This article has been accepted for publication in a future issue of IEEE Transactions on Networking. Citation information: DOI 10.1109/TNET.2022.3197669

Online Learning for Rate-Adaptive Task Offloading under Latency Constraints in Serverless Edge Computing

Feridun Tütüncüoğlu, Slađana Jošilo and György Dán
Division of Network and Systems Engineering,
School of Electrical Engineering and Computer Science
KTH, Royal Institute of Technology, Stockholm, Sweden E-mail: {feridun, josilo, gyuri}@kth.se

***Abstract*—We consider the interplay between latency constrained applications and function-level resource management in a serverless edge computing environment. We develop a game theoretic model of the interaction between rate adaptive applications and a load balancing operator under a function-oriented pay-as-you-go pricing model. We show that under perfect information, the strategic interaction between the applications can be formulated as a generalized Nash equilibrium problem, and use variational inequality theory to prove that the game admits an equilibrium. For the case of imperfect information, we propose an online learning algorithm for applications to maximize their utility through rate adaptation and resource reservation. We show that the proposed algorithm can converge to equilibria and achieves zero regret asymptotically, and our simulation results show that the algorithm achieves good system performance at equilibrium, ensures fast convergence, and enables applications to meet their latency constraints.**

***Index Terms*—generalized Nash equilibrium problem, online learning, serverless edge computing, resource allocation.**

I. Introduction

Edge computing brings computing resources close to the network edge, and is emerging as a key enabler for latency sensitive and bandwidth intensive applications. Examples of applications that could benefit from edge computing include augmented reality, computer vision-enabled automation and surveillance [1]–[3].

Nonetheless, large scale deployment of applications in edge computing environments will require a deployment and management interface that provides simple abstractions for the management and maintenance of physical resources, consisting of a small set of parameters that are configurable in real-time. These parameters should be such that they allow to control application performance, e.g., in terms of the trade-off between data rate and latency, while providing information about pricing and billing. At the same time, the abstraction should allow edge infrastructure operators to efficiently manage the available physical resources, subject to energy and reliability constraints.

A promising lightweight abstraction that could potentially suit a variety of edge applications is function as a service (FaaS). In the case of FaaS, applications are explicitly composed of the subsequent parallel or sequential invocation of subtasks, referred to as functions [4]. Functions are managed, i.e., instantiated, executed and shut down, by the infrastructure, relieving the programmer from the burden of system configuration. Stateless FaaS has already found adoption in cloud computing, referred to as serverless computing, as it provides autoscaling and follows the pay-as-you-go pricing model [5]. Recently proposed solutions for stateful FaaS could extend this offering with low-latency mutable state and communication in the near future [4], [6].

Nonetheless, compared to a cloud computing environment, resource management for FaaS in an edge computing environment faces a number of novel challenges [7]. First, it has to cater for heterogeneous hardware platforms, and has to consider the orchestration of communication and computing resources. Second, it should cater for the latency requirements of applications that involve the execution of multiple functions, and at the same time may be able to adjust their data rate so as to maximize their utility. Third, it has to deal with the strategic interaction between multiple applications for constrained resources. The outcome of the resulting interaction between infrastructure resource management and application behavior is, however, not well understood.

Motivated by the above challenges, in this paper we consider the interaction between rate control and infrastructure resource management for latency sensitive tasks in a serverless edge computing system, and make the following main contributions:

- We propose a queuing network model of task graph

The work was partly funded by the Vinnova Center for Trustworthy Edge Computing Systems and Applications (TECoSA) and the Swedish Research Council through project 2020-03860. The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Linköping University partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

execution and use it for formulating a game theoretic model of the interaction between self interested wireless devices that can reserve communication and computing resources, and a FaaS edge operator that allocates the resources.
- We show pseudoconvexity of the task sojourn time with respect to the arrival intensity in a G/G/1 queue and in a G/G/1 fork-join network, a result that may be of independent interest.
- We show that under perfect information the strategic interaction between *Wireless Devices* (WDs) can be formulated as a generalized Nash equilibrium problem, and we show the existence of Nash equilibria by using variational inequality theory.
- For the case of imperfect information, we propose an online algorithm called *Online Adaptive Rate Reservation and Control* (OARC) for learning equilibria in a distributed manner. We show that OARC converges to equilibria and achieves zero regret asymptotically.
- Our numerical results show that OARC outperforms the state of the art in *Online Convex Optimization* (OCO) for a variety of task graphs.

The rest of the paper is organized as follows. We present the system model and problem formulation in Section II, and prove pseudoconvexity and monotonicity of the sojourn time in fork-join networks in Section III. We consider equilibria under perfect information in Section IV, and learning equilibria under imperfect information in Section V. Section VI presents numerical results. Section VII discusses related work and Section VIII concludes the paper.

II. System Model and Problem Formulation

We consider an edge computing system that consists of a set $\mathcal{N} = \{1, 2, \ldots, N\}$ of wireless devices (WDs), a set $\mathcal{A} = \{1, 2, \ldots, A\}$ of access points (APs) and an edge cloud that hosts a set $\mathcal{C} = \{1, 2, \ldots, C\}$ of computing resources (CRs), illustrated in Figure 1. We define the set $\mathcal{R} = \mathcal{A} \cup \mathcal{C}$ of edge (communication and computing) resources.

Tasks and subtask graphs: We consider that WD $i \in \mathcal{N}$ generates latency sensitive computational tasks of type i with intensity λ_i. We model a type i task as a directed acyclic graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, where each node $v \in \mathcal{V}_i$ is a subtask. The source node $v_0^i \in \mathcal{V}_i$ represents wireless transmission of the task's input data via an AP $a \in \mathcal{A}$ to the edge cloud. Nodes $v \in \mathcal{V}_i \setminus \{v_0^i\}$ are computational (execution) subtasks, and correspond to the execution of the functions that constitute the task. The sink node $v^i_{|\mathcal{V}_i|}$ is the last execution subtask, and its completion marks the completion of the task. We denote by $\overline{T}_i$ the maximum average task completion time acceptable to tasks of WD i. A directed edge $e(v_m^i, v_o^i) \in \mathcal{E}_i$ indicates that subtask v_m^i has to finish before subtask v_o^i can start

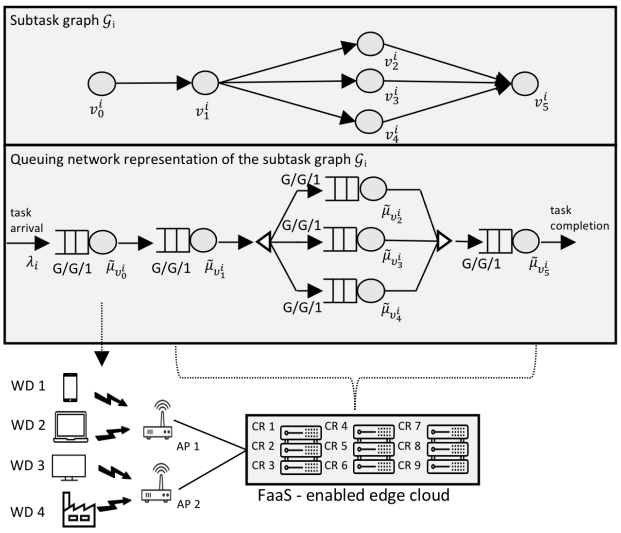

Fig. 1: FaaS-enabled edge cloud infrastructure with $N = 4$ WDs, $A = 2$ APs and $C = 9$ CRs, a fork-join subtask graph $\mathcal{G}_i$, and the corresponding queuing network.

execution. We refer to $\mathcal{G}_i$ as the task graph of WD i, and we consider that the task graphs $\mathcal{G}_i$ represent fork-join type jobs, i.e, subtasks are executed sequentially or in parallel. Finally, we define $\mathcal{V} = \cup_{i \in \mathcal{N}} \mathcal{V}_i$. Observe that for a task of type i the arrival rate of each subtask $v \in \mathcal{V}_i$ is $\lambda_v = \lambda_i$.

Communication and Computing Resources: We denote by $\mathcal{R}_v \subseteq \mathcal{R}$ the set of resources that can be used for performing subtask $v \in \mathcal{V}_i$. For a wireless transmission subtask $v_0^i \in \mathcal{V}_i$ the resources are $\mathcal{R}_{v_0^i} \subseteq \mathcal{A}$, i.e., a subset of the APs, while for execution subtasks $v \in \mathcal{V} \setminus \{\cup_{i \in \mathcal{N}} v_0^i\}$ they are $\mathcal{R}_v \subseteq \mathcal{C}$, i.e., a subset of CRs. Similarly, for a resource $r \in \mathcal{R}$ we define the set $\mathcal{V}_r = \{v \in \mathcal{V} | r \in \mathcal{R}_v\}$ of subtasks that can be performed using resource r. We denote by $\mu_{r,v}$ the service rate at which resource r can process subtask v; thus, μ_{r,v_0^i} is the achievable transmission rate of WD $i \in \mathcal{N}$ when using communication resource $r \in \mathcal{A}$, while for execution subtask $v \in \mathcal{V} \setminus \{\cup_{i \in \mathcal{N}} v_0^i\}$ the service rate is $\mu_{r,v}$ when using CR $r \in \mathcal{C}$. Heterogeneous service rates allow us to model infrastructures with heterogeneous communication and computing resources. Figure 1 illustrates the components of the considered system, including WDs, heterogeneous communication and computing resources and the corresponding modeling abstraction, which maps every subtask to a corresponding G/G/1 queue, resulting in a G/G/1 queuing network as a model of data transmission and subtask graph execution.

A. Edge Resource Allocation

Our model of resource allocation in the serverless edge infrastructure allows resources to be shared dynamically among subtasks. We denote by $p_{r,v}$ the fraction of resource r allocated for processing subtask $v \in \mathcal{V}$, and

by $\mathbf{p} = (p_{r,v})_{r\in\mathcal{R},v\in\mathcal{V}}$ the resulting resource allocation vector. Furthermore, we define the resource utilization $\rho_r = \sum_{v\in\mathcal{V}_r} p_{r,v} \leq 1$, and the vector $\boldsymbol{\rho} \in [0,1]^{|\mathcal{R}|}$, which contains the resource utilizations ρ_r in nonincreasing order. We consider that the processing capacity not allocated at a resource is shared among the subtasks in proportion to their allocations, thus the perceived allocation of resource r available to subtask v is

$$\tilde{p}_{r,v} = \frac{p_{r,v}}{\rho_r}, \tag{1}$$

We denote by $\tilde{\mu}_{r,v} = \tilde{p}_{r,v}\mu_{r,v}$ the resulting perceived service rate for subtask v on resource r, and we express the total perceived service rate for subtask v,

$$\tilde{\mu}_v = \sum_{r\in\mathcal{R}_v} \tilde{\mu}_{r,v}. \tag{2}$$

Similar to existing serverless offerings and to bandwidth SLAs in 5G networks [8], we consider that users can reserve computing capacity and communication resources. The ability to reserve compute capacity is akin to provisioned concurrency in existing serverless offerings[1] Nonetheless, unlike in existing commercial offerings, for simplicity we define the reservation in terms of processing rate (instead of processing capacity). This formulation is reasonable, as users can know the average service times of their subtasks. We denote by σ_{v_i} the service rate reservation made by WD $i \in \mathcal{N}$ for its subtask $v_i \in \mathcal{V}_i$. Furthermore, we denote by $\sigma_i = \sum_{v_i\in\mathcal{V}_i} \sigma_{v_i}$ the total rate reservation of WD i. Throughout the paper we consider that $\sigma_{v_i} = \frac{\sigma_i}{|\mathcal{V}_i|}, \forall v_i \in \mathcal{V}_i$, i.e., WDs make the same service rate reservation for all of their subtasks. We make this assumption for two reasons. First, a uniform allocation of service rates to the servers minimizes the mean sojourn time in a tandem network of $M/M/1$ queues. It may not be optimal for non $M/M/1$ queues, but it is likely not too far from optimal. Second, this model allows for a simple interaction between the users and the infrastructure as each user can reserve resources through a single parameter independent of the number of subtasks in its task graph, providing ease of use for customers. Considering non-homogeneous rate reservations could be an interesting extension of our work.

Load-balancing Network Operator: To effectively serve user requests, we consider that the network operator performs load balancing periodically. It does so by minimizing $\boldsymbol{\rho}$, i.e., the vector of the utilization of communication and computing resources, in the lexicographical sense, subject to rate stability constraints[2]. Thus, the operator periodically solves the optimization problem

$$\operatorname*{lex\,min}_{\mathbf{p}} \boldsymbol{\rho} \tag{3}$$

s.t.

$$\sigma_v \leq \sum_{r\in\mathcal{R}_v} p_{r,v}\mu_{r,v}, \forall v \in \mathcal{V} \tag{4}$$

$$\rho_r = \textstyle\sum_{v\in\mathcal{V}_r} p_{r,v}, \forall r \in \mathcal{R}, \tag{5}$$

$$\sigma_v = \tfrac{\sigma_i}{|\mathcal{V}_i|}, \forall v \in \mathcal{V}_i, \tag{6}$$

$$p_{r,v} = 0, \forall r \in \mathcal{R}, v \notin \mathcal{V}_r \tag{7}$$

$$p_{r,v} \geq 0, \forall r \in \mathcal{R}, v \in \mathcal{V}_r. \tag{8}$$

Constraint (4) ensures that each subtask receives the reserved rate and allows WD i to adjust the sojourn time for subtask v (c.f., Kingman's approximation of the waiting time in a G/G/1 queue [9]), constraint (5) defines the utilization of each resource $r \in \mathcal{R}$ under resource allocation vector $\mathbf{p}$, constraint (6) enforces resources to be allocated uniformly among execution subtasks of a WD, and constraints (7) and (8) ensure that the allocation of resources to the subtasks respects assignment constraints.

