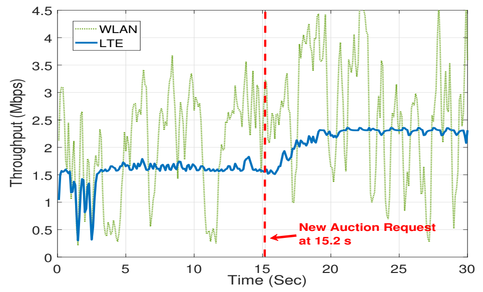


Fig. 6. Real-time MPTCP throughput monitoring of video streaming. The service provider increases w_1^s from 5 to 25 and starts a new auction during the transmission.

provider requests more LTE bandwidth, and allocates a larger portion of it to its second user. In Figure 7(d), we measure the performance of this new allocation, in comparison of an average allocation, where two smartphones still acquire the same portion of resources. It indicates that there is an improvement on the load time of the second user. The negative impact of worse signal is not totally eliminated, because the service provider needs to pay for the extra resource allocated. However, it is able to find a balance and achieve the optimal social welfare.

Multiple Slices. The proposed design also handles competitions among service providers. In this scenario, we run the two services above simultaneously. Figure 8(a) shows the performance of two services under different combinations of w_1^s and w_2^s. The service provider with higher purchase willingness is able to achieve better performance over the other one. We compare these results with the case in which slicing is not applied, e.g., the WLAN applies no queueing policies, and the LTE eNodeB allocates the same number of resource blocks to every user. We examine cases in which the eNodeB offers 20%, 60% and 100% of resource blocks. These plans lead to different costs as well, as depicted in Figure 8(b). In each case, either great performance degradation or significant additional cost incurs, and the resources allocated to two services are also severely imbalanced. All these factors lead to a worse (and even negative) social welfare than our optimal result.

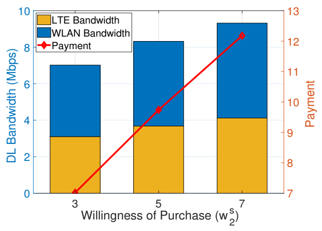
The time required for convergence does not dramatically increase with more bidders. Figure 8(c) shows the procedure of an auction with $w_1^s = 15$ and $w_2^s = 5$. Here the social welfare appears to decrease with time, because the constraints are not yet satisfied. The final result is still optimal. The performance of larger scale auctions will be further analyzed in the next subsection.

In Figure 8(d) we measure the CPU usage (of two cores) and memory consumption of SD-RAN components. The first column shows a baseline, the consumption of the original SD-RAN controller (EmPOWER and SDN controller for WLAN, FlexRAN and OpenAirInterface EPC for LTE) without the deployment of our agents. In the second column, the Slicing Agents are deployed. When not processing auctions, no additional CPU resource is required. Only a small extra portion of memory is occupied. In the third column we keep initiating auctions with an interval of 1 second, therefore the Slicing Agents are busy bidding and implementing the slicing schemes, leading to larger while still affordable CPU usage. From the results shown in the table, the Slicing Agent is lightweight and does not exert heavy extra burden on the SD-RAN controller.

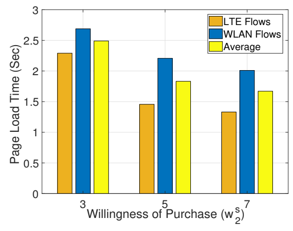
C. Numerical Results

Scalability. Having verified our design and implementation in the small-scale testbed, we now run simulations of larger network topology to guarantee that the performance of proposed design will not degrade when the network scaling up. More specifically, we consider multiple RANs and services in a $100m \times 100m$ area. Among K RANs, the first two are LTE while remaining are WLAN providers each with an Access Point. Each of M different service providers has I users with dual-connectivity of LTE and WLAN. Entities above are distributed uniformly in this area. A user's LTE provider is randomly assigned with uniform probability, and it connects to its nearest WLAN access point. Each LTE network covers the whole area with $\beta_{ki} = 1$, while WLAN's β_{ki} is proportional to the spectral efficiency of Shannon formula, following Rayleigh fading depending on the distance between the user and the Access Point. All other parameters (e.g., w_k^n, w_m^s, α) and utility/cost functions are identical to the testbed, except that we multiply each of them with a random factor uniformly distributed in $[0.9, 1.1]$, and enlarge the capacity by I times consistent with the increasing amount of users.

