D2.3 WP2 Research Report II

Project Name: Anticipatory Networking Techniques in 5G and Beyond
Acronym: ACT5G
Project no.: 643002

Start date of project: 01/05/2015 **Duration:** 48 Months

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions.

Document Properties

Document ID	EU-H2020-MSCA-ITN-2014-643002-ACT5G-D2.3
Document Title	D2.3 WP2 Research Report II
Contractual date of delivery to REA	Month 42
Lead Beneficiary	Linköpings universitet (LiU)
Editor(s)	D. Yuan – LiU
Work Package No.	1
Work Package Title	Reaction Techniques
Nature	Report
Number of Pages	8
Dissemination Level	**PUBLIC**
Contributors	LiU: V. Angelakis, D. Yuan POLIMI: A. Capone, M. Cesana Bell labs: I. Malanchini
Version Number	1

Contents

0 Executive Summary

This is the second research report within work package two Reaction Techniques of the ACT5G project. The document provides information of the conducted and expected work of early-stage researcher (ESR) three and four. The document first gives an overview of the focus area and research topics. Technical details of the work are then presented by means of research paper published or submitted since the previous research report.

1 Work Plan and Progress of ESR 3

Emmanouil Fountoulakis, ESR 3, joined the project in August 15, 2016. His main research activities are the control and performance analysis of 5G networks. More specifically, Emmanouil is working on modelling and optimizing the latency under different network set ups.

His main research direction includes optimization techniques (Lyapunov optimization) for developing dynamic real time algorithms. Furthermore, for analysing the performance of the networks, Emmanouil applies tools from queueing theory for deriving throughput and latency of the networks. Until May 2018, Emmanouil had been focusing on delay sensitive applications using Lyapunov optimization techniques for deriving dynamic algorithms. Under delay sensitive applications concept, he considered packets that are deadline constrained and are stored in a queue before transmission. The objective was the minimization of dropping rate under power consumption constraints. This work led to one conference paper accepted for publication in the IEEE GLOBECOM 2018 that will take place in December 2018.

From May 2018 the ESR 3 has been working on a new topic related to Network Function Virtualization. In order to better understand this new concept of 5G, ESR 3 analyzed a small topology that consists of a set Mobile Edge Computing and Core servers that host different Virtual Network Functions. Theoretical and simulation results are derived by using tools from queueing theory. A conference paper is currently in progress and it is going to be submitted during October. After the submission, the plan is to use the results from the analysis in order to optimize the performance of the system regarding to the end-to-end latency under the NFV concept for general network topologies.

2 Work Plan and Progress of ESR 4

By the time of the previous research report, ESR 4, Özgür Umut Akgül, presented a novel negotiation and trading platform in a multi-tenant network. The proposed resource market allows tenants to negotiate for resources according to their budget limits and utility expectations. The negotiation outcome is translated into sharing parameters in order to enable real-time resource scheduling in the shared infrastructure. Unlike the former studies in the literature, the proposed framework allows tenants to renegotiate their sharing parameters after a predefined time window (i.e. in the order of seconds). This proposed model has been published in ICC 2017 conference. Following this first market definition, the ESR 4 proposed a dynamic network slicing algorithm which is built upon the model in ICC as an extension of the model. The framework has been improved in order to accommodate the envisioned service heterogeneity in 5G. In this model, provided services are differentiated using a piece-wise linear utility function. This paper was presented in the IEEE Globecom 2017 conference.

Spanning the time between the previous report and the current one, ESR 4's work has primarily focused on consolidating advantages of the proposed dynamic network slicing and short time scale trading framework within a large set of simulations. Another level of differentiation among tenants is achieved by allowing tenants to choose their own time window in line with their strategies. More specifically, the renegotiation period is separated from the time windows of individual tenants and the tenants are allowed to differentiate their services via time window differentiation. Moreover, ESR 4 also implemented a simple prediction algorithm in order to enhance the framework's efficiency (both in terms of cost and spectrum). The analysis has proven that the proposed framework provides fairness among both tenants and services and can improve the efficiency of the resource allocation by exploiting simple prediction mechanisms. Despite the tenants share a common infrastructure, results have also demonstrated that it is possible for them to differentiate their services by tuning model parameters. It is also shown that the pricing model can allocate economic resources for capacity expansion and that this is crucial to keep infrastructure sharing convenient for tenants. The outcome of this study has been submitted to IEEE Transactions on Network and Service Management and currently on major review.

The previous research has shown that having a prior knowledge of the upcoming conditions can improve the performance of the slicing algorithm as high as

40%. Thus as the next step of his research, ESR 4 has focused on the anticipatory network slicing and trading, namely, using traffic and channel forecasting in order to improve the efficiency of the real time scheduler. Two candidate prediction methods, i.e. Feed-Forward Neural Networks (FFNN) and Auto Regressive Integrated Moving Average (ARIMA), are compared to be used in the trading framework. As a result of its low time complexity, high prediction accuracy and adaptation skill, ARIMA is used in the ongoing research. The prediction errors have a direct impact on the efficiency of the proposed framework. However, the prediction accuracy increases in parallel to the time complexity of the prediction algorithm. In order to enable the negotiation framework to be run in real time, a simple yet efficient model is required. With this objective in mind, ESR 4 defined a novel filtering approach that can exploit the advantages of prediction while the prediction quality is high and can also filter out the prediction data when the accuracy is very low. The numerical analysis showed that application of filter can decrease the tenants total cost while it also allows the infrastructure provider to serve more tenants compared to no prediction scenario, using the same infrastructure. The outcome of this work has been submitted to IEEE International Conference on Communications (ICC) 2019.

The ESR 4 is expected to complete his PhD studies by Spring 2019. The planned activities till the end of this period are to focus on extending the current work to include self-dimensioning and planning in sliced multi-tenant networks.

3 Appendices:

- Dynamic power control for packets with deadlines (accepted by IEEE Globecom, 2018)
- Dynamic resource trading in sliced Mobile networks (under revision for IEEE Transactions on Network and Service Management)
- Anticipatory resource allocation and trading in a sliced network (submitted to IEEE ICC, 2019)

Dynamic Power Control for Packets with Deadlines

Emmanouil Fountoulakis[†‡], Nikolaos Pappas[†], Qi Liao[‡], Anthony Ephremides [†*], Vangelis Angelakis [†]
[†] Department of Science and Technology, Linköping University, Sweden
[‡] Nokia Bell Labs, Stuttgart, Germany
[*] Electrical and Computer Engineering Department, University of Maryland, College Park
E-mails: {emmanouil.fountoulakis, nikolaos.pappas, vangelis.angelakis}@liu.se, etony@umd.edu
qi.liao@nokia-bell-labs.com