The resource allocation implemented by the operator determines the perceived service rates of the subtasks, and together with the task arrival rates it determines the average task completion times of the users. To express this dependence, we define the collection $\boldsymbol{\lambda} = (\lambda_i)_{i\in\mathcal{N}}$ of arrival intensities of the WDs. Similarly, we define the collection $\boldsymbol{\sigma} = (\sigma_i)_{i\in\mathcal{N}}$ of resource reservations of the WDs. Finally, we denote by $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ the mean completion time of tasks generated by WD i, which in our model equals the mean sojourn time of customers in a G/G/1 fork-join queuing network corresponding to the subtask graph $\mathcal{G}_i$.

B. User Utility

Aligned with the pay-as-you-go billing model widely used in serverless computing, we denote by c_i^λ and c_i^σ the unit cost per arrival rate and per resource reservation, respectively, and we define the computing cost for WD i as $C_i(\lambda_i, \sigma_i) = c_i^\lambda\lambda_i + c_i^\sigma\sigma_i$. The term c_i^λ accounts for the cost due to the number of invocations, but it can also account for the computational resources actually used for executing tasks, as usual in existing serverless offerings. Furthermore, we define the utility of WD i,

$$U_i(\lambda_i, \sigma_i) = f_i(\lambda_i) - C_i(\lambda_i, \sigma_i), \tag{9}$$

where $f_i(\lambda_i)$ is a continuously differentiable concave function of λ_i, i.e., $\frac{d^2 f}{d\lambda_i^2} < 0$. Concavity of the utility is a natural assumption for many monitoring and control applications, and is widely used as it captures diminishing marginal gains [10]–[14], while differentiability ensures analytical tractability. We also make the reasonable assumptions that $f_i(0) = 0$ and $c_i^\lambda < \frac{df}{d\lambda_i}|_{\lambda_i=0} \leq L_i \in \mathbb{R}_{>0}$.

Since the WDs pay for the rate at which they generate tasks and for the resource reservations they make (c.f. equation (9)), for each WD $i \in \mathcal{N}$ there exists a maximum

[1] Amazon Lambda allows function instances to be kept initialized, called provisioned concurrency.

[2] Let $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \mathbb{R}_{\geq 0}^{|\mathcal{R}|}$. Then $\boldsymbol{\rho} <_L \boldsymbol{\rho}'$ (smaller according to the lexicographical order) if and only if there exists $1 \leq r' \leq |\mathcal{R}|$ such that for $r < r'$ we have $\boldsymbol{\rho}_r = \boldsymbol{\rho}'_r$ and $\boldsymbol{\rho}_{r'} < \boldsymbol{\rho}'_{r'}$. Given that $\boldsymbol{\rho}$ consists of the utilizations in non-increasing order, lexicographical minimization results in a particular min-max solution. The two are equivalent for $|\mathcal{R}| = 2$.

rate $\overline{\lambda}_i$ and a maximum resource reservation $\overline{\sigma}_i$, which can be obtained as the solution to $\frac{\partial U_i}{\partial \lambda_i}(\overline{\lambda}_i, 0) = 0$ and to $U_i(\overline{\lambda}_i, \overline{\sigma}_i) = 0$, respectively. Therefore, we can consider that WD $i \in \mathcal{N}$ chooses σ_i from the compact set $\mathcal{S}_i = [\underline{\sigma}_i, \overline{\sigma}_i]$ and λ_i from the compact set $[\underline{\lambda}_i, \overline{\lambda}_i]$, for some $\underline{\sigma}_i \geq 0$ and $\underline{\lambda}_i \geq 0$.

$\mathcal{N}$	Set of WDs
N	Number of WDs
$\mathcal{A}$	Set of APs
A	Number of APs
$\mathcal{C}$	Set of CR
C	Number of CR
$\mathcal{R}$	Set of resources ($\mathcal{R} = \mathcal{C} \cap \mathcal{A}$)
i	Index of WDs
λ_i	Arrival intensity of WD i
$\boldsymbol{\lambda}$	Task intensity vector $(\lambda_i)_{i\in\mathcal{N}}$
$\boldsymbol{\lambda}_{-i}$	Task intensity vector except WD i
$\mathcal{V}_i$	Set of subtasks (i.e. nodes) for WD i
v_0^i	Wireless transmission subtask of WD i
$v_{\lvert\mathcal{V}_i\rvert}^i$	Last subtask (i.e. sink node) of WD i
$\overline{T}_i$	Maximum average completion time
$\mathcal{R}_v$	Set of resources used for subtask v
$\mu_{r,v}$	Service rate of r for processing subtask v
$p_{r,v}$	Fraction of resource r for subtask v
$\mathbf{p}$	Resource allocation vector $(p_{r,v})_{r\in\mathcal{R},v\in\mathcal{V}}$
ρ_r	Utilization of resource r, $\rho_r = \sum_{v\in\mathcal{V}_r} p_{r,v} \leq 1$
$\boldsymbol{\rho}$	Vector of utilization $\forall r \in \mathcal{R}, \boldsymbol{\rho} \in [0,1]^{\lvert\mathcal{R}\rvert}$
$\tilde{p}_{r,v}$	Perceived allocation for subtask v on resource r
$\tilde{\mu}_{r,v}$	Perceived service rate for subtask v on resource r
$\tilde{\mu}_v$	Total perceived service rate for subtask v
σ_v	Service rate reservation for subtask v
σ_i	Total reservation of WD i
$\boldsymbol{\sigma}$	Reservation vector $(\sigma_i)_{i\in\mathcal{N}}$
$\boldsymbol{\sigma}_{-i}$	Reservation vector except WD i
$\overline{S}_i$	Mean task completion time of WD i
c_i^λ	Cost per task intensity of WD i
c_i^σ	Cost per reservation of WD i
C_i	Computing cost of WD i
U_i	Utility of WD i

TABLE I: Table of Notations.

C. Serverless Stochastic Rate Allocation Game

In the considered system the WDs are engaged in repeated strategic interaction through the resource allocation $\mathbf{p}$, which they can influence through the resource reservations $\boldsymbol{\sigma}$. We consider that the WDs can update their resource reservations $\boldsymbol{\sigma}$ periodically, i.e., whenever the network operator updates the resource allocation $\mathbf{p}$ by solving (3)-(8). Between subsequent updates of the resource reservation the WDs can adjust their rates $\boldsymbol{\lambda}$. We adopt the game theoretic notation $\boldsymbol{\sigma}_{-i}$ and $\boldsymbol{\lambda}_{-i}$ to denote the resource reservations and the rates of all WDs except WD i, respectively.

Each WD $i \in \mathcal{N}$ aims at maximizing its utility (9) subject to its average task completion time constraint $\overline{T}_i$, by choosing resource reservation σ_i and rate λ_i. Thus, each WD i aims at solving the optimization problem

$$\arg\max_{\lambda_i,\sigma_i} U_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \sigma_i, \boldsymbol{\sigma}_{-i}) \tag{10}$$

$$\text{s.t. } \bar{S}_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \sigma_i, \boldsymbol{\sigma}_{-i}) \leq \overline{T}_i \tag{11}$$

The resulting game played by the WDs is a dynamic game in which not only the objective functions of WDs depend on each others' strategies, but also the strategy sets through stochastic constraints. Importantly, in practice the mean task sojourn times, and thus, the action sets are not known, but have to be learned by the WDs. We refer to the resulting game as the *Serverless Stochastic Rate Allocation* (SSRA) game. In what follows we investigate (i) whether the SSRA game admits an equilibrium, and (ii) whether WDs could learn an equilibrium strategy in a distributed manner.

III. Sojourn Time Characterization

In this section we first show monotonicity and pseudoconvexity of the mean task completion time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$, i.e., the sojourn time in a G/G/1 fork-join network, in the task arrival rate λ_i. We then characterize the structure of the optimal solution of the operator's load balancing problem (3)-(8), and finally we show monotonicity of the mean task completion time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ in the resource reservation σ_i. We use these results in Section IV and V.

A. Monotonicity and Pseudoconvexity of the Sojourn Time in the Arrival Rate

It is known that even in a single G/G/1 queue with FCFS service discipline the mean sojourn time need not be a convex function of the arrival rate [15]. Nonetheless, in what follows we show that the mean sojourn time is a monotone, pseudoconvex function of the arrival rate. The importance of this result is that pseudoconvexity is a sufficient condition for gradient-based learning algorithms to converge to the optimal solution.

We start with showing the result for tandem queues; we consider a set $\mathcal{V} = \{1, 2, \ldots, V\}$ of G/G/1 queues in series, and we assume that the service discipline is FCFS and work-conserving (i.e., a server is never idle when its queue is non-empty). We make the common assumption that the interarrival and service time distributions satisfy the stability criterion [16], [17]. We denote by I_n^v the time between the arrival of customer $n-1$ and customer n to queue $v \in \mathcal{V}$. Furthermore, we denote by s_n^v, w_n^v and S_n^v

the service, waiting and sojourn times of customer n in queue $v \in \mathcal{V}$, respectively, and we introduce the notation

$$\mathcal{I}^v_{l,m} = \sum_{k=l}^{m} I^v_k, \quad {}_{\Sigma}S^v_{l,m} = \sum_{k=l}^{m} s^v_k, \quad S^{1:V}_n = \sum_{v=1}^{V} S^v_n, \tag{12}$$

where for $l > m$ the sums are empty and are thus 0. Before we present our results, let us recall two fundamental results concerning the waiting times and the sojourn times in tandem queues, respectively. We first present the waiting time expression for a single G/G/1 queue, and then we extend the result to tandem queues.

Lemma 1. *[16] Let m_v represent the m^{th} customer in queue v. Lindley's recursion has the unique solution*

$$w^v_n = \max_{m_v \le n} ({}_{\Sigma}S^v_{m_v,n-1} - \mathcal{I}^v_{m_v+1,n}), v \in \mathcal{V}.$$

The second result follows from Lemma 1 and provides a closed-form expression for the sojourn time of G/G/1 tandem queues.

Lemma 2. *[17] The total time $S^{1:V}_n$ that customer n spends in a system of $V \ge 1$ queues connected in series can be expressed as*

$$S^{1:V}_n = \max_{m_1 \le \dots m_V \le m_{V+1}=n} \Big(\sum_{v=1}^{V} {}_{\Sigma}S^v_{m_v,m_{v+1}} - \mathcal{I}^1_{m_1+1,n} \Big). \tag{13}$$

We note that both results hold for stable queuing systems, including the heavy traffic regime, whenever the offered load is less than 1. In what follows we prove our first main result concerning the sojourn time of individual tasks based on Lemma 2.

Theorem 1. *Consider a G/G/1 tandem queue consisting of V queues, and an arbitrary customer n. The total sojourn time $S^{1:V}_n$ of customer n is an increasing pseudoconvex function of the customer arrival rate λ.*

Proof. For an arrival rate of λ, let us denote by $\tau_{k-1} = \frac{1}{\lambda} t_{k-1}$ and $\tau_k = \frac{1}{\lambda} t_k$ the time at which customers $k-1$ and k arrive in the system (i.e., in the first queue), respectively. t_{k-1} and t_k can assume any non-negative values and they can be any realizations of random variables. Then, the interarrival time of customer k and customer $k-1$ at the first queue is $I^1_k = \tau_k - \tau_{k-1} = \frac{t_k - t_{k-1}}{\lambda}$. Therefore, it follows from (12) and Lemma 2 that the total sojourn time $S^{1:V}_n$ of customer n can be expressed as

$$S^{1:V}_n = \max_{m_1 \le \dots m_V \le m_{V+1}=n} \Big(\sum_{v=1}^{V} {}_{\Sigma}S^v_{m_v,m_{v+1}} - \sum_{k=m_1+1}^{n} \frac{t_k - t_{k-1}}{\lambda} \Big). \tag{14}$$

First observe that for two successive jobs $k-1$ and k we have that $t_k - t_{k-1} > 0$. Furthermore, since ${}_{\Sigma}S^v_{m_v,m_{v+1}}$ is not a function of λ (c.f. equation (12)), we have that $S^{1:V}_n$ is defined as the maximum of increasing functions, is continuous, but it is not necessarily a differentiable function of λ. Therefore, to prove pseudoconvexity of $S^{1:V}_n$ we need to consider the upper Dini derivative of $S^{1:V}_n$, which we denote by $D^+S^{1:V}_n$. It is easy to see from (14) that $S^{1:V}_n$ is an increasing function of λ such that $D^+S^{1:V}_n(\lambda') > 0$ for any $\lambda' > 0$. To prove pseudoconvexity, we need to show that $S^{1:V}_n$ is increasing in any direction where the upper Dini derivative is positive. Since $D^+S^{1:V}_n(\lambda') > 0$ for any $\lambda' > 0$, we have that $D^+S^{1:V}_n(\lambda')(\lambda'' - \lambda') \ge 0$ is true only if $\lambda' \le \lambda''$. Therefore, to check pseudoconvexity it suffices to show that $\lambda' \le \lambda''$ implies $S^{1:V}_n(\lambda') \le S^{1:V}_n(\lambda)$ for all λ on the line segment connecting λ' and λ'', i.e., that $S^{1:V}_n$ is nondecreasing in λ, which is clearly the case. This proves the theorem. □

Next, we extend the above result to fork-join networks.

Theorem 2. *Consider a G/G/1 fork-join network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of queues with FCFS and work-conserving service discipline. Then the sojourn time S_n of customer n is an increasing pseudoconvex function of the arrival rate λ.*

Proof. Let us denote by $\Pi = \{v_1, \dots, v_{|\Pi|}\}$ the set of parallel queues and let v_0 and $v_{|\mathcal{V}|}$ be the first and the last queue in the network, i.e., $\mathcal{V} = \{v_0\} \cup \Pi \cup \{v_{|\mathcal{V}|}\}$, respectively. Furthermore, let us denote by $S^{p_\pi}_n$ the sojourn time of customer n on the simple path $p_\pi = \{(v_0, v_\pi), (v_\pi, v_{|\mathcal{V}|})\}$, which connects the first queue v_0 with the last queue $v_{|\mathcal{V}|}$ via parallel queue $v_\pi \in \Pi$. Then, the total sojourn time S_n of customer n in the fork-join network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be expressed as

$$S_n = \max_{\pi \in \Pi} S^{p_\pi}_n. \tag{15}$$

By Theorem 1 we know that $S^{p_\pi}_n$ is an increasing pseudoconvex function of λ. Furthermore, it is easy to see from (14) and (15) that S_n is also an increasing function of λ with the upper Dini derivative $D^+S_n > 0$. By following a similar approach to the one used in the proof of Theorem 1 it follows that S_n is also pseudoconvex in λ, which proves the result. □

Finally, we extend the result to the mean sojourn times.