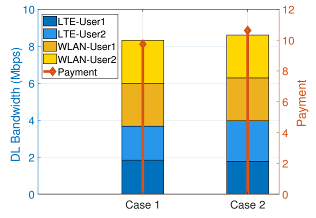
We investigate the impact of network scale by observing the speed of convergence with different numbers of network and service providers. Figure 9(a) depicts the number of bidding rounds until convergence with up to 8 network providers and 8 services (each has 10 users). It does not grow dramatically with more providers participating the auction. Besides, the convergence speed can be adjusted by setting proper step size of gradient descent at the orchestrator.



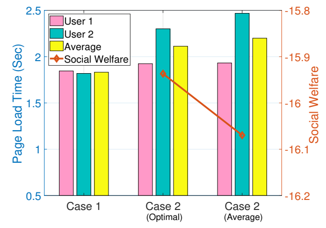
(a) Converged resource allocation.



(b) Performance of web browsing.

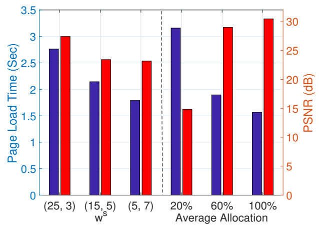


(c) Adaptive adjustment of requests.

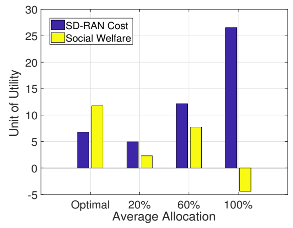


(d) Performance comparison.

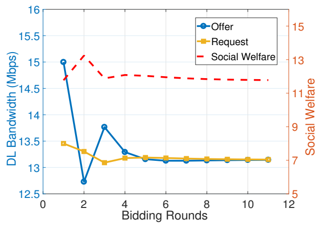
Fig. 7. The (a) resource allocation, payment schemes and (b) performance of web browsing service. (c)(d) shows how the service provider adjusts its bids depending on the signal strength of its users.



(a) Competition between services.



(b) RAN costs and social welfare.

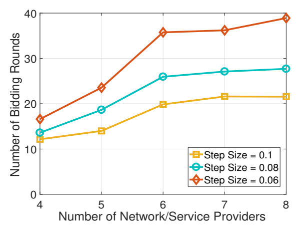


(c) Convergence of an auction.

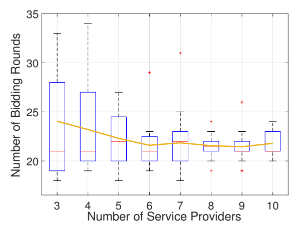
	CPU (%)		Memory (MB)	
	WLAN	LTE	WLAN	LTE
w/o Agent	27.7	21.9	106.5	194.2
Idle	27.7	21.9	159.7	234.7
Busy	38.2	25.5	159.7	234.7

(d) CPU and memory costs.

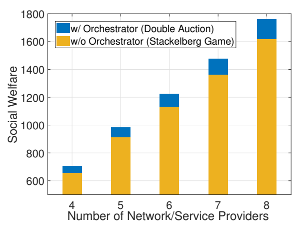
Fig. 8. (a) Performance of two different services under their competition. (b) Cost and social welfare comparisons between the optimal and average allocations. (c) Number of bidding rounds until convergence. (d) CPU and memory consumption of SD-RANs.



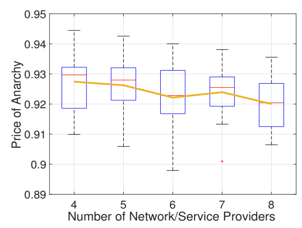
(a) Convergence speed of different scales.