***Abstract*—Wireless devices need to adapt their transmission power according to the fluctuating wireless channel in order to meet constraints of delay sensitive applications. In this paper, we consider delay sensitivity in the form of strict packet deadlines arriving in a transmission queue. Packets missing the deadline while in the queue are dropped from the system. We aim at minimizing the packet drop rate under average power constraints. We utilize tools from Lyapunov optimization to find an approximate solution by selecting power allocation. We evaluate the performance of the proposed algorithm and show that it achieves the same performance in terms of packet drop rate with that of the Earliest Deadline First (EDF) when the available power is sufficient. However, our algorithm outperforms EDF regarding the trade-off between packet drop rate and average power consumption.**

***Index Terms*—Deadline-constrained traffic, power efficient algorithms, Lyapunov optimization, centralized scheduler, dynamic algorithms.**

I. Introduction

In many applications, data packets must be successfully transmitted within a particular time frame, i.e., by some deadline. If a packet is not transmitted before its deadline expiration, then, its information is considered to be useless and the packet is removed from the system [1]. This is the case for a multitude of applications, such as multimedia streaming, online gaming, and the new 5G applications such as autonomous driving that has strict round trip delay constraint. With the pervasiveness of mobile communications, such applications need to perform over wireless devices. In wireless communications, transmission errors occur due to the fluctuating nature of the channel. Assuming perfect channel knowledge at the transmitter, the elimination of errors due to fading can be achieved by increasing the transmission power, for a given transmission rate. However, in many cases, e.g., Internet of Things (IoT), power-limited wireless devices require low average power consumption. Therefore, energy efficiency issues become very important.

Delay constrained network optimization has been extensively investigated and different optimization approaches have been applied to different scenarios, refer to [2] and the references therein. For deadline-constrained scheduling, Earliest Deadline First (EDF) has been shown to be optimal in terms of number of served packets over error free (wired) channels [3]. For the case of wireless fading channels (wireless communications), the authors in [4] propose an optimal scheduling scheme for single transmitter and receiver with energy constraints by using a dynamic algorithm. Similar scenarios have been studied in [5]–[7], where dynamic programming and Markov decision theory are applied. Authors in [8] develop a scheduling scheme that minimizes the number of dropped packets transmitted over fading channels by using dynamic programming. In addition, they assume that the deadlines of the packets satisfy some particular requirements, i.e., the deadlines of subsequent packets depend on each other. Analytical results are provided by the authors in [9] regarding on how the power should be selected in order to approach deadlines. Authors in [10], [11] consider deadline-constrained traffic and decide on the channel or power allocation. In addition, authors in [12] examine the impact of packet deadline on the age of information for queing systems.

In this paper, we develop a dynamic algorithm that finds an approximate solution to the problem of minimizing packet drop rate by optimizing power allocation under average power consumption constraints. The algorithm observes the channel conditions and the remaining deadline of the users' packets and optimizes the power allocation without knowledge of arrival packet statistics. We use Lyapunov drift and Lyapunov optimization theory to develop a dynamic algorithm. The proposed dynamic algorithm decides the power allocation at each time slot by minimizing an upper bound on the drift-plus-penalty expression. We compare the performance of our algorithm with that of EDF. EDF searches across the users the packet with the shortest expiration time and assigns to that user the appropriate power. Numerical and simulation results show that our scheduling scheme achieves the same performance in terms of packet drop rate with that of EDF when the available power is sufficient. Also, our dynamic algorithm is able to satisfy the average power constraint. On the other hand, EDF violates the average power consumption constraints when the available power is not sufficient. In addition, our dynamic algorithm offers a good trade-off between average power consumption and packet drop rate.

II. System Model

We consider N users transmitting packets to a single receiver over wireless fading channels. Let $\mathcal{N} \triangleq \{1, \ldots, N\}$

This work has been supported by the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 643002.

be the set of the users in the system. Time is assumed to be slotted. Let $t \in \mathbb{Z}$ be the t^{th} slot. We consider the users to be synchronized and at most one user can transmit at each time slot. Each user i, where $i \in \mathcal{N}$, is associated with a queue where the packets are held or dropped. Let $Q_i(t)$ be the number of packets in queue i in the t^{th} slot. Each user i generates a packet with a probability π_i at each time slot t. Let $\boldsymbol{\alpha}(t) \triangleq \{\alpha_i(t)\}_{i\in\mathcal{N}}$, where $\alpha_i(t) \in \{0,1\}$, represent the packet arrival process for each user i in the t^{th} slot. The random variables of packet arrival process are independent and identically distributed (i.i.d.). Furthermore, we assume that at most one packet can be transmitted at each time slot and no collisions are allowed.

Each packet that arrives in a queue has a deadline by which it must be transmitted. Otherwise, it is dropped and removed from the system. For simplicity, the deadlines of the packets in the same queue are assumed to be the same. However, deadlines of different queues may vary. We denote the packet deadline of the i^{th} queue with $m_i \in \mathbb{Z}_+$, $\forall i \in \mathcal{N}$. We assume that in each queue, packets are served in the order that they arrive following the First In First Out (FIFO) discipline. Let $d_i(t)$ be the number of slots left in the t^{th} slot before the packet that is at the head of queue i expires.

We assume that the channel state at the beginning of each time slot is known. The channel state remains constant within one slot but it changes from slot to slot. Let $\mathbf{S}(t) \triangleq \{S_i(t)\}_{i\in\mathcal{N}}$ represent the channel state for each user i during slot t. We assume that the channel can be either in "Bad" state (deep fading) or in "Good" state (mild fading). The possible channel states of each user i are described by the set $\mathcal{S} \triangleq \{\mathrm{B},\mathrm{G}\}$, and $S_i(t) \in \mathcal{S}$, $\forall i \in \mathcal{N}$. For simplicity, we assume that the random variables of the channel process $\mathbf{S}(t)$ are i.i.d. from one slot to the next.

Let $\mathbf{p}(t) \triangleq [p_1(t), \ldots, p_N(t)]$ denote the power allocation vector in the t^{th} slot. We consider a set of discrete power levels $\left\{0, P^{(\text{Low})}, P^{(\text{High})}\right\}$. We assume that $P^{(\text{High})}$ is needed for a packet to be successfully transmitted under "Bad" channel condition, and $P^{(\text{Low})}$ under "Good" channel condition. At each time slot, the set of selectable power levels $\mathcal{P}_i(t)$ for each user is conditioned on the channel state $S_i(t)$. For example, if the current channel state is "Bad", then $P^{(\text{Low})}$ cannot be selected. Thus, we have

$$p_i(t) \in \begin{cases} \left\{0, P^{(\text{High})}\right\}, \text{ if } S_i(t) = \mathrm{B} \\ \left\{0, P^{(\text{Low})}\right\}, \text{ if } S_i(t) = \mathrm{G} \end{cases}, \forall i \in \mathcal{N}. \quad (1)$$

Let $\mu_i(t)$ be the power allocation, or packet serving, indicator for the user i in the t^{th} slot, we have

$$\mu_i(t) \triangleq \begin{cases} 1, \text{ if } p_i(t) > 0 \\ 0, \text{ otherwise} \end{cases}, \forall i \in \mathcal{N}. \quad (2)$$

At most one packet can be transmitted in a timeslot t, i.e., the vector $\mathbf{p}(t)$ has at most one non-zero element. The set of power constraints for $\mathbf{p}(t)$ is then defined by

$$\mathcal{P}(t) \triangleq \left\{ \mathbf{p}(t) : \sum_{i=1}^{N} \mathbf{1}_{\{\mu_i(t)=1\}} \leq 1 \right\}, \quad (3)$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function.