Theorem 3. *The mean sojourn time $\bar{S}$ in a G/G/1 fork-join network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an increasing pseudoconvex function of the arrival rate λ.*

Proof. Since Theorem 1 is true for any non-negative values of t_{k-1} and t_k (c.f., equation (14)), it is also true when the realizations of t_{k-1} and t_k are random variables, hence the result. □

Using the above we can obtain a useful characterization of the service times of the tasks generated by the WDs in the considered serverless edge computing system.

Corollary 1. *The mean sojourn time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ of a task generated by WD $i \in \mathcal{N}$ is an increasing pseudoconvex function of the task arrival rate λ_i.*

Proof. The result follows from Theorem 3. □

B. Perceived Service Rate under Load Balancing

We now turn our attention to the perceived service rate $\tilde{\mu}_v^*$ of the WDs. In order to obtain a characterization, we first analyze the structure of an optimal solution of the operator's problem (3)-(8).

Proposition 1. *Consider an optimal solution $(\boldsymbol{p}^*, \boldsymbol{\rho}^*)$ to (3)-(8), a subtask $v \in \mathcal{V}$ and a subset $\mathcal{R}'_v \subseteq \mathcal{R}_v$ of resources such that $p^*_{r,v} > 0$ for every $r \in \mathcal{R}'_v$. Then, the solution $\boldsymbol{p}^*$ is such that*

(i) equality holds in each constraint (4) and
(ii) $\rho^*_r = \rho^*_{r''}$ holds for any two resources $r, r'' \in \mathcal{R}'_v$.

Proof. We start with proving (i). Let us assume that there is an optimal solution $\mathbf{p}^*$ to (3)-(8) such that $\sigma_v < \sum_{r\in\mathcal{R}_v} p^*_{r,v}\mu_{r,v}$ holds for some subtask $v \in \mathcal{V}$. Next, let us consider $\mathbf{p}'$ such that $p'_{r,v} < p^*_{r,v}$ holds for some resource $r \in \mathcal{R}'_v$, $p'_{r'',w} = p^*_{r'',w}$ holds for $(r'', w) \in \mathcal{R}\times\mathcal{V}\setminus\{(r, v)\}$, and $\sigma_v = \sum_{r\in\mathcal{R}_v} p'_{r,v}\mu_{r,v}$ is satisfied. Then, $\rho'_r < \rho^*_r$ and $\rho'_{r''} = \rho^*_{r''}$, $r'' \in \mathcal{R}\setminus\{r\}$ hold. Since $\boldsymbol{\rho}$ contains the utilizations of resources in nonincreasing order we obtain that $\boldsymbol{\rho}' \preceq_L \boldsymbol{\rho}^*$, which contradicts the assumption that $(\mathbf{p}^*, \boldsymbol{\rho}^*)$ is an optimal solution to (3)-(8), and proves (i).

We continue with proving (ii). Let us assume that there is an optimal solution $\mathbf{p}^*$ to (3)-(8) such that $\rho^*_r > \rho^*_{r''}$ holds for two resources $r, r'' \in \mathcal{R}'_v$. Furthermore, let us consider $\mathbf{p}'$ where $p'_{r,v} < p^*_{r,v}$, $p'_{r'',v} > p^*_{r'',v}$ and $p'_{r',w} = p^*_{r',w}$, $(r', w) \in \mathcal{R}\times\mathcal{V}\setminus\{(r, v), (r'', v)\}$ hold, and $\rho'_r = \rho'_{r''}$ is satisfied. Then, $\rho'_r = \rho'_{r''} \leq \rho^*_r$ and $\rho'_{r'} = \rho^*_{r'}$, $r' \in \mathcal{R}\setminus\{r, r''\}$ hold. Since $\boldsymbol{\rho}$ contains the utilizations of resources in nonincreasing order we obtain that $\boldsymbol{\rho}' \preceq_L \boldsymbol{\rho}^*$, which contradicts the assumption that $(\mathbf{p}^*, \boldsymbol{\rho}^*)$ is an optimal solution to (3)-(8), and proves (ii). This concludes the proof. □

Proposition 1 allows us to formulate the following results.

Corollary 2. *Consider an optimal solution $(\boldsymbol{p}^*, \boldsymbol{\rho}^*)$, a subtask $v \in \mathcal{V}$, a subset $\mathcal{R}'_v \subseteq \mathcal{R}_v$ of resources such that $p^*_{r,v} > 0$ for every $r \in \mathcal{R}'_v$, and a resource $r'' \in \mathcal{R}'_v$. Then the perceived service rate is*

$$\tilde{\mu}^*_v = \frac{\sigma_v}{\rho^*_{r''}} = \frac{\sigma_i}{|\mathcal{V}_i|\rho^*_{r''}}. \tag{16}$$

Proof. First, from (ii) in Proposition 1 we have that $\rho^*_r = \rho^*_{r''}$ for any resource $r \in \mathcal{R}'_v \setminus \{r''\}$, and thus the perceived service rate $\tilde{\mu}^*_v$ defined in (2) can be expressed as $\tilde{\mu}^*_v = \sum_{r\in\mathcal{R}'_v} \frac{p^*_{r,v}\mu_{r,v}}{\rho^*_r} = \frac{\sum_{r\in\mathcal{R}'_v} p^*_{r,v}\mu_{r,v}}{\rho^*_{r''}}$. Second, from (i) in Proposition 1 we have that $\sum_{r\in\mathcal{R}'_v} p^*_{r,v}\mu_{r,v} = \sigma_v = \frac{\sigma_i}{|\mathcal{V}_i|}$, which proves the result. □

Corollary 3. *The perceived service rate $\tilde{\mu}^*_v$ of every subtask $v \in \mathcal{V}_i$ is a nondecreasing function of the resource reservation σ_i.*

We can provide a stronger result if we restrict our attention to the case that resources form equivalence classes, defined as follows.

Assumption 1 (A1). *Consider subtasks $v, v' \in \mathcal{V}_i$. If $\mathcal{R}_v \cap \mathcal{R}_{v'} \neq \emptyset$ then $\mathcal{R}_v = \mathcal{R}_{v'}$.*

Corollary 4. *Under Assumption A1 the utilization $\rho^*_{r''}$ is an affine function of σ_i. Furthermore, the perceived service rate $\tilde{\mu}^*_v$ of every subtask $v \in \mathcal{V}_i$ is a concave nondecreasing function of the resource reservation σ_i.*

We proceed with providing a general result concerning the sojourn time in a fork-join network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. To do so, we denote by μ_v the service rate in queue $v \in \mathcal{V}$.

Theorem 4. *Consider a fork-join network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of G/G/1 queues with FCFS and work-conserving service discipline. The sojourn time S_n of customer n and the mean sojourn time $\bar{S}$ are decreasing functions of the service rates μ_v.*

Proof. Let us consider three customers l, n and m such that $l \leq n \leq m$. For a service rate of μ_v we can express the time required to serve customer n as $s^v_n = \frac{x_{n,v}}{\mu_v}$, where $x_{n,v}$ is a non-negative random variable with $E[x_{n,v}] = 1$. For any realization of $x_{n,v}$, it follows from the definitions of ${}_\Sigma S^v_{l,m}, S^{1:V}_n$ and S_n (c.f., equations (12),(14) and (15)) that the sojourn time S_n of customer n is a decreasing function of service rate μ_v in queue $v \in \mathcal{V}$. Taking expectation, it follows that the mean sojourn time $\bar{S}$ in a fork-join network is also a decreasing function of the service rate μ_v in queue $v \in \mathcal{V}$, which proves the result. □

Theorem 4 allows us to formulate the following result.

Corollary 5. *The mean sojourn time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ of a task generated by WD $i \in \mathcal{N}$ is a decreasing function of the perceived service rate $\tilde{\mu}_v$ for each subtask $v \in \mathcal{V}_i$.*

Proof. The result follows from the proof of Theorem 4. □

Finally, we use the above result to show that the mean sojourn time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is a monotonic function of the resource reservation σ_i.

Theorem 5. *Consider an optimal solution to the operator's problem (3)-(8), and the resulting perceived service rates $\tilde{\mu}^*_v$ of subtasks $v \in \mathcal{V}_i$. The mean sojourn time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ of a task generated by WD $i \in \mathcal{N}$ is a nonincreasing function of the resource reservation σ_i.*

Proof. First, from Corollary 5 we have that the mean sojourn time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is a decreasing function of the perceived service rate $\tilde{\mu}^*_v$ for each subtask $v \in \mathcal{V}_i$. Second, from Corollary 3 we have that the perceived service rate $\tilde{\mu}^*_v$ of each subtask $v \in \mathcal{V}_i$ is a nondecreasing function of resource reservation σ_i. Hence, we have that $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is nonincreasing in σ_i, which proves the result. □

IV. Equilibria Under Perfect Information

We first consider the case of perfect information, i.e., each WD i knows $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$, and can infer its mean task

completion time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$. Observe that the sets of feasible rates and reservations of players form coupled constraints, and hence the resulting game is a generalized Nash equilibrium problem. In what follows we use *Variational Inequality* (VI) theory to prove the existence of equilibria in the SSRA game under perfect information. First, we recall the definition of a VI$(\mathcal{K}, F)$ problem from [18].

Definition 1. Let $\mathcal{K} \subseteq \mathbb{R}^n$ be a closed convex set and $F : \mathcal{K} \to \mathbb{R}^n$ a continuous function. The VI$(\mathcal{K}, F)$ problem is to find a point $x^* \in \mathcal{K}$ such that $F(x^*)^T(x - x^*) \geq 0$, for $\forall x \in \mathcal{K}$.

We are now ready to formulate one of our main results.

Theorem 6. *The SSRA game under perfect information admits a pure strategy Nash equilibrium.*

Proof. First, let us recall that the WDs can update their resource reservations periodically, and that between two updates of the resource reservations they can adjust their rates. In order to model the dynamics of the game played by the WDs, we introduce two fictitious players i^σ and i^λ for each WD $i \in \mathcal{N}$, which decide about the resource reservation σ_i and the rate λ_i, respectively. Furthermore, we denote by $\mathcal{N}_\sigma, |\mathcal{N}_\sigma| = N$ and $\mathcal{N}_\lambda, |\mathcal{N}_\lambda| = N$ the sets of fictitious players that decide about the resource reservations and rates, respectively. Finally, we denote by $\mathcal{N}_f$ the set of all fictitious players, i.e., $\mathcal{N}_f = \mathcal{N}_\lambda \cup N_\sigma$.

In order to model how the fictitious players interact with each other we define for each $i_\sigma \in \mathcal{N}_\sigma$ the set $\mathcal{K}_{i_\sigma}(\boldsymbol{\lambda}, \boldsymbol{\sigma}_{-i}) \triangleq \{\sigma_i | \bar{S}_i(\boldsymbol{\lambda}, \sigma_i, \boldsymbol{\sigma}_{-i}) \leq \overline{T}_i\}$ of feasible resource reservations, and for each $i_\lambda \in \mathcal{N}_\lambda$ the set $\mathcal{K}_{i_\lambda}(\boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma}) \triangleq \{\lambda_i | \bar{S}_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma}) \leq \overline{T}_i\}$ of feasible rates. We can then define the generalized Nash equilibrium problem (GNEP) $\Gamma^f =< \mathcal{N}_f, (\mathcal{K}_{i_f})_{i_f \in \mathcal{N}_f}, (U_{i_f}(\boldsymbol{\lambda}, \boldsymbol{\sigma}))_{i_f \in \mathcal{N}_f} >$ in which both fictitious players i_σ and i_λ aim at maximizing utility $U_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ of WD i with respect to the latency constraint of WD i. Therefore, Γ^f is a strategic game in which each fictitious player $i_\sigma \in \mathcal{N}_\sigma$ aims at maximizing its utility $U_{i_\sigma}(\boldsymbol{\lambda}, \boldsymbol{\sigma}) = U_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ by solving

$$\arg\max_{\sigma_i} U_i(\boldsymbol{\lambda}, \sigma_i, \boldsymbol{\sigma}_{-i}) \tag{17}$$

$$\text{s.t. } \sigma_i \in \mathcal{K}_i^\sigma(\boldsymbol{\lambda}, \boldsymbol{\sigma}_{-i}), \tag{18}$$

and each fictitious player $i_\lambda \in \mathcal{N}_\lambda$ aims at maximizing its utility $U_{i_\lambda}(\boldsymbol{\lambda}, \boldsymbol{\sigma}) = U_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ by solving

$$\arg\max_{\lambda_i} U_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma}) \tag{19}$$

$$\text{s.t. } \lambda_i \in \mathcal{K}_i^\lambda(\boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma}). \tag{20}$$

Clearly, a pure strategy Nash equilibrium of Γ^f is an equilibrium of the SSRA game in which the WDs update their resource reservations and rates separately. We thus have to prove that Γ^f has a pure strategy Nash equilibrium.