(b) Range and average of convergence speed.



(c) Social welfare of different architectures.



(d) PoA without an orchestrator.

Fig. 9. (a) The number of bidding rounds required for convergence when the network scales up. (b) Box plots and average values of auctions among 8 network providers and different number of service providers. (c) Social welfare of proposed architecture where an orchestrator holds Double Auctions and another architecture where providers compete as a Stackelberg game. (d) Price of Anarchy in slicing games without an orchestrator.

Noticing that the marginal increase of bidding rounds becomes even slighter with more providers, we investigate it further and have Figure 9(b) changing the number of service providers (and users) while keeping 8 network providers. Both variance and average values are larger when there are fewer services, because users sparse in the area are more likely to result in unbalanced resource requests to each RAN, which need more iterations to converge. Due to features shown above, our approach has good scalability.

PoA. In Figure 9(c) and 9(d), we investigate the social welfare improvement compared with the Stackelberg game model without an orchestrator, as stated in the previous section. we plot the box plot and the average values of PoA in different topology, indicating an improvement of social welfare between 7% and 10% in most cases.

VII. Conclusion

In this paper we have proposed a new architecture for resource slicing across multiple selfish network providers using diverse technologies. Our proposed double auction mechanism guarantees convergence to optimal social welfare in finite time. Our central Slicing Orchestrator enables a unified resource abstraction to compare and compose resources exposed by diverse RAN technologies. We have demonstrated the feasibility of our architecture by deploying our orchestrator along with open source RAN slicing systems such as EmPOWER and FlexRAN. Our future plans include deploying our prototype in a larger testbed with many network and service providers to have more scalable and comprehensive evaluations. In addition, we will incorporate other RATs and different types of resources into our architecture.

References

[1] T. Chen, H. Zhang, X. Chen, and O. Tirkkonen, "Softmobile: Control evolution for future heterogeneous mobile networks," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 70–78, 2014.

[2] A. Boubendir, F. Guillemin, C. Le Toquin, M.-L. Alberi-Morel, F. Faucheux, S. Kerboeuf, J.-L. Lafragette, and B. Orlandi, "Federation of cross-domain edge resources: a brokering architecture for network slicing," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, 2018, pp. 415–423.

[3] M. Bernaschi, F. Cacace, G. Iannello, M. Vellucci, and L. Vollero, "Opencapwap: An open source capwap implementation for the management and configuration of wifi hot-spots," *Computer Networks*, vol. 53, no. 2, pp. 217–230, 2009.

[4] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise wlans with odin," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 115–120.

[5] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming abstractions for software-defined wireless networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 146–162, 2015.

[6] E. Coronado, R. Riggio, J. Villa1ón, and A. Garrido, "Lasagna: Programming abstractions for end-to-end slicing in software-defined wlans," in *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2018, pp. 14–15.

[7] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "Flexran: A flexible and programmable platform for software-defined radio access networks," in *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. ACM, 2016, pp. 427–441.

[8] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proceedings of the 23rd annual international conference on mobile computing and networking*. ACM, 2017, pp. 127–140.

[9] S. Kang and W. Yoon, "Sdn-based resource allocation for heterogeneous lte and wlan multi-radio networks," *The Journal of Supercomputing*, vol. 72, no. 4, pp. 1342–1362, 2016.

[10] A. S. D. Alfoudi, M. Dighriri, G. M. Lee, R. Pereira, and F. P. Tso, "Traffic management in lte-wifi slicing networks," in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*. IEEE, 2017, pp. 268–273.

[11] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic network slicing for multitenant heterogeneous cloud radio access networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2146–2161, 2018.

[12] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[13] N. Nisan and A. Ronen, "Algorithmic mechanism design (extended abstract)," in *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, ser. STOC '99, 1999, pp. 129–140.

[14] D. E. Charilas and A. D. Panagopoulos, "A survey on game theory applications in wireless networks," *Computer Networks*, vol. 54, no. 18, pp. 3421 – 3430, 2010.