In our system, a packet is dropped if its deadline has expired. Since the queue follows FIFO discipline, a packet is dropped under the following conditions: 1) it is at the head of the queue; 2) the remaining number of the slots to serve the packet is 1; and 3) power is not assigned to i at the current slot. Let $D_i(t)$ be the indicator of the packet drop for user i at time t. The queue evolution is described as

$$Q_i(t+1) \triangleq \max\left[Q_i(t) - \mu_i(t), 0\right] + \alpha_i(t) - D_i(t), \forall i \in \mathcal{N}. \quad (4)$$

Furthermore, we assume that $Q_i(0) = 0$, and $D_i(0) = 0$, $\forall i \in \mathcal{N}$. Let

$$\overline{D}_i \triangleq \lim_{t\to\infty} \overline{D}_i(t), \forall i \in \mathcal{N}, \quad (5)$$

$$\overline{p}_i \triangleq \lim_{t\to\infty} \overline{p}_i(t), \forall i \in \mathcal{N}, \quad (6)$$

respectively denote the packet drop rate and the average power consumption, where $\overline{D}_i(t) = \frac{1}{t}\sum_{\tau=0}^{t-1} D_i(\tau)$ and $\overline{p}_i(t) = \frac{1}{t}\sum_{\tau=0}^{t-1} p_i(\tau)$. The packet drop rate represents the average number of dropped packets per time slot. The average power consumption represents the average of transmit power over all time slots. These metrics are connected and we will show in the following sections how the average power consumption affects the packet drop rate.

III. Problem Formulation

We desire a scheduling scheme that offers fairness among users when minimizing their packet drop rate under average power constraints. Furthermore, we are interested in the trade-off between packet drop rate and time average power consumption. To this end, we present the following problem

$$\min_{\boldsymbol{p}(t)} \quad \sum_{i=1}^{N} \overline{D}_i \quad (7a)$$

$$\text{s. t.} \quad \overline{p}_i \leq \gamma_i, \forall i \in \mathcal{N}, \quad (7b)$$

$$\boldsymbol{p}(t) \in \mathcal{P}(t), \quad (7c)$$

where $\gamma_i \in \left[0, P^{(\text{High})}\right]$ indicates the allowed average power consumption. The constraint in (7b) ensures that average power consumption of each user i remains below γ_i power units.

The formulation above represents our intended goal which is the minimization of the packet drop rate. However, the objective function in (7a) has a basic disadvantage that makes the solution approach non-trivial. The decision variable, $\mathbf{p}(t)$ (power allocation), is optimized slot-by-slot for minimization of the objective function that is defined over infinite horizon. We have to cope with one critical point: We do not have any knowledge about the future states of the channel and packet arrival in the system. Therefore, we are not able to predict the values of the objective function in the future slots in order to decide on the power allocation that minimizes the cost. We aim to design a function whose future values are affected by

TABLE I: Notation Table.

$\mathcal{N}$	Set of users in the system	$\mathcal{P}_i(t)$	Set of selectable power levels of user i
t	t^{th} slot	$\mu_i(t)$	Power allocation indicator of user i
$Q_i(t)$	Number of packets in queue i	$\mathcal{P}(t)$	Set of power constraints for $\mathbf{p}(t)$
π_i	Packet arrival probability of user i	$D_i(t)$	Packet drop indicator of user i
$\alpha_i(t)$	Packet arrival indicator of user i	$\overline{D}_i$	Packet drop rate of user i
m_i	Deadline of packet of user i	$\overline{p}_i$	Average power consumption of user i
$d_i(t)$	Number of slots left before the deadline of user i	$X_i(t)$	Length of virtual queue of user i
$\mathbf{S}(t)$	Channel states	$L(\cdot)$	Quadratic Lyapunov function
$\mathbf{p}(t)$	Power allocation vector	$\Delta(L(\cdot))$	Lyapunov drift
γ_i	Allowed average power consumption for user i	$\boldsymbol{\alpha}(t)$	Packet arrival indicator vector

the current decision and the remaining expiration time of the packets. To this end, we introduce a function incorporating the relative difference between the packet deadline m_i and the number of remaining future slots $(d_i(t)-1)$ before its expiration as described below

$$f_i(t) \triangleq \frac{m_i-(d_i(t)-1)}{m_i}\mathbf{1}_{\{\mu_i(t)=0\}}. \tag{8}$$

The function in (8) takes its extreme value $f_i(t)=0$ when a packet of user i is served, or $f_i(t)=1$ when a packet of user i is dropped. Therefore, that function takes the same values with those of (5) in the extreme cases. In addition, the function in (8) assigns the cost according to the remaining time of a packet to expire in the intermediate states, i.e., when a packet is waiting in the queue. The cost increases when there is less time left for serving the packet with respect to the defined deadline. The time average of $f_i(t)$ is

$$\overline{f_i} \triangleq \lim_{t\to\infty}\overline{f}_i(t), \tag{9}$$

where $\overline{f}_i(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1} f_i(\tau)$. Finally, we formulate a minimization problem by using (9) as shown below

$$\min_{\boldsymbol{p}(t)} \quad \sum_{i=1}^{N}\overline{f}_i \tag{10a}$$

$$\text{s. t.} \quad \overline{p}_i \le \gamma_i,\ \forall i\in\mathcal{N}, \tag{10b}$$

$$\boldsymbol{p}(t)\in\mathcal{P}(t). \tag{10c}$$

IV. Proposed Approximate Solution

The problem in (10) includes time average constraints. In order to satisfy these constraints, we aim to develop a policy that uses techniques different from classic optimization methods based on static and deterministic models. For example, policies that select power less than γ_i at every time slot ensures that constraint (10b) is satisfied. However, this kind of policies decrease the degrees of freedom of power selection. In Table II, we provide an illustrative example with one user. We consider that $P^{(\text{Low})}=1$ power units, and $P^{(\text{High})}=2$ power units. In this example, the average power consumption must be less than or equal to 1.5 power units, i.e., $\gamma=1.5$ power units (subscripts are omitted for simplicity). We compare the performance of two policies ω_1 and ω_2. Policy ω_1 selects power less than γ power units at every time slot in order to

		$t=1$ $S(t)=\text{B}$	$t=2$ $S(t)=\text{G}$	$t=3$ $S(t)=\text{G}$
$d(t)$	ω_1	1	2	empty queue
	ω_2	1	2	1
$p(t)$	ω_1	0	1	0
	ω_2	2	0	1

		$\overline{p}(t)$	drop packets
$p(t)$	ω_1	1	1
	ω_2	1.5	0

TABLE II: Example showing the gain achieved by deciding different power allocations.

restrict the average power consumption below 1.5 power units. On the other hand, ω_2 allows power selection greater than 1.5 power units for each time slot. We observe that policy ω_2 achieves better performance than ω_1 by satisfying the power constraint. This motivates us to look for a more efficient way to satisfy the average power consumption constraint.