In the following we use VI to prove the result concerning the existence of equilibria in Γ^f. Therefore, we need to define a suitable VI$(\mathcal{K}, F)$ problem that corresponds to game Γ^f. To do so, we have to specify the set $\mathcal{K}$ and the function F [18]–[20]. First, we define the set

$$\mathcal{K} = \Pi_{i_\sigma \in \mathcal{N}_\sigma} \mathcal{K}_{i_\sigma}(\boldsymbol{\lambda}, \boldsymbol{\sigma}_{-i}) \Pi_{i_\lambda \in \mathcal{N}_\lambda} \mathcal{K}_{i_\lambda}(\boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma}). \tag{21}$$

Second, we define the function

$$F = \begin{pmatrix} \nabla_\sigma U(\boldsymbol{\lambda}, \boldsymbol{\sigma}) \\ \nabla_\lambda U(\boldsymbol{\lambda}, \boldsymbol{\sigma}) \end{pmatrix},$$

where $\nabla_\sigma U(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ and $\nabla_\lambda U(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ are the gradient vectors given by

$$\nabla_\sigma U(\boldsymbol{\lambda}, \boldsymbol{\sigma}) = \begin{pmatrix} \frac{dU_1(\boldsymbol{\lambda}, \boldsymbol{\sigma})}{d\sigma_1} \\ \vdots \\ \frac{dU_N(\boldsymbol{\lambda}, \boldsymbol{\sigma})}{d\sigma_N} \end{pmatrix}, \nabla_\lambda U(\boldsymbol{\lambda}, \boldsymbol{\sigma}) = \begin{pmatrix} \frac{dU_1(\boldsymbol{\lambda}, \boldsymbol{\sigma})}{d\lambda_1} \\ \vdots \\ \frac{dU_N(\boldsymbol{\lambda}, \boldsymbol{\sigma})}{d\lambda_N} \end{pmatrix}.$$

The proof relies on showing that that set $\mathcal{K}$ is compact and convex and that the utility $U_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ of each WD i is continuously differentiable in $(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ and concave in σ_i and λ_i [20]. We start with proving the compactness of set $\mathcal{K}$. Let us recall that WD $i \in \mathcal{N}$ can choose σ_i and λ_i from the compact sets $[\underline{\sigma}_i, \overline{\sigma}_i]$ and $[\underline{\lambda}_i, \overline{\lambda}_i]$, respectively. Therefore, it is easy to see that $\mathcal{K}_{i_\sigma}(\boldsymbol{\lambda}, \boldsymbol{\sigma}_{-i})$ and $\mathcal{K}_{i_\lambda}(\boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma})$ are compact subsets of $[\underline{\sigma}_i, \overline{\sigma}_i]$ and $[\underline{\lambda}_i, \overline{\lambda}_i]$, respectively. Since the Cartesian product of compact sets is compact (c.f., Tychonoff's theorem), we obtain that set $\mathcal{K}$ defined in (21) is compact.

We continue with proving the convexity of set $\mathcal{K}$. From Corollary 1 and Theorem 5 we have that $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is an increasing pseudoconvex function of the task arrival rate λ_i and a nonincreasing function of σ_i, respectively. Therefore, $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is quasiconvex in λ_i and in σ_i, and thus sublevel sets $\mathcal{K}_{i_\lambda}(\boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma})$ and $\mathcal{K}_{i_\sigma}(\boldsymbol{\lambda}, \boldsymbol{\sigma}_{-i})$ are convex [21]. Since the Cartesian product of convex sets is a convex set [21] we obtain that the set $\mathcal{K}$ defined in (21) is convex as well.

Finally, it is easy to check that the utility function $U_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ defined in (9) is continuously differentiable in $(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ and concave in σ_i and λ_i. Hence, it follows from Theorem 2.1 and Proposition 2.2 in [20] that the solution of VI$(\mathcal{K}, F)$ exists and it is also a Nash equilibrium of Γ^f, and thus of the SSRA game. This proves the theorem. □

We have thus shown that equilibria exist in the SSRA game under perfect information, which is a prerequisite for the study of learning equilibria under imperfect information considered in the following section. In the Appendix, included in the supplementary material, we also show that rate reservation is essential in the considered problem, as the interaction between rate control and resource allocation may lead to starvation otherwise. Next, we study whether equilibria can be reached under imperfect information.

V. Learning to Play Equilibrium using Online Optimization

In what follows we propose an online optimization algorithm for WDs to maximize their individual utility

Result: Resource reservation σ_i of WD i

1 **for** $t = 1, \dots$ **do**
2 $\quad \eta_t = \frac{1}{t^{\gamma 2}}, \alpha_t = t^{\gamma 1}$
3 $\quad \sigma_i{}^-(t) = \sigma_i(t) - \frac{\eta_t}{2}$,
4 $\quad$ Report $\sigma_i{}^-(t)$ to Operator;
5 $\quad (S_i^-, \lambda_i^-) \leftarrow RA(\lambda_i(t), \overline{T}_i)$ ▷ Rate adaptation;
6 $\quad \sigma_i{}^+(t) = \sigma_i(t) + \frac{\eta_t}{2}$;
7 $\quad$ Report $\sigma_i{}^+(t)$ to Operator;
8 $\quad (S_i^+, \lambda_i^+) \leftarrow RA(\lambda_i(t), \overline{T}_i)$ ▷ Rate adaptation;
$\quad$ `// Subgradient computation`
9 $\quad \hat{U}_i^-(t) = U_i(\lambda_i^-, \sigma_i^-(t))$;
10 $\quad \hat{U}_i^+(t) = U_i(\lambda_i^+, \sigma_i^+(t))$;
11 $\quad \nabla \hat{U}_i(t) \leftarrow (\hat{U}_i^+(t) - \hat{U}_i^-(t))/\eta_t$;
12 $\quad \lambda_i(t+1) = (\lambda_i^+ + \lambda_i^-)/2$
13 $\quad \sigma_i(t+1) =$
$\quad\quad \arg\min_{\sigma \in \mathcal{S}_i} -[\nabla \hat{U}_i(t)]^\intercal(\sigma - \sigma_i(t)) + \frac{1}{\alpha_t}||\sigma - \sigma_i(t)||^2$;
14 **end**

Algorithm 2: Pseudocode of the OARC algorithm.

based on measured sojourn times of their computational tasks, called OARC. The pseudo-code of the algorithm is shown in Figure 2. The algorithm makes use of online gradient ascent based on a perturbation of σ_i, used for estimating the gradient of the utility function U_i, and in between perturbations it ensures that the latency constraint is met through rate adaptation (RA). In each iteration, the algorithm first updates the perturbation size (η_t), and the learning rate (α_t) (Line 1). It then computes the perturbed reservations $(\sigma_i^-(t), \sigma_i^+(t))$ and reports those to the operator (Lines 3, 4 and 6, 7). WDs estimate the resulting arrival intensities and average response times $((S_i^+, \lambda_i^+),(S_i^-, \lambda_i^-))$ corresponding to the rate reservations $(\sigma_i^-(t), \sigma_i^+(t))$ (Lines 5 and 8). Between Lines $9-11$, the algorithm computes the stochastic subgradient with respect to the rate reservation. Finally, it computes the estimated arrival rate and updates the reservation using a gradient ascent step, based on the computed stochastic subgradient (Lines $12-13$). In what follows we first show that the proposed algorithm can indeed ensure to meet the mean sojourn time constraint, and that under certain assumptions it converges to an equilibrium.

Proposition 2. *Let σ_i be fixed, and $\lambda_i^*(\sigma_i) = \arg\max_{\lambda_i \in [\underline{\lambda}_i, \overline{\lambda}_i]} U_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma})$. Then the set of solutions of the problem*

$$\min_{\lambda_i \le \lambda_i^*(\sigma_i)} [\bar{S}_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma}) - \overline{T}_i]^2. \tag{22}$$

is compact and convex.

Proof. We prove the result by first showing convexity and compactness of the solution set. By Corollary 1 $\bar{S}_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma})$ is increasing and pseudoconvex in λ_i. Thus, the objective $[\bar{S}_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma}) - \overline{T}_i]^2$ is pseudoconvex. Pseudoconvexity implies quasiconvexity, and every sublevel set of a quasiconvex function is convex, which together with the finiteness of $\lambda_i^*(\sigma_i)$ proves the result. □

Observe that pseudoconvexity of the objective in (22) implies that stochastic gradient descent algorithms, such as stochastic approximation and the Adam algorithm [22] can be used for finding a solution efficiently (c.f., Theorem 4.1 in [23]). We can thus consider that users are able to solve (22) using a rate adaptation (RA) algorithm, which we formulate as the following assumption.

Assumption 2 (A2). *Denote by $\hat{\lambda}_i(\sigma_i(t))$ the estimated solution to (22). The arrival rate estimation error $\zeta_{i,t} = \hat{\lambda}_i(\sigma_i(t)) - \lambda_i^*(\sigma_i(t))$ satisfies $E[\zeta_{i,t}] = 0$, $t = 1, 2, \dots$ and $E[\zeta_{i,t}^2] \le c_{i,t}$, $\lim_{t\to\infty} c_{i,t} = 0$.*

The assumption that the estimate is unbiased is justified by that $\eta_t \to 0$, which makes that the perturbed reservations converge to $\sigma_i(t)$, and hence the computed arrival rates converge to the actual optimal arrival rate. We now turn to the analysis of the task arrival rate and the utility under the following assumption.

Assumption 3 (A3). *Consider two strategies (λ_i, σ_i) and (λ_i', σ_i'), and let $0 \le \theta \le 1$. Then*

$$\bar{S}_i(\theta\lambda_i + (1-\theta)\lambda_i', \lambda_{-i}, \theta\sigma_i + (1-\theta)\sigma_i', \sigma_{-i}) \le \max(\bar{S}_i(\lambda_i, \lambda_{-i}, \sigma_i, \sigma_{-i}), \bar{S}_i(\lambda_i', \lambda_{-i}, \sigma_i', \sigma_{-i})). \tag{23}$$

In what follows we show that under Assumption 3 the maximum task arrival rate of each user is concave in its rate reservation.

Proposition 3. *Let us define the maximum task intensity $\lambda_i(\sigma_i) = \max\{\lambda_i | \bar{S}_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \sigma_i, \boldsymbol{\sigma}_{-i}) \le \overline{T}_i\}$. If Assumption 3 holds then $\lambda_i(\sigma_i)$ is a concave function of σ_i.*

Proof. Recall that by Corollary 1 and Theorem 5 the mean sojourn time $\bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is increasing and pseudoconvex in λ_i, and is nonincreasing in σ_i, respectively. Assumption 3 implies that the mean sojourn time $\bar{S}_i$ is jointly quasiconvex in (λ_i, σ_i). Quasiconvexity implies that each sublevel set $\{(\lambda_i, \sigma_i) | \bar{S}_i(\boldsymbol{\lambda}, \boldsymbol{\sigma}) \le T\}$ is convex. Since $\bar{S}_i$ is quasiconvex nondecreasing in λ_i, convexity of the sublevel set implies that $\lambda_i(\sigma_i)$ is concave in σ_i. □

A consequence of the above result is that the utility is concave in the rate reservation.

Corollary 6. *Let $\tilde{U}_i(\sigma_i) = U_i(\lambda_i(\sigma_i), (\sigma_i, \boldsymbol{\sigma}_{-i}))$. Then $\tilde{U}_i(\sigma_i)$ is concave in σ_i, and there is $L > 0$ such that $\tilde{U}_i(\sigma_i)$ is L-Lipschitz continuous on $\mathcal{S}_i$.*

Proof. Concavity follows from Proposition 3, and the concavity of f_i. L-Lipschitz continuity follows from that $\bar{S}(\lambda_i, \sigma_i, \boldsymbol{\sigma_{-i}})$ is bounded by T_i, and λ_i and σ_i have compact domain, thus $\bar{S}$ is L-Lipschitz. Observe that for any $\overline{T}_i < \infty$, the set $\mathcal{S}_i \subset [0, \infty)$ is compact, and since $\tilde{U}_i$ is concave, it is Lipschitz continuous in the relative interior of its domain ([24], Proposition 2.107).

In addition, f_i is L_i-Lipschitz continuous by assumption, thus U_i is L-Lipschitz continuous for some $L > 0$. □

Our first main result about OARC establishes that if OARC converges then it indeed converges to an equilibrium of the SSRA game.

Theorem 7. *Assume that the sequence $\boldsymbol{\sigma}(t)$ generated by OARC converges to $\boldsymbol{\sigma}^*(t)$. Then $\boldsymbol{\sigma}^*(t)$ is a Nash equilibrium of the SSRA game.*

Before we present the proof, we introduce three technical results related to the update expression and to the measured utility under noisy rate estimates.

Lemma 3. *The update expression in Line 13 of the OARC algorithm can be written as the projected gradient update*

$$\sigma_i(t+1) = \mathcal{P}_i[\sigma_i(t) - \frac{1}{2\alpha_t}\nabla\hat{U}_i(t)], \tag{24}$$

where $\mathcal{P}_i$ is the Euclidean projection on $\mathcal{S}_i$. The projected gradient is equivalent to

$$\begin{aligned} \mathcal{P}_i(\tilde{\sigma}_i(t)) &= \arg\max_{\sigma_i\in\mathcal{S}_i} \langle\tilde{\sigma}_i(t), \sigma_i\rangle - \frac{1}{2}\|\sigma_i\|^2 \quad (25)\\ \tilde{\sigma}_i(t+1) &= \tilde{\sigma}_i(t) + \frac{1}{2\alpha_t}\nabla\hat{U}_i(t), \quad (26)\end{aligned}$$

where the term $h(\sigma_i) = \frac{1}{2}\|\sigma_i\|^2$ is called the penalty function, and $\tilde{\sigma}_i(t) \in \mathbb{R}$ is called the aggregated gradient.

Proof. The first statement follows from Lemma 1 in [25]. The second statement follows from (3.7) in [26]. □

Second, we characterize the bias of the gradient estimates used in OARC.

Lemma 4. *Consider the measured central difference derivative estimate $\nabla\hat{U}_i(t)$. The estimate has a bias of*

$$\nabla\hat{U}_i(t) - \nabla\tilde{U}_i(t) = \mathcal{O}(\frac{\eta_t^2}{4}) + \frac{\theta(\zeta_{i,t})}{\eta_t}, \tag{27}$$

where $\theta(\zeta_{i,t})$ is the error due to the arrival rate estimation error.