[15] U. Habiba and E. Hossain, "Auction mechanisms for virtualization in 5g cellular networks: Basics, trends, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2264–2293, 2018.

[16] T. Roughgarden, *Selfish Routing and the Price of Anarchy*. The MIT Press, 2005.

[17] S. D'Oro, F. Restuccia, T. Melodia, and S. Palazzo, "Low-complexity distributed radio access network slicing: Algorithms and experimental results," *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 6, pp. 2815–2828, 2018.

[18] L. Zhong, Q. Huang, F. Wu, and G. Chen, "TRADE: A truthful online combinatorial auction for spectrum allocation in cognitive radio networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 9, pp. 1320–1330, 2015. [Online]. Available: https://doi.org/10.1002/wcm.2411

[19] X. Zhou, S. Gandhi, S. Suri, and H. Zheng, "ebay in the sky: Strategy-proof wireless spectrum auctions," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, ser. MobiCom '08, 2008, pp. 2–13.

[20] D. Yang, X. Fang, and G. Xue, "Truthful auction for cooperative communications," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '11, 2011, pp. 9:1–9:10.

[21] P. C. Garces, A. Banchs, G. de Veciana, and X. Costa-Pérez, "Network slicing games: Enabling customization in multi-tenant mobile networks," *CoRR*, vol. abs/1612.08446, 2016.

[22] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp." in *NSDI*, vol. 11, 2011, pp. 8–8.

[23] A. Nikravesh, Y. Guo, F. Qian, Z. M. Mao, and S. Sen, "An in-depth understanding of multipath tcp on mobile devices: Measurement and system design," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 189–201.

[24] Connectify, "Speedify," http://speedify.com/, 2014.

[25] A. Kirszenberg, "Dispatch-proxy," https://github.com/alexkirsz/dispatch-proxy, 2014.

[26] K.-K. Yap, T.-Y. Huang, M. Kobayashi, Y. Yiakoumis, N. McKeown, S. Katti, and G. Parulkar, "Making use of all the networks around us: a case study in android," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 455–460, 2012.

[27] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar *et al.*, "The design and implementation of open vswitch," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, 2015, pp. 117–130.

[28] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" in *Proceedings of the 6th International COnference*. ACM, 2010, p. 26.

[29] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng, "Multiple mobile data offloading through delay tolerant networks," in *Proceedings of the 6th ACM workshop on Challenged networks*. ACM, 2011, pp. 43–48.

[30] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Deploying carrier-grade wifi: Offload traffic, not money," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2016, pp. 131–140.

[31] J. Lee, Y. Yi, S. Chong, and Y. Jin, "Economics of wifi offloading: Trading delay for cellular capacity," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1540–1554, 2014.

[32] S. Paris, F. Martisnon, I. Filippini, and L. Clien, "A bandwidth trading marketplace for mobile data offloading," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 430–434.

[33] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 5, pp. 1634–1647, 2015.

[34] V. Rakocevic, J. Griffiths, and G. Cope, "Performance analysis of bandwidth allocation schemes in multiservice ip networks using utility functions," in *Teletraffic Science and Engineering*. Elsevier, 2001, vol. 4, pp. 233–243.

[35] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[36] C. Courcoubetis and R. Weber, *Pricing communication networks: economics, technology and modelling*. John Wiley & Sons, 2003.

[37] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[38] V. Gajić, J. Huang, and B. Rimoldi, "Competition of wireless providers for atomic users," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 512–525, 2014.

[39] H. Kim and M. Thottan, "A two-stage market model for microgrid power transactions via aggregators," *Bell Labs Technical Journal*, vol. 16, no. 3, pp. 101–107, 2011.

[40] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

[41] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[42] R. Riggio, D. Miorandi, and I. Chlamtac, "Airtime deficit round robin (adrr) packet scheduling algorithm," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2008, pp. 647–652.

[43] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "Openairinterface: A flexible platform for 5g research," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 33–38, 2014.