We apply the technique developed in [13] and further discussed in [14] and [15] in order to develop a policy that ensures that the constraint in (10b) is satisfied. Each inequality constraint in (10b) is mapped to a virtual queue. We show below that the power constraint problem is transformed into a queue stability problem.

Let $\{X_i(t)\}_{i\in\mathcal{N}}$ be the virtual queues associated with constraint (10b). We update each virtual queue i at each time slot t as

$$X_i(t+1) \triangleq \max\left[X_i(t)-\gamma_i,0\right]+p_i(t). \tag{11}$$

Process $X_i(t)$ can be viewed as a queue with "arrivals" $p_i(t)$ and "service rate" γ_i.

Before describing the motivation behind the mapping of power constraints to virtual queues, let us recall one basic theorem that comes from the general theory of stability of stochastic processes [16]. Consider a system with K queues. The number of unfinished jobs of queue i are denoted by $q_i(t)$ and $\mathbf{q}(t)=\{q_i(t)\}_{k=1}^{K}$. The Lyapunov function and the Lyapunov drift are denoted by $L(\mathbf{q}(t))$ and $\Delta(L(\mathbf{q}(t))) \triangleq E\left\{L(\mathbf{q}(t+1))-L(\mathbf{q}(t))|\mathbf{q}(t)\right\}$ respectively [16]. Before describing the Lyapunov Drift theorem, let us recall the definition of the Lyapunov function [16].

Definition 1 (Lyapunov function): A function $L:\mathbb{R}^K\to\mathbb{R}$ is said to be a Lyapunov function if it has the following properties

- $L(\mathbf{x}) \geq 0$, $\forall \mathbf{x} \in \mathbb{R}^K$,
- It is non-decreasing in any of its arguments,
- $L(\mathbf{x}) \to +\infty$, as $||\mathbf{x}|| \to +\infty$.

Theorem 1 (Lyapunov Drift): If there are positive values B, ϵ such that for all time slots t we have $\Delta(L(\mathbf{q}(t)) \leq B - \epsilon \sum_{k=1}^{K} q_n(t)$, then the system $\mathbf{q}(t)$ is *strongly* stable.

In below, we show that the power constraint problem is transformed into a queue stability problem. Then, we develop a dynamic algorithm that satisfies Theorem 1 in order to achieve stability.

Theorem 2: If $X_i(t)$ is *rate stable*[1], then the constraint in (10b) is satisfied.

Proof. Using the basic sample property [14, Lemma 2.1, Chapter 2], we have

$$\frac{X_i(t)}{t} - \frac{X_i(0)}{t} \geq \frac{1}{t}\sum_{\tau=0}^{t-1} p_i(\tau) - \frac{1}{t}\sum_{\tau=0}^{t-1}\gamma_i. \tag{12}$$

Therefore, if $X_i(t)$ is rate stable, so that $\frac{X_i(t)}{t} \to 0$, $\forall i$, with probability 1, then constraint (10b) is satisfied with probability 1 [17]. □

Note that strong stability implies all of the other forms of stability [14, Chapter 2] including the rate stability. Therefore, the problem is transformed into a queue stability problem. In order to stabilize the virtual queues $X_i(t)$, $\forall i \in \mathcal{N}$, we first define our Lyapunov function as

$$L(\mathbf{X}(t)) \triangleq \frac{1}{2}\sum_{i=1}^{N} X_i(t)^2, \tag{13}$$

where $\mathbf{X}(t) = \{X_i(t)\}_{i\in\mathcal{N}}$ and the Lyapunov drift as

$$\Delta(\mathbf{X}(t)) \triangleq \mathbb{E}\left\{L(\mathbf{X}(t+1)) - L(\mathbf{X}(t))|\mathbf{X}(t)\right\}. \tag{14}$$

The above conditional expectation is with respect to the random channel states and the arrivals.

To minimize the time average of the desired cost $f_i(t)$ while stabilizing the virtual queues $X_i(t)$, $\forall i \in \mathcal{N}$, we use the *drift-plus-penalty* minimization approach introduced in [15]. The approach seeks to minimize an upper bound on the following drift-plus-penalty expression at every slot t:

$$\Delta(\mathbf{X}(t)) + V\sum_{i\in\mathcal{N}} \mathbb{E}\left\{f_i(t)|\mathbf{X}(t)\right\}, \tag{15}$$

where $V > 0$ is an "importance" weight to scale the penalty.

We derive an upper bound for the drift by using the fact $(\max[Q-b,0]+A)^2 \leq Q^2 + A^2 + b^2 + 2Q(A-b)$ as shown below

$$X_i(t+1)^2 \leq X_i(t)^2 + p_i^2(t) + 2X_i(t)(p_i(t)-\gamma_i) + \gamma_i^2. \tag{16}$$

[1]A discrete time process $Q(t)$ is *rate stable* if $\lim_{t\to\infty}\frac{Q(t)}{t} = 0$ with probability 1 [14].

Taking the sum over all the queues in (16) we have

$$\sum_{i=1}^{N}\frac{X_i(t+1)^2}{2} - \sum_{i=1}^{N}\frac{X_i(t)^2}{2} \leq \sum_{i=1}^{N}\frac{X_i(t)^2 + p_i(t)^2 + \gamma_i^2}{2} + \sum_{i=1}^{N} X_i(t)(p_i(t)-\gamma_i). \tag{17}$$

Taking the expectations in (17), we have

$$\Delta(\mathbf{X}(t)) \leq B + \sum_{i=1}^{N} X_i(t)\mathbb{E}\left\{y_i(t)|\mathbf{X}(t)\right\}, \tag{18}$$

where $y_i(t) = p_i(t) - \gamma_i$, and B is constant,

$$B \geq \frac{1}{2}\sum_{i=1}^{N}\mathbb{E}\left\{X_i(t)^2 + p_i(t)^2 + \gamma_i^2|\mathbf{X}(t)\right\}. \tag{19}$$