Proof. Consider the Taylor expansion of $\tilde{U}_i$ at $\sigma_i(t)$,

$$\begin{aligned}\tilde{U}_i(\sigma_i(t) \pm \frac{\eta_t}{2}) &= \tilde{U}_i(\sigma_i(t)) \pm \frac{\eta_t}{2}\frac{\partial\tilde{U}_i(\sigma_i(t))}{\partial\sigma_i(t)}\\ &+ \frac{\eta_t^2}{8}\frac{\partial^2\tilde{U}_i(\sigma_i(t))}{\partial\sigma_i^2(t)} \pm \mathcal{O}(\frac{\eta_t^3}{8}), \quad (28)\end{aligned}$$

and use it to express the true gradient at $\sigma_i(t)$ as a function of the central difference derivative estimate,

$$\nabla\tilde{U}_i(t) = \frac{\tilde{U}_i(\sigma_i(t)+\frac{\eta_t}{2}) - \tilde{U}_i(\sigma_i(t)-\frac{\eta_t}{2})}{\eta_t} - \mathcal{O}(\frac{\eta_t^2}{4}). \tag{29}$$

Consider now the measured utility based on (9),

$$\begin{aligned}\hat{U}_i^+(t) &= f_i(\lambda_i(\sigma_i(t)+\frac{\eta_t}{2}) + \zeta_{i,t}) \quad (30)\\ &-c_i^\lambda\lambda_i(\sigma_i(t)+\frac{\eta_t}{2}) - c_i^\lambda\zeta_{i,t} - c_i^\sigma(\sigma_i(t)+\frac{\eta_t}{2}),\\ \hat{U}_i^-(t) &= f_i(\lambda_i(\sigma_i(t)-\frac{\eta_t}{2}) + \zeta_{i,t}) \quad (31)\\ &-c_i^\lambda\lambda_i(\sigma_i-\frac{\eta_t}{2}) - c_i^\lambda\zeta_{i,t} - c_i^\sigma(\sigma_i(t)-\frac{\eta_t}{2}),\end{aligned}$$

where $\lambda_i(\sigma_i(t))$ is the arrival intensity at $\sigma_i(t)$. We can perform a Taylor series expansion of (30) and (31) at $\lambda_i(\sigma_i(t)+\frac{\eta_t}{2})$, and $\lambda_i(\sigma_i(t)-\frac{\eta_t}{2})$ respectively, to obtain

$$\begin{aligned}\hat{U}_i^+(t) &= \tilde{U}_i(\sigma_i(t)+\frac{\eta_t}{2}) +\\ &+ \zeta_{i,t}(f_i'(\lambda_i(\sigma_i(t)+\frac{\eta_t}{2})) - c_i^\lambda) +\\ &+ \frac{\zeta_{i,t}^2}{2}f_i''(\lambda_i(\sigma_i(t)+\frac{\eta_t}{2})) + \mathcal{O}(\zeta_{i,t}^3),\\ &= \tilde{U}_i(\sigma_i(t)+\frac{\eta_t}{2}) + \theta^+(\zeta_{i,t}) \quad (32)\\ \hat{U}_i^-(t) &= \tilde{U}_i(\sigma_i(t)-\frac{\eta_t}{2}) +\\ &+ \zeta_{i,t}(f_i'(\lambda_i(\sigma_i(t)-\frac{\eta_t}{2})) - c_i^\lambda) +\\ &+ \frac{\zeta_{i,t}^2}{2}f_i''(\lambda_i(\sigma_i(t)-\frac{\eta_t}{2})) + \mathcal{O}(\zeta_{i,t}^3)\\ &= \tilde{U}_i(\sigma_i(t)-\frac{\eta_t}{2}) + \theta^-(\zeta_{i,t}), \quad (33)\end{aligned}$$

where $\theta^+(\zeta_{i,t})$ and $\theta^-(\zeta_{i,t})$ are the utility estimation error due to the arrival rate estimation error. Let us subtract (33) from (32) and divide it by η_t, we then obtain

$$\nabla\hat{U}_i(t) = \frac{\tilde{U}_i(\sigma_i(t)+\frac{\eta_t}{2}) - \tilde{U}_i(\sigma_i(t)-\frac{\eta_t}{2})}{\eta_t} + \frac{\theta^+(\zeta_{i,t}) - \theta^-(\zeta_{i,t})}{\eta_t}, \tag{34}$$

which together with (29) and using $\theta(\zeta_{i,t}) = \theta^+(\zeta_{i,t}) - \theta^-(\zeta_{i,t})$ concludes the proof. □

We note that the above result may be extended to non-differentiable functions following the analysis in [27] Third, we show that the utility estimation error due to the arrival rate estimate vanishes.

Lemma 5. *Assume that f_i is smooth and Assumption 2 holds. Then*

$$\lim_{t\to\infty} E\Big[\frac{\theta(\zeta_{i,t})}{\eta_t}\Big] \to 0. \tag{35}$$

Proof. Recall that $\theta(\zeta_{i,t}) = \theta^+(\zeta_{i,t}) - \theta^-(\zeta_{i,t})$, and consider the Taylor series expansion, similar to (32) and (33),

$$\begin{aligned}\frac{\theta(\zeta_{i,t})}{\eta_t} &= \frac{\zeta_{i,t}(f'(\lambda_i(\sigma_i(t)+\frac{\eta_t}{2}) - f'(\lambda_i(\sigma_i(t)-\frac{\eta_t}{2}))}{\eta_t}\\ &+\frac{\frac{\zeta_{i,t}^2}{2}(f''(\lambda_i(\sigma_i(t)+\frac{\eta_t}{2}) - f''(\lambda_i(\sigma_i(t)-\frac{\eta_t}{2}))}{\eta_t} + \dots \quad (36)\end{aligned}$$

Consider now (36) and the limit of its expectation, recalling that the denominator is deterministic, the difference of the first order derivatives in the first term of (36) is equal to the second order derivative by definition. Following

the same logic, the difference in the second term is equal to the third order derivative by definition. This holds for all higher order derivatives in (36) as $\eta_t \to 0$. Now, by assumption f_i is a smooth and L-Lipschitz continuous function, hence its derivatives are bounded. Furthermore, by Assumption 2 we have $E[\zeta_{i,t}^2] \to 0$ as $t \to \infty$, hence higher moments do so too with probability 1, which concludes the proof.

□

Using the above results we are now ready to prove Theorem 7.

Proof of Theorem 7. Let $g^* = g(\boldsymbol{\sigma}^*) = \nabla \tilde{U}(\boldsymbol{\sigma}^*)$ and assume that $\boldsymbol{\sigma}^*$ is not a Nash equilibrium. By the characterization of Nash equilibria (see [26] for details), there exists a player $i \in \mathcal{N}$ and a deviation $q_i \in [\underline{\sigma}_i, \bar{\sigma}_i] = \mathcal{S}_i \subseteq \mathbb{R}$ and $\langle g_i^*, q_i - \sigma_i^* \rangle > 0$. By continuity, there exist some $c > 0$ and neighborhoods U and G of $\boldsymbol{\sigma}^*$ and g^* respectively such that

$$\langle g_i', q_i - \sigma_i' \rangle \geq c \tag{37}$$

whenever $\boldsymbol{\sigma}' \in U$ and $g' \in G$. Now, let Ω be the event that $\boldsymbol{\sigma}(t)$ converges to $\boldsymbol{\sigma}^*$, so $\mathbb{P}(\Omega) > 0$ by assumption. Within Ω we can also assume for simplicity that $\boldsymbol{\sigma}(t) \in U$ and $g(\boldsymbol{\sigma}(t)) \in G$ for all t. Recall that in OARC the learning rate α_t satisfies

$$\sum_{t=1}^{\infty} \Big(\frac{1}{\alpha_t \tau_t}\Big)^2 < \sum_{t=1}^{\infty} \frac{1}{\alpha_t} = \infty, \tag{38}$$

where $\tau_t = \sum_{t'=1}^{t} \frac{1}{\alpha_{t'}}$. By using the update rule given in Lemma 3, and Assumption 2, we can rewrite the update rule in terms of the bias and the error term

$$\begin{aligned}
\tilde{\sigma}_i(t) &= \tilde{\sigma}_i(1) + \sum_{t'=1}^{t} \frac{1}{\alpha_{t'}} \nabla \hat{U}_i(t') \\
&= \tilde{\sigma}_i(1) + \sum_{t'=1}^{t} \frac{1}{\alpha_{t'}} \Big(g_i(t') + \mathcal{O}\Big(\frac{\eta_{t'}^2}{4}\Big) + \frac{\theta(\zeta_{t'})}{\eta_{t'}} \Big) \\
&= \tilde{\sigma}_i(1) + \tau_t \bar{g}_i(t),
\end{aligned} \tag{39}$$

where $\bar{g}_i(t) = \tau_t^{-1} \sum_{t'=1}^{t} \frac{1}{\alpha_{t'}} \Big(g_i(t') + \mathcal{O}\Big(\frac{\eta_{t'}^2}{4}\Big) + \frac{\theta(\zeta_{i,t'})}{\eta_{t'}} \Big)$. By Lemma 5, the term due to the arrival intensity estimation error satisfies $\tau_t^{-1} \sum_{t'=1}^{t} \frac{1}{\alpha_{t'}} \frac{\theta(\zeta_{i,t'})}{\eta_{t'}} \to 0 \, (a.s)$. Let us define some positive constant $M > 0$, we can then rewrite the latter term as

$$\begin{aligned}
\tau_t^{-1} \sum_{t'=1}^{t} \frac{1}{\alpha_{t'}} \mathcal{O}\Big(\frac{\eta_{t'}^2}{4}\Big) &= \tau_t^{-1} \sum_{t'=1}^{t} \frac{1}{(t')^{\gamma_1}} \mathcal{O}\Big(\frac{1}{4(t')^{\gamma_2}}\Big) \\
&\leq \tau_t^{-1} \sum_{t'=1}^{t} \frac{1}{(t')^{\gamma_1}} \frac{1}{4(t')^{\gamma_2}} M \\
&\leq \frac{\sum_{t'=1}^{t} \frac{1}{(t')^{\gamma_1}} \frac{1}{4(t')^{\gamma_2}}}{\sum_{t'=1}^{t} \frac{1}{(t')^{\gamma_1}}} M \to 0 \, (a.s.).
\end{aligned} \tag{40}$$

Consequently, $g(\boldsymbol{\sigma}(t)) \to g^*$ in Ω and $\mathbb{P}(\Omega) > 0$, and hence by (40) we can conclude that $\mathbb{P}(\bar{g}(t) \to g^* | \Omega) = 1$. Consider now the penalty function h defined in Lemma 3, and define its subdifferential

$$\partial h(x) = \{ y \in \mathbb{R} : h(x') \geq h(x) + \langle y, x' - x \rangle, \, \forall x' \in \mathbb{R} \}. \tag{41}$$

Function h is called subdifferentiable at $x \in \mathbb{R}$ whenever $\partial h(x)$ is nonempty, and by (Theorem 12.60(b) in [28], and theorem 23.5 in [29]) for the subdifferential ∂h it holds that $\tilde{\sigma}_i(t) \in \partial h(\sigma_i(t)) \iff \sigma_i(t) = \mathcal{P}_i(\tilde{\sigma}_i(t))$. Thus using the definition of the subdifferential and (39) we have

$$\begin{aligned}
h(q_i) - h(\sigma_i(t)) &\geq \langle \tilde{\sigma}_i(t), q_i - \sigma_i(t) \rangle \\
&\geq \langle \tilde{\sigma}_i(1), q_i - \sigma_i(t) \rangle \\
&\quad + \tau_t \langle \bar{g}_i(t), q_i - \sigma_i(t) \rangle.
\end{aligned} \tag{42}$$

Since $\bar{g}(t) \to g^*$ almost surely on Ω, (37) yields $\langle \bar{g}_i(t), q_i - \sigma_i(t) \rangle \geq c > 0$ for all sufficiently large t. We find that $|\langle \tilde{\sigma}_i(1), q_i - \sigma_i(t) \rangle| \leq \|\tilde{\sigma}_i(1)\|_* \|q_i - \sigma_i(t)\| \leq \|\tilde{\sigma}_i(1)\|_* \|\mathcal{S}_i\| = \mathcal{O}(1)$. By substituting this into (42), we obtain $h(q_i) - h(\sigma_i(t)) > c\tau_t \to \infty$ with positive probability. This is a contradiction since h is continuous and 1-strongly convex, and $\mathcal{S}_i$ is compact. Thus we conclude that $\boldsymbol{\sigma}^*(t)$ is a NE, which proves the result. □

We have so far shown that if OARC converges then it converges to an equilibrim of the SSRA game. In what follows we also show that OARC achieves zero regret asymptotically. For simplicity we present the proof for the case of noiseless rate estimates, but the proof can be easily extended to noisy rate estimates for the expected regret.

Proposition 4. *Let* $\overline{U}_i(\sigma_i(t)) = \frac{1}{2}(\tilde{U}_i(\sigma_i^+(t)) + \tilde{U}_i(\sigma_i^-(t)))$*, and let* $\alpha_t = \sqrt{t}$*. Also, let* $\|\mathcal{S}_i\|^2 = \overline{\sigma}_i - \underline{\sigma}_i$*. If every WD* i *can find the minimizer of* $[S_i(\lambda_i, \boldsymbol{\lambda}_{-i}, \boldsymbol{\sigma}) - \overline{T}_i]^2$ *then the regret of the OARC algorithm is*

$$\begin{aligned}
R_i(T) &= \sum_{t=1}^{T} \tilde{U}_i(\sigma_i^{opt}) - \overline{U}_i(\sigma_i(t)) &(43)\\
&\leq \|\mathcal{S}_i\|^2 \sqrt{T} + \Big(\frac{\|L\|^2}{4} + L \Big) (2\sqrt{T} - 1) &(44)
\end{aligned}$$

Thus, $\limsup_{T \to \infty} R_i(T)/T = 0$.