Therefore, an upper bound for the drift plus penalty expression is

$$\begin{aligned}&\Delta(\mathbf{X}(t)) + V\sum_{i=1}^{N}\mathbb{E}\left\{f_i(t)|\mathbf{X}(t)\right\}\\ &\leq B + \sum_{i=1}^{N} X_i(t)\mathbb{E}\left\{y_i(t)|\mathbf{X}(t)\right\} + V\sum_{i=1}^{N}\mathbb{E}\left\{f_i(t)|\mathbf{X}(t)\right\}.\end{aligned} \tag{20}$$

A. Min-Drift-Plus-Penalty Algorithm

Note that the power allocation decision on slot t affects only the last two terms in (20). The proposed algorithm observes the virtual queue backlogs $\mathbf{X}(t)$ and the channel states and makes a control action to minimize the following expression

$$\sum_{i=1}^{N} X_i(t)\mathbb{E}\left\{y_i(t)|\mathbf{X}(t)\right\} + V\sum_{i=1}^{N}\mathbb{E}\left\{f_i(t)|\mathbf{X}(t)\right\}. \tag{21}$$

The algorithm decides the power allocation by solving the following optimization problem at each time slot

$$\min_{\boldsymbol{p}(t)} \quad V\sum_{i=1}^{N} f_i(t) + \sum_{i=1}^{N} X_i(t)y_i(t) \tag{22a}$$

$$\boldsymbol{p}(t) \in \mathcal{P}(t). \tag{22b}$$

In the following we show that the optimal solution to problem (22) minimizes the upper bound of the drift-plus-penalty expression given in the right-hand-side of (20). Let $\mathbf{p}(t)$ represent any, possibly randomized, power allocation decision made at slot t. Suppose that $\mathbf{p}^*(t)$ is the optimal solution to problem (22), and under action $\mathbf{p}^*(t)$ the value of $f_i(t)$ yields $f_i^*(t)$, and that of $y_i(t)$, $y^*(t)$, we have

$$V\sum_{i=1}^{N} f_i^*(t) + \sum_{i=1}^{N} X_i(t)y_i^*(t) \leq V\sum_{i=1}^{N} f_i(t) + \sum_{i=1}^{N} X_i(t)y_i(t). \tag{23}$$

Taking the conditional expectations of (23), we have

$$V\sum_{i=1}^{N}\mathbb{E}\left\{f_i^*(t)|\mathbf{X}(t)\right\}+\sum_{i=1}^{N}X_i(t)\mathbb{E}\left\{y_i^*(t)|\mathbf{X}(t)\right\}\leq V\sum_{i=1}^{N}\mathbb{E}\left\{f_i(t)|\mathbf{X}(t)\right\}+\sum_{i=1}^{N}X_i(t)\mathbb{E}\left\{y_i(t)|\mathbf{X}(t)\right\}. \quad (24)$$

In view of the above, it is concluded that the optimal solution to problem (22) minimizes the upper bound given in the right-hand-side of (20). Note that the solution we provide is an approximate solution because we minimize an upper bound of the drift defined in (20). Furthermore, we find an approximate solution of the problem in (10) by solving a snapshot problem (22) for a particular time slot t.

We summarize the steps of the proposed dynamic control algorithm to solve problem (10) in Algorithm 1, named dynamic power allocation (DPA) algorithm. DPA uses exhaustive search that solves the problem in (22).

Algorithm 1: DPA

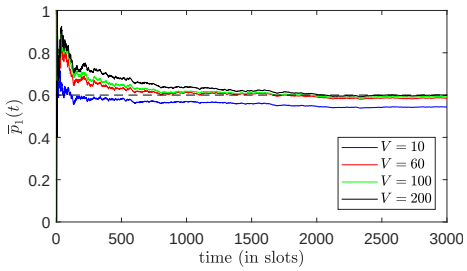
1 Input constant V, Initialization $X_i(0)=0$, γ_i, $\forall i\in\mathcal{N}$
2 **for** $t=1:\ldots$ **do**
3 $MinObj\leftarrow\infty$
4 **for** $i\in\mathcal{N}$ **do**
5 $p_i(t)\in\mathcal{P}(t)$, Calculate $f_j(t)$, $\forall j\in\mathcal{N}$
6 $Obj\leftarrow V\sum_{j=1}^{N}f_j(t)+\sum_{j=1}^{N}X_j(t)y_j(t)$
7 **if** *MinObj*>*Obj* **then**
8 $\mathbf{p}'(t)\leftarrow\mathbf{p}(t)$
9 $MinObj\leftarrow Obj$
10 $\mathbf{p}(t)\leftarrow\mathbf{p}'(t)$
11 $X_j(t+1)\leftarrow\max\left[X_j(t)-\gamma_j,0\right]+p_j(t)$, $\forall j\in\mathcal{N}$

In step 1, we initialize V and the length of virtual queues. We calculate the value of the objective function for each possible value of vector $\mathbf{p}(t)$ as shown in steps 5–6. In step 7, we compare each possible value of the objective function (for different power allocations) and keep the corresponding power allocation in vector $\mathbf{p}'(t)$ as shown in step 8. We decide power allocation as shown in step 10. The complexity of DPA is $\mathcal{O}(N^2)$.

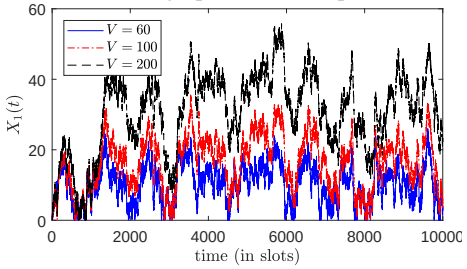
V. Numerical and Simulation Results

In this section, we compare the performance of DPA with that of earlier deadline first (EDF) algorithm. Recall that EDF finds across the users the packet with the shortest remaining expiration time and it assigns to its user the appropriate power according to the channel conditions. We compare the performance of the algorithms in terms of packet drop rate and average power consumption and we show the trade-off between them. Additionally, we provide results showing the performance of our algorithm for different values of V and how they affect the average power consumption.

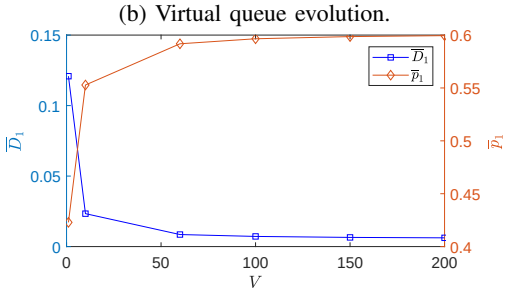
In the simulation setup, the probability a channel to be in "Bad" and "Good" state is 0.6 and 0.4, respectively. Also, we consider that the arrival process for each user i is an i.i.d. Bernoulli process with probability $\overline{\lambda}_i$. In addition, we consider



(a) Average power consumption.



(b) Virtual queue evolution.



(c) Tradeoff between packet drop rate and average power consumption.

Fig. 1: DPA performance depending on V. $\gamma_1=\gamma_2=0.6$, $\overline{\lambda}_1=\overline{\lambda}_2=0.4$.

that $P^{(\text{Low})}=1$, and $P^{(\text{High})}=2$. The deadlines are $m_1=m_2=5$ time slots.

Fig. 1 depicts how different values of V affect the packet drop rate and the average power consumption of user 1. We observe that the larger the value of V the slower the convergence of the algorithm in terms of power rate consumption constraint. However, it is shown in Fig. 1a that even for large values of V, DPA is able to keep the power rate consumption below γ_i and, therefore to satisfy the power consumption constraint. For large values of V, DPA allows virtual queue backlogs to take large values as shown in Fig. 1b. The reason why the backlogs of the virtual queues increase is because the dominant term of the objective function is the one that includes V. However, as the time passes by the virtual queue backlog increases and dominates the penalty term that includes V. Thus, DPA allocates lower power in order to decrease the virtual queue backlog and stabilizes it as shown in Fig. 1b. In Fig. 1c, we provide results for different values of V. We show the trade-off between the average power consumption and the packet drop rate. As expected, the average power consumption increases with increasing value of V. However, the average power consumption is always below 0.6.

Values of V that are larger than 60 do not affect significantly

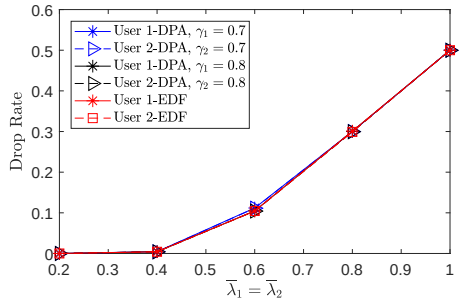


Fig. 2: Packet drop rate for EDF and DPA for different values of $\overline{\lambda}_i$ and γ_i.

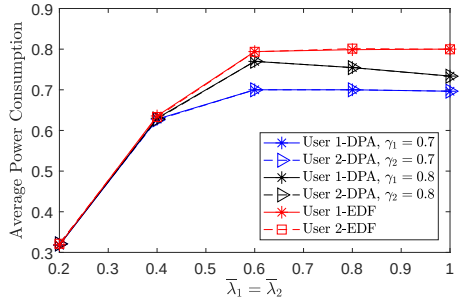


Fig. 3: Average power consumption for EDF and DPA for different values of $\overline{\lambda}_i$ and γ_i.

the packet drop rate. Thus we present the rest of the simulation results for $V = 60$. In Fig. 2 and Fig. 3, we compare the performance of the two algorithms in terms of packet drop rate and average power consumption. Note that EDF does not take into account the average power consumption of each user. Therefore, for some values of γ_i, EDF algorithm violates the average power constraints. For example, we see in Fig. 3 that EDF algorithm violates the average power constraints for $\gamma_1 = \gamma_2 = 0.7$. The performance of DPA in terms of packet drop rate is very close to that of EDF. However, we observe in Fig. 3, the average power consumption of DPA is lower than that of EDF by 0.1 power units. For $\gamma_1 = \gamma_2 = 0.8$, we observe that our algorithm has the same performance in terms of packet drop rate with that of EDF. However, in Fig. 3, we see that the average power consumption of DPA decreases when the traffic arrival exceeds a sufficiently large value, i.e., for $\overline{\lambda}_1 = \overline{\lambda}_2 > 0.6$. The reason why the average power consumption decreases is because that for large values of λ_i, the scheduler has often to cope with users having packets with one time slot left before their expiration. Thus, it selects to assign power to the user who has the best channel condition and drops the packet of the user with the worst channel condition.

Overall, we observe that our algorithm performs as the EDF algorithm when the power limit is sufficiently high. Furthermore, the proposed algorithm is able to satisfy the average power constraints of the users and offer a good trade-off between packet drop rate and average power consumption as shown in Fig. 2 and Fig. 3.

VI. Conclusions

In this paper, we propose a dynamic algorithm that decides power allocation at each time slot by minimizing an objective function. The proposed algorithm is based on Lyapunov optimization theory. We evaluate the performance of the proposed algorithm through simulations and compare it with EDF. We observe that our proposed algorithm has the same performance with EDF in terms of packet drop rate when the available power is sufficient. Furthermore, the proposed scheduling scheme can handle packets with deadlines and control the transmission power of the devices. Since we have systems with mobile devices and therefore, limited available power, it is important to develop a dynamic algorithm that satisfies the average power constraints of each user.

References

[1] I.-H. Hou and P. R. Kumar, "Packets with deadlines: A framework for real-time wireless networks," *Synthesis Lectures on Communication Networks*, vol. 6, no. 1, pp. 1–116, 2013.

[2] Y. Cui, V. K. N. Lau, R. Wang, H. Huang, and M. Shunqing Zhang, "A survey on delay-aware resource control for wireless systems–large deviation theory, stochastic lyapunov drift, and distributed stochastic learning," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1677–1701, Mar. 2012.

[3] L. Georgiadis, R. Guerin, and A. Parekh, "Optimal multiplexing on a single link: Delay and buffer requirements," *IEEE Trans. Inf. Theory*, vol. 43, no. 5, pp. 1518–1535, Sept. 1997.

[4] A. Fu, E. Modiano, and J. N. Tsitsiklis, "Optimal transmission scheduling over a fading channel with energy and deadline constraints," *IEEE Trans. Wireless Commun.*, vol. 5, no. 3, pp. 630–641, Mar. 2006.

[5] M. Goyal, A. Kumar, and V. Sharma, "Power constrained and delay optimal policies for scheduling transmission over a fading channel," in *Proc. IEEE INFOCOM*, vol. 1, May 2003, pp. 311–320.

[6] N. Salodkar, A. Bhorkar, A. Karandikar, and V. S. Borkar, "An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 732–742, May 2008.

[7] A. E. Gamal, E. Uysal, and B. Prabhakar, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *Proc. IEEE INFOCOM*, vol. 1, 2001, pp. 386–394.

[8] A. Dua and N. Bambos, "Downlink wireless packet scheduling with deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 12, pp. 1410–1425, Dec. 2007.

[9] N. Master and N. Bambos, "Power control for packet streaming with head-of-line deadlines," *Performance Evaluation*, vol. 106, pp. 1 – 18, 2016.

[10] E. Fountoulakis, N. Pappas, Q. Liao, V. Suryaprakash, and D. Yuan, "An examination of the benefits of scalable TTI for heterogeneous traffic management in 5G networks," in *Proc. IEEE WiOpt*, May 2017, pp. 1–6.

[11] A. Ewaisha and C. Tepedelenlioglu, "Power control and scheduling under hard deadline constraints for on-off fading channels," in *Proc. IEEE WCNC*, March 2017, pp. 1–6.

[12] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "On the age of information with packet deadlines," *IEEE Trans. Inf. Theory*, 2018.

[13] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, July 2006.