Proof. Since $\tilde{U}_i$ is concave and L-Lipschitz, for any $\sigma_i(t)$ we have

$$\tilde{U}_i(\sigma_i) \leq \overline{U}_i(\sigma_i(t)) + \nabla \overline{U}_i(t)(\sigma_i - \sigma_i(t)) + L\eta_t \tag{45}$$

for any σ_i, including for σ_i^{opt}. Thus,

$$\tilde{U}_i(\sigma_i^{opt}) - \overline{U}_i(\sigma_i(t)) \leq \nabla \overline{U}_i(t)(\sigma_i^{opt} - \sigma_i(t)) + L\eta_t. \tag{46}$$

At the same time we can use the update equation and Lemma 3 for obtaining the bound

$$\begin{aligned}
&(\sigma_i(t+1) - \sigma_i(t))^2 \leq (\sigma_i(t) - \sigma_i^{opt})^2 \\
&- \tfrac{1}{\alpha_t} (\sigma_i(t) - \sigma_i^{opt}) \nabla \overline{U}_i(t) + \tfrac{1}{4\alpha_t^2} \|\nabla \overline{U}_i(t)\|^2,
\end{aligned} \tag{47}$$

where the inequality is due to the projection $\mathcal{P}_i$. Rearranging the inequality we obtain

$$(\sigma_i(t)-\sigma_i^{opt})\nabla\overline{U}_i(t) \leq \alpha_t((\sigma_i(t)-\sigma_i^{opt})^2 - (\sigma_i(t+1)-\sigma_i^{opt})^2) + \frac{1}{4\alpha_t^2}\|\nabla\overline{U}_i(t)\|^2. \quad (48)$$

We can combine (46) and (48) to obtain

$$\begin{aligned} R_i(T) &\leq \sum_{t=1}^{T}\{(\sigma_i^{opt}-\sigma_i(t))\nabla\overline{U}_t + L\eta_t\} && (49)\\ &\leq \sum_{t=1}^{T}\alpha_t((\sigma_i(t)-\sigma_i^{opt})^2-(\sigma_i(t+1)-\sigma_i^{opt})^2) \\ &\quad + \sum_{t=1}^{T}\{\frac{1}{4\alpha_t}\|\nabla\overline{U}_t\|^2 + L\eta_t\} && (50)\\ &\leq \alpha_1(\sigma_i(1)-\sigma_i^{opt})^2 - \alpha_T(\sigma_i(T+1)-\sigma_i^{opt})^2 \\ &\quad + \sum_{t=2}^{T}(\alpha_t-\alpha_{t-1})(\sigma_i(t)-\sigma_i^{opt})^2 \\ &\quad + \|L\|^2\sum_{t=1}^{T}\frac{1}{4\alpha_t} + L\sum_{t=1}^{T}\eta_t && (51)\\ &\leq \|\mathcal{S}_i\|^2\left(\alpha_1 + \sum_{t=2}^{T}(\alpha_t-\alpha_{t-1})\right) \\ &\quad + \|L\|^2\sum_{t=1}^{T}\frac{1}{4\alpha_t} + L\sum_{t=1}^{T}\eta_t && (52)\\ &\leq \|\mathcal{S}_i\|^2\alpha_T + \|L\|^2\sum_{t=1}^{T}\frac{1}{4\alpha_t} + L\sum_{t=1}^{T}\eta_t. && (53) \end{aligned}$$

Using $\alpha_t = t^{\gamma_1}$, $\eta_t = \frac{1}{t^{\gamma_2}}$, and the bound $\sum_{t=1}^{T} t^{-\gamma} \leq 1 + \int_1^T t^{-\gamma}dt$, we obtain

$$R_i(T) \leq \|\mathcal{S}_i\|^2 T^{\gamma_1} + \frac{\|L\|^2}{4}\frac{T^{1-\gamma_1}-\gamma_1}{1-\gamma_1} + L\frac{T^{1-\gamma_2}-\gamma_2}{1-\gamma_2}. \quad (54)$$

For $0 < \gamma_1, \gamma_2 \leq 1$ we obtain $\limsup_{T\to\infty} R_i(T)/T = 0$. Furthermore, using $\gamma_1 = \gamma_2 = 0.5$ we obtain (44), which proves the result. □

Thus, the OARC algorithm can compute a solution that is asymptotically optimal in hindsight.

VI. Numerical Results

We performed extensive simulations in order to assess equilibrium behavior and to validate the proposed OARC algorithm. For the evaluation we consider three scenarios with different task graphs and queue types. In Scenario 1 the task graph consists of two subtasks in series corresponding to a wireless transmission subtask followed by one computational subtask executed in series. Scenario 2 consists of three subtasks in series, corresponding to wireless transmission subtask followed by two computational subtasks executed in series. Scenario 3 is a fork-join queuing system in which a wireless transmission subtask is followed by two computational subtasks executed in parallel, followed by a computational subtask. For all of the scenarios, we have $|\mathcal{A}| = 4$ APs and $|\mathcal{C}|$=8 servers. We assigned up to $\lceil|\mathcal{N}|/|\mathcal{A}|\rceil$ users at random to each AP.

We set the WDs' latency constraints $\overline{T}_i$ uniform at random on $[0.1, 0.01]$ s, which is reasonable for a variety of low latency applications envisioned for 5G systems [30]. We choose the service rate of each resource and subtask $\mu_{r,v}$ to be uniformly distributed on $[\frac{2}{\overline{T}_i}, \frac{3}{\overline{T}_i}]$ for Scenario 1. For Scenario 2 and Scenario 3 we set the service rate to be 50% higher, on average. Finally, as an example of a non-negative concave function we use $f_i(\lambda_i) = \log(1+\lambda_i)$ for computing the WD's utility [31], and set $c_\lambda = c_\sigma = 0.02$. Note that with these parameters $\overline{\lambda}_i = 49$, and we set $\underline{\lambda}_i = 0$. For the evaluation we consider Poisson arrival processes, the service times are exponentially distributed (M) or deterministic (D), allowing us to validate our results under significantly different service processes.

We used two algorithms as baselines for comparison. The first algorithm is the OCO proposed in [32]. OCO is an extension of the Zinkevich algorithm, meant to satisfy convex stochastic constraints, and maximizes the expected utility by adjusting (λ_i, σ_i) simultaneously. We used perturbations to estimate the local gradients, as those are assumed to be known by OCO. The second baseline is obtained by applying *Online Adaptive Rate Reservation and Control - Sum of Utilization* (OARC-SUM) using the sum utility of all users as objective function, i.e., considering that users cooperate for maximizing their sum utility instead of competing. We refer to this baseline as the OARC-SUM algorithm. In addition, to be able to assess the impact of *Stochastic Approximation* (SA) on the performance of OARC, we consider a baseline for Scenario 1 where we compute the optimal arrival rates λ_i analytically instead of using SA. We refer to this as OARC-Model. The results shown are the averages and the 95% confidence intervals computed based on 30 simulations.

A. Utility Performance

Fig. 3 shows the total utility as a function of the number of WDs for Scenario 1 with exponential service times, for OARC, OCO, OARC-SUM and OARC-Model. Surprisingly, the total utility for OARC is not monotonically increasing. The reason for this is that above $N = 4$ the WDs can no longer achieve their maximum rate $\overline{\lambda}_i$ and thus they contend for the communication and computing resources. Contention in turn decreases the maximum service capacity of the system due to the latency constraints (c.f., the achievable rate in an M/M/1 queue with service rate μ under latency constraint T, vs the sum of the achievable rates in two M/M/1 queues with service rate $\mu/2$ under latency constraint T).

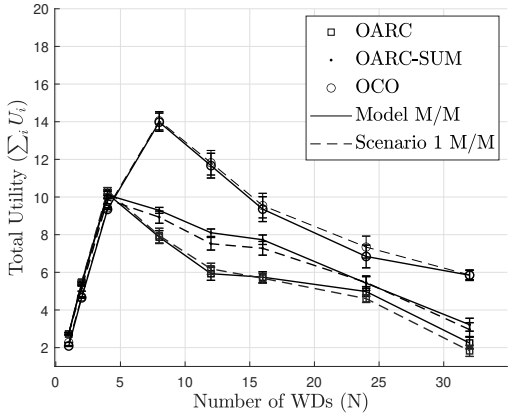

Fig. 3: Utility vs. number of WDs for Scenario 1.

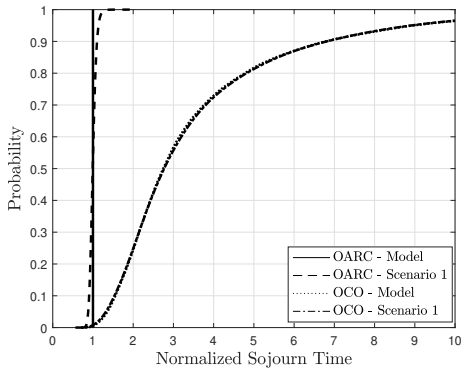

Fig. 4: CDF of normalized sojourn times for $N = 12$ in Scenario 1.

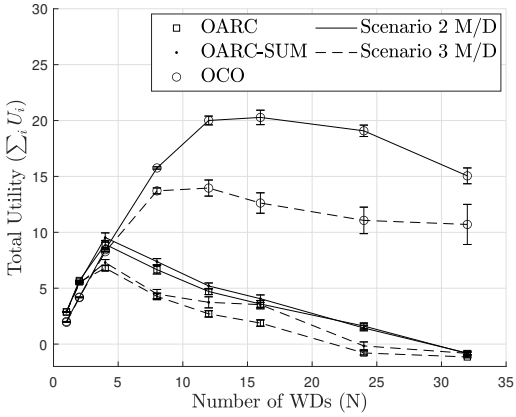

Fig. 5: Utility vs. number of WDs for Scenarios 2 and 3, M/D queue.

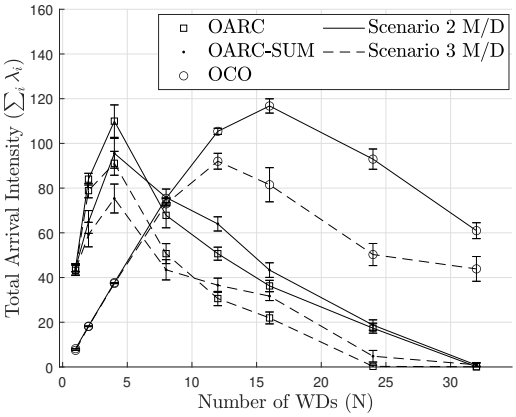

Fig. 6: Arrival intensity vs. number of WDs for Scenarios 2 and 3, M/D queue.

The figure also shows that OARC-SUM outperforms OARC which is justified by that OARC-SUM aims at maximizing the sum utility of all WDs, i.e., WDs do not act independently. The figure also allows us to assess the effect of rate adaptation on the utility obtained by OARC. Comparing the curves for OARC and OARC-Model, we can observe that the impact of stochastic rate adaptation is negligible.

Comparing the results for OARC and OCO, it may be surprising that OCO achieves higher utility than OARC for $N > 4$. To explain why this is possible, Fig. 4 shows the empirical CDF of the normalized sojourn times of the WDs for the two algorithms for $N = 12$. We compute the normalized sojourn time as the ratio of the average sojourn time of a WD divided by its latency constraint. The figure shows that OCO leads to a significant violation of the latency constraint for the majority of WDs. On the contrary, OARC-Model does not lead to a violation of the latency bound, while OARC leads to minor violations of the latency constraint due to SGD-based rate adaptation. Another observation that can be drawn from Fig. 4 is that in the heavy traffic regime OARC enables WDs to adjust their rates and prevents latency violations with high probability. On the contrary, OCO fails to keep the mean sojourn time of the WDs under their latency constraints: when there are many WDs, OCO might lead to unstable queues whereas OARC ensures queue stability by keeping the mean sojourn time of the WDs at their latency constraints. We can thus conclude that OCO does not solve the SSRA problem, mainly due to that the utility is not jointly convex in the arrival rate and in the rate reservation, which highlights the importance of the approach followed by OARC.

Corresponding results for deterministic service times, included in the Appendix, show that the utility for deterministic service times is slightly higher than for exponential service times, but the curves show similar characteristics. In what follows we will show results for deterministic service times for clarity of exposition.

Fig. 5 shows the total utility as a function of the number of WDs for Scenario 2 and Scenario 3 with deterministic service times. The results show that OARC performs close to OARC-SUM for more complex subtasks graphs as well, including a fork-join task graph (Scenario 3). Importantly, it also shows that the shapes of the curves are not affected by the subtask graph topology, i.e., the

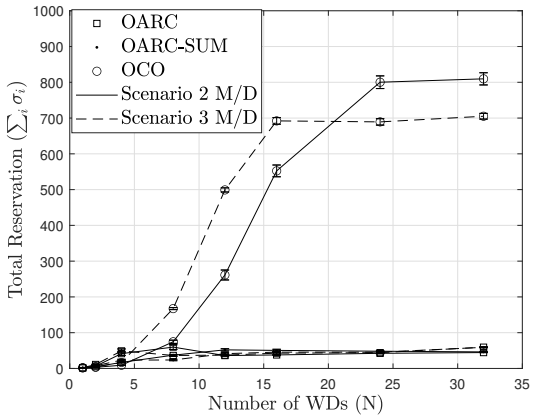

Fig. 7: Reservation vs. number of WDs for Scenarios 2 and 3, M/D queue.

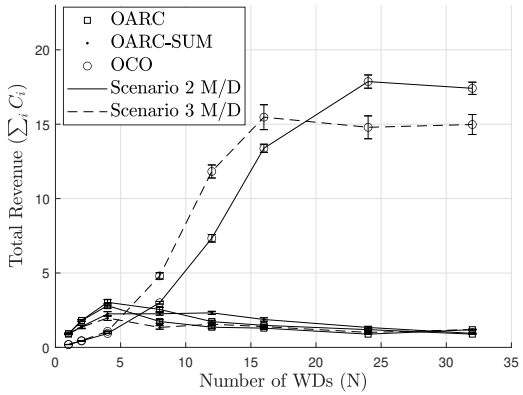

Fig. 8: Revenue vs. number of WDs for Scenarios 2 and 3, M/D queue.

utility decreases due to contention for resources. The superior performance of OCO in Scenario 2 and Scenario 3 is again due to that OCO results in significant latency constraint violations (we omit the figure for brevity).