[14] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[15] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, 2006.

[16] S. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. 2nd ed. New York, NY, USA: Cambridge University Press, 2010.

[17] M. J. Neely, "Queue Stability and Probability 1 Convergence via Lyapunov Optimization," *ArXiv e-prints*, Aug. 2010. [Online]. Available: https://arxiv.org/abs/1008.3519

Dynamic Resource Trading in Sliced Mobile Networks

Özgür Umut Akgül, Ilaria Malanchini, and Antonio Capone

***Abstract*—Expanding the market of mobile network services and defining solutions that are cost efficient are the key challenges for next generation mobile networks. Network slicing is commonly considered the main instrument to exploit the flexibility of the new radio interface and core network functions in order to split resources among services with different requirements and tailor system parameters according to their needs. However, network slicing tends to be viewed by regulation authorities also as a way to open the market to new players that can specialize in providing new mobile services acting as "tenants" of the slices. Resources can be traded between infrastructure providers and tenants so as to match the requirements of the services offered. In this paper, we propose a model for mobile network resources that allows dynamic trading in a market that can automatically optimize technical parameters and economic prices according to high level policies set by the tenants. We introduce a mathematical formulation of the resource allocation and price setting problem and show how the proposed approach can cope with quite diverse service scenarios presenting a large set of numerical results.**

***Index Terms*—Network slicing, infrastructure sharing, wireless market, pricing mechanism, dynamic resource sharing, mobile virtual network operators (MVNO)**

I. Introduction

THE traditional business model of mobile networks is centered on operators that acquire licenses for spectrum use, build their own infrastructure, and control the resource allocation according to their needs. This model is currently being challenged by a number of economic, regulatory, and technical issues that are expected to change the mobile landscape in the near future.

The first well known issue is the exponential mobile traffic increase (cf. [1]) that is pushing operators to rapidly expand the capacity of their network with technology upgrades, coverage densification, and spectrum refarming. Unfortunately, the average revenues per user are not growing with the same pace of traffic (in some countries are even reducing), and the number of traditional users can no longer be increased. This is leading to an aggressive cost optimization and reduction that is however not sustainable in the long run. A possible way out has been identified in the evolution of the technology to support a much larger set of applications in addition to the traditional mobile broadband so as not only to expand the market but also to use once more the network infrastructure to stimulate the growth of the digital economy.

In the last years the focus of research first and then standardization on 5G networks has exactly been that of shaping a new technology not only able to improve the performance of previous ones, but also to support a wide range of vertical applications with very diverse and stringent requirements in terms of throughput, delay, reliability and energy [2]. However, because of fundamental technical limits, pushing to the extreme the performance on all these indicators simultaneously is not possible, and the network must be optimized on different working points depending on the specific application domain. The concept of network slicing has been introduced with the goal of allowing resource allocation to different applications and traffic classes so as to meet different quality requirements [3].

Even if slicing can be seen as a precious instrument for operators to manage their new generation networks, it nevertheless poses new challenges. A straightforward way of allocating resources to different slices can be through a (almost) static partitioning, which however can lead to low efficiency. Dynamic resource allocation can be a solution, but it must accurately consider traffic evolution and performance constraints of all applications. Moreover, the possibility to slice the network makes rather natural to consider new players, known as tenants with 5G terminology, that act as slice operators acquiring resources from traditional operators that tend to become infrastructure providers. From a regulatory perspective, the possibility to use slicing as a tool for infrastructure sharing is considered a way for creating new market opportunities and even explore new spectrum licensing strategies.

Generally speaking, the idea of infrastructure sharing among multiple mobile virtual operators has a relatively long history. Among the alternative sharing approaches listed by the Organization for Economic Co-operation and Development (OECD) report, active sharing is considered to be most cost-efficient sharing approach [4]. The active sharing includes sharing of both active network elements and spectrum resources. Virtual operators can then share resources with other operators and decrease costs [5]. Although a number of different sharing scenarios exist, the most common one includes a single infrastructure provider and a set of mobile virtual network operators that acquire resources in order to serve their users. For a given quality level, sharing allows saving resources with respect to the scenario of separate physical networks. The increased efficiency in resource usage and the adaptability to traffic conditions, are clear advantages [6] [7].

Most of proposed sharing models rely on pre-negotiated service level agreements (SLAs) that regulates responsibilities of each party and the fraction of resources to be assigned. Obviously, long term agreement with static resource assign-

Ö. U. Akgül and A. Capone are with Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milano, Italy.
I. Malanchini is with Nokia Bell Labs, Stuttgart, Germany

ments are not able to follow the fluctuations in the network demand [8]. Moreover, in wireless networks, there are some geographical areas that are not profitable for the virtual operators but still need to be covered by the infrastructure provider and associated costs are hard to be mapped into SLAs. For these reasons, dynamic sharing of infrastructure resources is a more attractive alternative where virtual operators or tenants can negotiate resource allocation based on needs following traffic and channel fluctuations [5] [9]. As argued in [7], the opportunity to dynamically adjust resource allocated, gives operators the possibility to take more business risks and thus, a dynamically shared wireless market tends to foster innovation. In this context, however, providing quality guarantees with heterogeneous traffic and different performance parameters is still an open problem that need to be addressed in order to apply infrastructure sharing to network slicing scenarios.

Unlike infrastructure sharing, network slicing is a relatively new concept. Despite the commonly accepted definition of vertically grouped network resources, the specific negotiable attributes of each slice and the instruments for service differentiation are still under discussion in the literature and standardization bodies. In this work, we adopt the concept of a slice as a set of dedicated network resources assigned for specific services in a time interval. In order to assign resources to slices efficiently, the channel conditions, traffic characteristics and variations, and service heterogeneity must be considered [10]. The benefits of network slicing are investigated in [11]–[13] considering static SLAs without dynamic resource adaptation. A virtualization framework, where the resources are scaled according to tenants' dynamic needs and fairness is guaranteed not only between tenants, but also between users of different services is proposed in [14]. The model however does not consider adaptation to channel conditions and economic aspects of resource trading. In [15], we have proposed a first step towards dynamic network slicing in a shared network where tenants are able to renegotiate their slice sizes. In our proposed scheme, tenants retain service level guarantees, but they can trade resources on a very short time scale so as to exploit fluctuations in traffic and channel condition and efficiently control costs.

An important element for tenants and their business strategies (i.e. making long term plans, analyzing the possible risks and performing innovation) is a reasonable and predictable pricing model [7]. In the conventional network provisioning model, the infrastructure provider (whether it is a local operator or a specialized entity) charges tenants according to costs associated to its long-term infrastructure strategy, which may not always be in line with their market and even not able to meet requests for all tenants [16]. More in general, the pricing model of infrastructure providers can create barriers for the entrance of new players, as already shown for the traditional mobile virtual operator approach [17]. The structure of the competition on a geographical distributed resource tends to favor a small number of big operators [18], eventually leading to a monopoly that can slow down innovation [7]. However, with dynamic infrastructure sharing, since the resources are pooled and tenants can adjust their shares dynamically, a more efficient and neutral pricing framework can be potentially achieved [19].