Fig. 6 shows the total arrival intensity as a function of the number of WDs for Scenario 2 and Scenario 3 with deterministic service times. The results show that the utility is to a large extend determined by the arrival intensity, both for OARC and for OCO. It is interesting to note that OARC-SUM has lower total arrival intensity (particularly for $N < 4$) even though it has higher total utility compared to OARC. This is due to that OARC-SUM prevents that a few users achieve a very high arrival intensity, harming the rest of the users. We also note that the total utility and arrival rate are far from the social optimum for $N > 4$, as the utility obtained for $N = 4$ would be achievable for $N > 4$ by assigning zero rate to all but 4 users, this is, however, not an equilibrium.

B. Operator Revenue

Fig. 7 shows the total reservation as a function of the number of WDs for Scenario 2 and Scenario 3 with deterministic service times. Surprisingly, the total reservation for OARC does not increase linearly with the number of users beyond $N > 4$, which can be explained by that WDs learn that they cannot increase their utility by increasing their reservation parameter due to the congestion on the resources. Interestingly, OCO results in significantly higher resource reservations compared to OARC and OARC-SUM, which is due to that the latency constraint is not met by the WDs, allowing significantly higher rates.

Fig. 8 shows the total revenue of the edge cloud operator as a function of the number of WDs for Scenario 2 and Scenario 3 with deterministic service times. Since the revenue is a linear function of the reservation parameter and the arrival intensity, its shape is similar to that of the curves shown in Figs. 7 and 6. Somewhat surprisingly, the results in Fig. 8 show that the total revenue decreases beyond $N > 4$ when using OARC and OARC-SUM, i.e, the edge cloud operator looses revenue due that the WDs contend for the resources, and consequently reduce their arrival rates so as to meet their latency constraints. This observation leads us to conclude that operators would need to implement admission control to maximize their revenue in a serverless computing environment with latency constrained tasks.

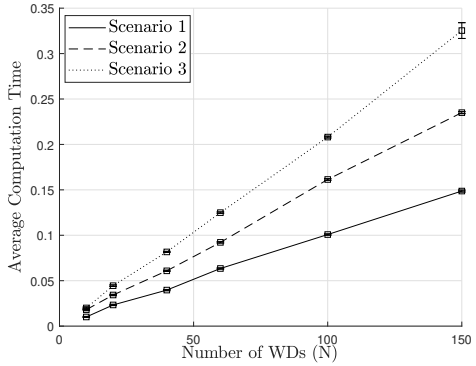

Fig. 9: Average computation time vs. number of rounds for solving Problem (3)-(8).

C. Computation Time of Problem (3)-(8)

Fig. 9 shows the average computation time for solving problem (3)-(8) for all scenarios, based on a Python implementation executed on an Intel i9-10900 CPU. Recall that the task graphs in Scenario 1, Scenario 2 and Scenario 3 contain $2, 3$ and 4 subtasks for each user, respectively, which is why the computation time is highest for Scenario 3. Overall, we observe that the computation time increases approximately linearly as the number of WDs increases. This is because as the number of WDs (N) increases, so does the number of subtasks $|\mathcal{V}| = |\cup_{i\in\mathcal{N}} \mathcal{V}_i|$, indicating that the average complexity of the problem (3)-(8) is linear in the number of subtasks.

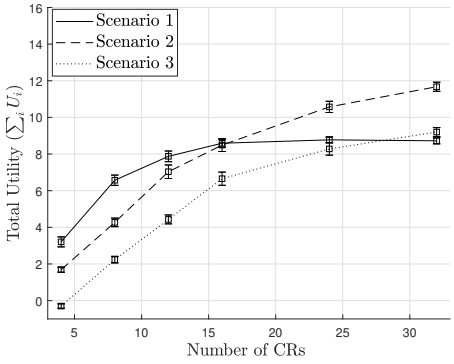

Fig. 10: Total utility vs. number of computing resources for Scenario 1, 2, and 3 with M/M queue where $|\mathcal{A}| = 4$ APs, OARC.

D. Sensitivity Analysis

Fig. 10 shows the total utility as a function of the number of computing resources for Scenarios 1, 2 and 3 with exponential service times for $|\mathcal{A}| = 4$ APs. The figure shows that the utility is a monotonically increasing concave function with respect to the number of computing resources for all scenarios, and indicates that the proposed algorithm utilize the available computing resources. We note that the concavity of the curves is due to the concavity of the utility functions.

VII. Related work

Our problem is related to network utility maximization introduced in [33], later extended to, e.g., packet losses [34], and to queuing networks subject to a stability constraint [35]. Unlike in the case of network utility maximization, in the problem we consider the objective of the network is not aligned with that of the users, which makes the two problems fundamentally different.

Related to ours are recent works on rate control in queuing networks. In [36] authors considered distributed rate control for a fork-join processing network under a static server assignment, and proposed a solution akin to the back-pressure algorithm. The focus of this work was on rate stability, and thus the issue of utilities and latency constraints was not considered. Authors in [37] analyze the convexity of the system time in queuing networks, and authors in [38] consider constrained stochastic approximation and provide unbiased estimators that can be used for GI/G/1 queues. The results hold as long as the cost function is strictly unimodal, including convex.

There are few works focusing on resource management for serverless computing [5]. Authors in [5] use Bayesian optimization for learning the execution time and cost of serverless functions on Amazon AWS. Their approach does not consider server side resource allocation and the interaction among users explicitly, and the solution requires the repeated solution of an integer linear program based on estimated parameters for choosing parameters for service chains.

Our work is related to recent work on online learning. Closely related to our algorithm is the Zinkevich algorithm for unconstrained online convex optimization [32]. The algorithm was extended in [32] to online convex optimization with stochastic constraints. These works focus on a single decision maker, and assume that the cost and the constraint functions are revealed after every round. Similarly, authors in [39], [40] propose algorithms for nested stochastic approximation, but the problem formulations do not consider stochastic constraints.

In the area of computation offloading, authors in [41] propose an offline policy for a dynamic computation offloading and resource scheduling problem under task completion constraints, consider that both wireless devices and the network operator are decision makers, and assume that the task of each device can be modeled as a DAG with the same number of subtasks. Authors in [42] model an application as a directed acyclic data flow graph, consider a system with limited wireless and abundant computing resources shared by multiple applications, and address the problem of deciding which components in the data flow graph should be offloaded onto the cloud such that the throughput of the applications is maximized. Authors in [43] model a computational task as a DAG, consider the congestion on computing resources only, and propose a heuristic for solving an offline task placement problem in which the objective is to minimize the sum cost of the devices under constraints on the dependency among subtasks, the task completion time deadlines and the amount of available computing resources. Finally, authors in [44] consider a task graph with loops, cycles and branches, under the assumption of deterministic service and waiting times. They present heuristic algorithms for solving two related optimization problems, minimizing the response time under a budget constraint, and minimizing the cost under a response time constraint.

These works do not consider, however, the interaction between application rate control, server side resource management and the stochastic service processes. To the best of our knowledge, ours is the first work that considers this interaction, analyzes the existence of equilibria and proposes an online optimization algorithm for learning equilibria in a distributed manner.

VIII. Conclusion

In this paper, we proposed a modeling abstraction and a problem formulation for investigating the interaction between latency constrained services and resource management for serverless edge computing. The proposed abstraction is based on a queuing network model of task graph execution and allows the analysis of the interaction between selfish WDs that reserve edge resources and a serverless operator that allocates resources among WDs, formulated as a non-cooperative game. Our analytical

results show that rate reservation plays an essential role for latency sensitive services, at the same time a simple abstraction for rate reservation allows conceptually simple algorithms, like the proposed OARC, to converge to equilibria with good performance. Our numerical results confirm the analytical findings and also reveal that current practice of serverless service rate allocation leads to a loss of service capacity under latency constraints, and to a loss of operator revenue at the same time. Consequently, solutions for admission control complemented with new abstractions and related scheduling policies would be desirable for latency constrained computing tasks in a serverless edge computing infrastructure. Our model could be extended to consider that the computing price is dependent on the total reservation, i.e., increasing with the contention for computing resources, it could be used to study the impact of different forms of signaling between the WDs and the operator on convergence speed and the resulting utility, and it could be extended to consider more complex models of task graphs. We leave these to be subject of our future work.

References

[1] M. Hakkarainen, C. Woodward, and M. Billinghurst, "Augmented assembly using a mobile phone," in *Proc. of IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2008, p. 167–168.

[2] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[3] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2014, p. 68–81.

[4] V. Sreekanti, C. Lin, J. Faleiro, J. Gonzalez, J. Hellerstein, and A. Tumanov, "Cloudburst: Stateful Function-as-a-Service," *Proc. of VLDB Endowment*, vol. 13, no. 11, Aug. 2020.

[5] N. Akhtar, A. Raza, V. Ishakian, and I. Matta, "Cose: Configuring serverless functions using statistical learning," in *Proc. of IEEE INFOCOM*, 2020.

[6] M. S. Aslanpour, A. N. Toosi, C. Cicconetti, B. Javadi, P. Sbarski, D. Taibi, M. Assuncao, S. S. Gill, R. Gaire, and S. Dustdar, "Serverless edge computing: Vision and challenges," in *Australasian Computer Science Week Multiconference*, 2021.

[7] A. Mampage, S. Karunasekera, and R. Buyya, "A holistic view on resource management in serverless computing environments: Taxonomy and future directions," *ACM Comput. Surv.*, 2022, accepted.

[8] S. Redana, Ö. Bulakci, A. Zafeiropoulos, A. Gavras, A. Tzanakaki, A. Albanese, A. Kousaridas, A. Weit, B. Sayadi, B. T. Jou *et al.*, "5G PPP architecture working group: View on 5G architecture." European Commission, 2019.

[9] J. F. C. Kingman, "The single server queue in heavy traffic," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 57, no. 4, p. 902–904, 1961.

[10] L. Li, T. Q. Quek, J. Ren, H. H. Yang, Z. Chen, and Y. Zhang, "An incentive-aware job offloading control framework for multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 63–75, 2021.

[11] L. Li, M. Siew, and T. Q. Quek, "Learning-based pricing for privacy-preserving job offloading in mobile edge computing," in *Pric. of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 4784–4788.

[12] M. Siew, D. Cai, L. Li, and T. Q. S. Quek, "Dynamic pricing for resource-quota sharing in multi-access edge computing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2901–2912, 2020.

[13] W. Fang, X. Yao, X. Zhao, J. Yin, and N. Xiong, "A stochastic control approach to maximize profit on service provisioning for mobile cloudlet platforms," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 522–534, 2018.

[14] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2021.

[15] R. R. Weber, "A note on waiting times in single server queues," *Operations Research*, vol. 31, no. 5, pp. 950–951, 1983.

[16] R. M. Loynes, "The stability of a queue with non-independent inter-arrival and service times," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 58, no. 3, 1962, pp. 497–520.

[17] A. J. Ganesh, "Large deviations of the sojourn time for queues in series," *Annals of Operations Res.*, vol. 79, pp. 3–26, 1998.

[18] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2007.

[19] G. Scutari, D. P. Palomar, F. Facchinei, and J.-S. Pang, "Convex optimization, game theory, and variational inequality theory," *IEEE Signal Proc. Mag.*, vol. 27, no. 3, 2010.

[20] F. Facchinei, A. Fischer, and V. Piccialli, "On generalized nash games and variational inequalities," *Operations Research Letters*, vol. 35, no. 2, pp. 159–164, 2007.

[21] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[23] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *Proc. of Conference on Learning Theory (COLT)*, vol. 49, Jun. 2016, pp. 1246–1257.

[24] J. F. Bonnans and A. Shapiro, *Perturbation analysis of optimization problems*. Springer, 2013.

[25] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. of International Conference on Machine Learning (ICML)*, 2003, p. 928–935.

[26] P. Mertikopoulos and Z. Zhou, "Learning in games with continuous action sets and unknown payoff functions," *Mathematical Programming*, vol. 173, no. 1-2, pp. 465–507, 2019.

[27] Y. He, M. Fu, and S. Marcus, "Convergence of simultaneous perturbation stochastic approximation for nondifferentiable optimization," *IEEE Transactions on Automatic Control*, vol. 48, no. 8, pp. 1459–1463, 2003.

[28] R. T. Rockafellar, *Convex analysis*. Princeton University Press, 1970.

[29] R. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Springer, 1998.

[30] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*. Academic Press, 2018.

[31] R. Hassin, *Rational queueing*. CRC Press, 2016.

[32] H. Yu, M. J. Neely, and X. Wei, "Online convex optimization with stochastic constraints," in *Proc. of NeurIPS*, 2017, p. 1427–1437.

[33] 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.

[34] J.-W. Lee, M. Chiang, and R. Calderbank, "Price-based distributed algorithms for rate-reliability tradeoff in network utility maximization," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 24, no. 5, 2006.

[35] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, pp. 401–457, 2005.

[36] H. Zhao, C. H. Xia, Z. Liu, and D. Towsley, "Distributed resource allocation for synchronous fork and join processing networks," in *Proc. of IEEE INFOCOM*, 2010, pp. 1–5.

[37] J. G. Shanthikumar and D. D. Yao, "Second-order stochastic properties in queueing systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 162–170, 1989.

[38] P. L'Ecuyer and P. W. Glynn, "Stochastic optimization by simulation: Convergence proofs for the GI/G/1 queue in steady-state," *Management Science*, vol. 40, no. 11, pp. 1562–1578, 1994.

[39] S. Ghadimi, A. Ruszczyski, and M. Wang, "A single timescale stochastic approximation method for nested stochastic optimization," *SIAM Journal on Optimization*, vol. 30, no. 1, pp. 960–979, 2020.
[40] M. Wang, E. Fang, and H. Liu, "Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions," *Mathematical Programming*, vol. 161, pp. 419—449, 2017.
[41] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *Proc. of IEEE INFOCOM*, 2016, pp. 1–9.
[42] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 4, pp. 23–32, 2013.
[43] S. Sundar and B. Liang, "Offloading dependent tasks with communication delay and deadline constraint," in *Proc. of IEEE INFOCOM*, 2018, pp. 37–45.
[44] C. Lin and H. Khazaei, "Modeling and optimization of performance and cost of serverless applications," *IEEE Trans. on Parallel and Distributed Systems*, vol. 32, no. 3, pp. 615–632, 2021.
[45] J. M. Harrison, "Brownian models of open processing networks: Canonical representation of workload," *Annals of Applied Probability*, vol. 13, no. 1, pp. 390–393, 2003.

Feridun Tütüncüoğlu is a Ph.D. student at the Division of Network and Systems Engineering in KTH Royal Institute of Technology, Stockholm, Sweden. He received M.Sc in Electrical & Electronics Engineering from Bilkent University, Turkey in 2019. He worked as a research engineer at the department of Electrical & Electronics Engineering, Bilkent University from 2017 to 2019. His research interests include design and analysis of online learning algorithms and game theoretical models of edge computing resource management and allocation.

Slađana Jošilo received the M.Sc. degree in electrical engineering from the University of Novi Sad, Serbia in 2012, and the Ph.D. in electrical engineering from KTH Royal Institute of Technology, Stockholm, Sweden in 2020. She worked as a research engineer with the Department of Power, Electronics and Communication Engineering at the University of Novi Sad in 2013-2014, and as a postdoctoral researcher with the Division of Network and Systems Engineering at KTH in 2020-2021. Currently, she works as a researcher at Ericsson, Stockholm, Sweden. Her research interests include $5G$ networks, edge computing systems, and applied game theory.

György Dán (M'07, SM'17) is a professor at KTH Royal Institute of Technology, Stockholm, Sweden. He received the M.Sc. in computer engineering from the Budapest University of Technology and Economics, Hungary in 1999, the M.Sc. in business administration from the Corvinus University of Budapest, Hungary in 2003, and the Ph.D. in Telecommunications from KTH in 2006. He worked as a consultant in the field of access networks, streaming media and videoconferencing 1999-2001. He was a visiting researcher at the Swedish Institute of Computer Science in 2008, a Fulbright research scholar at University of Illinois at Urbana-Champaign in 2012-2013, and an invited professor at EPFL in 2014-2015. He served as area editor of Computer Communications 2014-2021, and has been editor of IEEE Transactions on Mobile Computing since 2019. His research interests include the design and analysis of content management and computing systems, game theoretical models of networked systems, and cyber-physical system security and resilience.

APPENDIX

A. Starvation without Rate Reservation

In order to show the importance of rate reservation, let us consider the interaction between autonomous rate adjustment and operator load balancing under the hypothetical scenario that the reservation equals the rate ($\sigma_i = |\mathcal{V}_i|\lambda_i$), i.e., rate-driven resource allocation. We start with a simple observation concerning rate stability of the solution to (3)-(8).

Lemma 6. *Let $\sigma_v = \lambda_i, \forall v \in \mathcal{V}$, and $\boldsymbol{p}^* = (p^*_{r,v})_{r\in\mathcal{R}, v\in\mathcal{V}}$ an optimal solution to (3)-(8), $\rho^*_r = \sum_{v\in\mathcal{V}_r} p^*_{r,v}$. Then $\boldsymbol{p}^*$ is rate stable if and only if $\rho^* = \max_r \rho^*_r \leq 1$. It is stable (in the sense of bounded queue length) if and only if $\rho^* < 1$.*

Proof. Let $\sigma_v = \lambda_i$, and observe that at an optimal solution $(\rho^*, \mathbf{p}^*)$

$$\mu^*_v = \sum_{r\in\mathcal{R}_v} \mu_{r,v} p^*_{r,v} \geq \lambda_i, \forall v \in \mathcal{V}, \tag{55}$$

and hence rate stability follows from $\rho^* \leq 1$ [45]. On the contrary, if $\rho^* > 1$ then $\exists r \in \mathcal{R}$ such that $\rho^*_r > 1$, and hence the subtask queue at resource r is not rate stable. For bounded queue length stability, observe that $\rho^* < 1$ implies that $\lambda_i < \sum_{r\in\mathcal{R}_v} \frac{1}{\rho^*_r}\mu_{r,v}p_{r,v}$, while $\rho^* = 1$ implies that there is a $v \in \mathcal{V}$ such that $\rho^*_r = 1\ \forall r \in \mathcal{R}_v$, and thus the queue length is unbounded for subtask v. □

We now turn to the analysis of the interaction between WDs and the operator, and use the following example.

Example 1. Consider a system with WDs $\mathcal{N} = \{1, 2\}$, a single AP ($A = 1$) and a single computing resource ($C = 1$), referred to as resources 1 and 2, respectively. Each subtask graph $\mathcal{G}_i$ consists of a transmission subtask and an execution subtask, i.e., $|\mathcal{V}_i| = 2$, and $\mathcal{V} = \cup\mathcal{V}_i = \{1, 2, 3, 4\}$. Task arrivals follow independent Poisson processes with rates λ_i, service times are exponentially distributed with rates $\mu_{1,i}$ and $\mu_{2,i+2}$, and $\sigma_v = \lambda_i$.

Consider now that WDs and the operator periodically update their rates and the resource allocation vector, respectively. As the following result shows, the resulting rate adjustment leads to starvation.

Lemma 7. *Consider the interaction between strategic WDs and a load balancing operator (eqns. (3)-(8)). The resulting rate adjustment under latency constraints can lead to starvation.*

Proof. Consider Example 1. We can express the maximum arrival rate λ_i of WD i as a function of the actual service rates $\tilde{\mu}_{1,i} = \tilde{p}_{1,i}\mu_{1,i}$ and $\tilde{\mu}_{2,2+i} = \tilde{p}_{2,2+i}\mu_{2,2+i}$ and the latency constraint $\overline{T}_i$ as

$$\lambda_i = \max(0, \frac{\overline{T}_i(\tilde{\mu}_{1,i} + \tilde{\mu}_{2,2+i}) - 2 - D(\overline{T}_i, \tilde{\mu}_{1,i}, \tilde{\mu}_{2,2+i})}{2\overline{T}_i}), \tag{56}$$

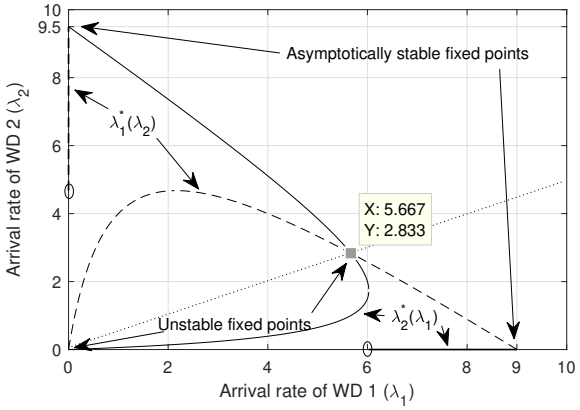

Fig. 11: Correspondences $\lambda^*_1 = \mathcal{L}_1(\lambda^*_1, \lambda_2)$ and $\lambda^*_2 = \mathcal{L}_2(\lambda_1, \lambda^*_2)$ for $\mu = 10$, $\overline{T}_1 = 2$, $\overline{T}_2 = 4$. Intersections are fixed points of $\mathcal{L}$. The stable fixed points are $(0, 9.5)$ and $(9, 0)$. The dotted line separates the basins of attraction of the two stable fixed points.

where

$$D(\overline{T}_i, \mu_1, \mu_2) = \sqrt{\overline{T}_i^2(\mu_1 - \mu_2)^2 + 4}$$

It is easy to see that (56) is a concave increasing function of $\tilde{\mu}_{1,i}$ and $\tilde{\mu}_{2,2+i}$, and since its Hessian with respect to $(\tilde{\mu}_{1,i}, \tilde{\mu}_{2,2+i})$ is negative semi-definite, it is jointly concave in $(\tilde{\mu}_{1,i}, \tilde{\mu}_{2,2+i})$. Furthermore, it is a concave increasing function of $\overline{T}_i$.

Consider now problem (3)-(8), and observe that for the considered example the solution can be expressed as $p_{1,i} = \lambda_i/\mu_{1,i}$ and $p_{2,2+i} = \lambda_i/\mu_{2,2+i}$ for $i \in \{1, 2\}$, and thus $\tilde{p}_{1,i} = \frac{\lambda_i/\mu_{1,i}}{\lambda_1/\mu_{1,1}+\lambda_2/\mu_{1,2}}$, and $\tilde{p}_{2,2+i} = \frac{\lambda_i/\mu_{2,2+i}}{\lambda_1/\mu_{2,3}+\lambda_2/\mu_{2,4}}$. We can substitute these in (56) to obtain the mapping $\mathcal{L} : \mathbb{R}^2 \to \mathbb{R}^2$, whose fixed points correspond to rates (λ_1, λ_2) that could be achieved by the WDs after iteratively updating their rates in response to the operator's optimization of the service rates, so as to meet their latency constraints.

For simplicity, let us consider uniform service rates $\mu_{1,i} = \mu_{2,2+i} = \mu$, and observe that the latency constraint is feasible for WD i if $\overline{T}_i \geq 2/\mu$. After substitution, we obtain the mapping $\mathcal{L}_i(\lambda_1, \lambda_2) = \max(0, \frac{\lambda_i}{\lambda_1+\lambda_2}\mu - \frac{2}{\overline{T}_i})$. Assume now that $\overline{T}_1 < \overline{T}_2$, then $\mathcal{L}$ has four fixed points, $\boldsymbol{\lambda}^{*(0)} = (0, 0)$, $\boldsymbol{\lambda}^{*(1)} = (0, \mu - 2/\overline{T}_2)$, $\boldsymbol{\lambda}^{*(3)} = (\mu - 2/\overline{T}_1, 0)$, and an interior point where $\lambda^*_1 + \lambda^*_2 = \mu - 2/\overline{T}_1 - 2/\overline{T}_2$, i.e., $\boldsymbol{\lambda}^{*(4)} = (\frac{\overline{T}_2\mu}{\overline{T}_1+\overline{T}_2} - \frac{2}{\overline{T}_1}, \frac{\overline{T}_1\mu}{\overline{T}_1+\overline{T}_2} - \frac{2}{\overline{T}_2})$. Nonetheless, the only fixed points that are asymptotically stable are $\boldsymbol{\lambda}^*(1)$ and $\boldsymbol{\lambda}^*(2)$, as illustrated in Figure 11, and at these fixed points only one WD has non-zero rate. □

The above result shows that without rate reservation the interaction between rate control and resource allocation under latency constraints can lead to undesirable outcomes. Hence in the main body of the paper we consider the case of rate reservation.

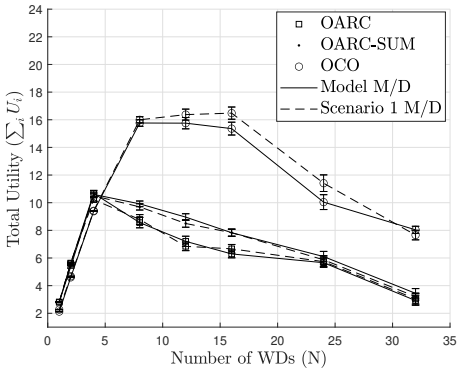

Fig. 12: Utility vs. number of users for Scenario 1, M/D queue.

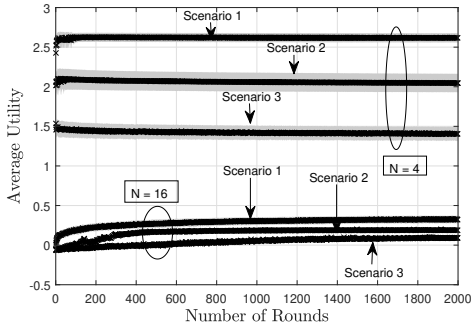

Fig. 13: Average utility vs. number of rounds for OARC

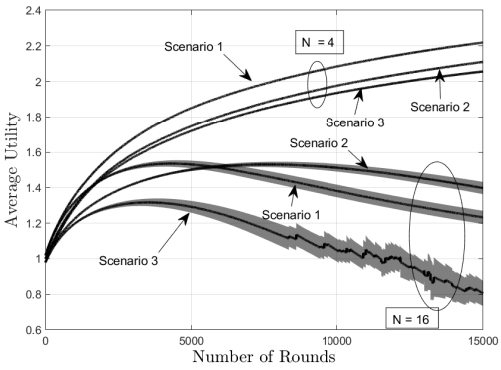

Fig. 14: Average utility vs. number of rounds for OCO

B. Additional Numerical Results

1) Utility Results for Deterministic Service Times: Fig. 12 shows the total utility as a function of the number of WDs for Scenario 1 with deterministic service times, for OARC, OCO, OARC-SUM and OARC-Model. The characteristics of the curves are similar to those obtained using exponential services times, but the total utility is higher. This is justified by that deterministic services times allow a higher rate compared to exponential services times when subject to the same response time constraint.

2) Rate of Convergence: Fig. 13 and Fig. 14 show the average utility per WD as a function of the number of rounds for OARC and for OCO, respectively, for $N = 4$ and $N = 16$. The figure shows that OARC converges relatively fast, within a few hundred rounds, to an equilibrium. We can also conclude that convergence is slower when the number of WDs is higher, which is partly due to the increasing contention for resources. While the rate of convergence is fairly good considering that OARC does not require signaling among WDs, it may be still too slow for practical deployment. Additional signaling may accelerate convergence, and would be an interesting direction of future research. OCO converges much slower (for $N = 4$, notice the different horizontal axes) and it did not converge at all for $N = 16$ in our simulations, highlighting the superior performance of OARC.