A reasonable approach is that of using variable market driven prices and allowing tenants trading the resources to acquire on a short time scale based on needs and current prices. Unfortunately, without a clear model that allows to understand the relation of economic aspects with technical performance, it is unlikely that tenants can exploit the full potential of the dynamic sharing. Thus, a scheme able to automatically define prices and resource allocation based on high level tenant strategies and traffic estimation is of fundamental importance [8]. Even if there is a quite large literature focused on the economic aspects (such as [17], [20]) and technical aspects (such as [21], [22]) separately, the definition of techno-economic models for resource sharing in sliced networks is still an open point.

In this paper, we propose a dynamic wireless market model that can flexibly adjust the share of resources assigned to network slices in order to achieve the maximum utility for tenants. The contributions of this work can be summarized as follows:

- An enhanced short time scale wireless market model based on different services and service quality metrics,
- Integration of channel-aware opportunistic resource allocation and trading in a sliced shared network,
- Self-optimization of the network slices based on the tenants' market power and the traffic mix,
- Self-adaptation of the resource distribution according to the wireless channel condition,
- Dynamic and automated market driven pricing of the wireless resources.

The remainder of the paper is organized as follows: Section II contains the system model and the main assumptions. Following the system model, the optimization model is presented in Section III. In Section IV, the behavior and the validity of the optimization model are investigated through simulations. Section V concludes the paper and discuss possible extensions of the proposed approach.

II. System model

In order to provide a flexible and adaptive resource sharing algorithm for network slicing in a multi-tenant environment, we introduce a dynamic negotiation platform, shown in Fig. 1, which interacts with the different stakeholders and, based on the received inputs, allocates resources, assesses the performance and evaluates the corresponding costs. Namely, in our system model, the stakeholders are as follows: a set of tenants M, with index m, sharing the downlink of a base station, an infrastructure provider (InP) who provides the shared base station, and a set of users K, which requires heterogeneous services to their corresponding tenant. Also, let the set K_m be the set of users of tenant m, and thus $\sum_{m \in M} |K_m| = |K|$. In particular, we assume that each user requests only one type of service and the number of active users per tenant, i.e. the cardinality of K_m, is the same for all tenants (i.e. tenants have similar market shares). Time is discretized into slots, n, where N is the set of all time slots, i.e. simulation horizon.

In order to regulate the sharing of resources, service level agreements (SLAs) exist between the InP and the tenants. In

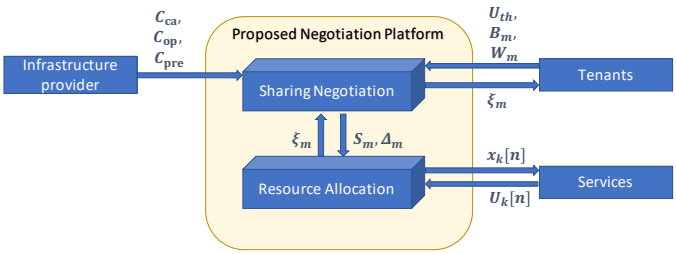


Fig. 1. Proposed negotiation platform.

particular, we assume that the slice of tenant m is defined by three parameters, S_m, Δ_m and W_m. $S_m \in (0,1)$, referred to as *guaranteed resource share*, indicates the ratio of resources that tenant m expects to receive in average. Furthermore, to guarantee flexibility, we assume that the resource allocation can deviate from the guaranteed resource share. In particular, the *maximum average allowed deviation* is denoted as Δ_m (as introduced in [23]). Namely, Δ_m sets the limit on the maximum deviation from S_m within a tenant-specific time window, W_m (over which the average is computed). Therefore, within each time window W_m, tenant m receives (in average) a fraction of resources between $(S_m - \Delta_m, S_m + \Delta_m)$. Note that, the time constrained imposed by the time window W_m can also be used to achieve differentiation among tenants and corresponding services. Differently from [23], where sharing parameters were assumed to be constant, in this work, to fully exploit the advantages of dynamic trading, S_m and Δ_m are periodically updated. Namely, the period of such updates is set by the InP and is referred to as "renegotiation interval".

Furthermore, we assume that tenants set their utility targets, $U_{\text{th}} \in (0,1)$ and their available budgets, B_m. In contrast, the InP is responsible for setting the respective costs of the wireless resources (c.f. Fig. 1). The total cost of the wireless resources consists of three parts, i.e., capital expenses, C_{ca}, operational expenses, C_{op} and pressure cost, C_{pre}. We assumed that the infrastructure provider does not have profit constraints and his main objective is to run a sustainable business model. Therefore, C_{ca} and C_{op} are scaling the cost of the conventional infrastructure and the operational cost of the resources. The pressure cost helps the regularization of the resource allocation. Similar to any demand based market, the pressure cost also regulates the resource consumption. For instance, if the system does not have sufficient resources to satisfy all the users, i.e. *resource scarcity*, the pressure cost is set to be greater than zero, so that tenants will have less incentive to buy resources (in terms of S_m), but more incentive to trade resources (via Δ_m). In contrast, in case the system has more than sufficient resources for all the users, i.e, *resource surplus*, the pressure cost is set to zero, reducing the overall cost and increasing the incentive to buy. Moreover, pressure cost can be seen as a way for the InP to collect the necessary revenue in order to upgrade or expand the existing network capacity (in case of resource scarcity). The pricing mechanism is further explained in Section III.

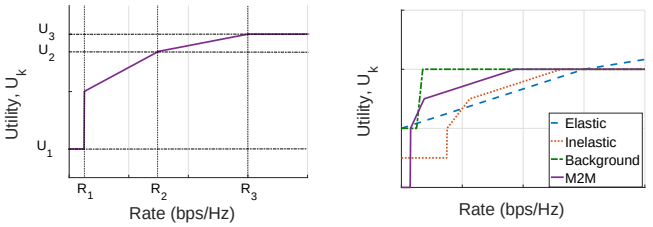
Based on all the inputs described above as well as the users' channel conditions, the proposed negotiation platform optimally allocates the resources to the different slices. Namely, let $x_k[n]$ be the wireless resources allocated to user k at time slot n, and $r_k[n]$ the achievable rate for user k at time slot n. The actual achieved rate of user k at time slot n is then given by $r_k[n]x_k[n]$. Furthermore, we assume that each user k produces a utility $U_k[n]$ that depends on the achieved rate as well as the requested service type. The average achieved utility of tenant m at n is the average achieved utility over all its users, i.e. $\sum_{k \in K_m} \frac{U_k[n]}{|K_m|}$. The difference between the utility target U_{th} and the average achieved utility is defined as the tenant's gap and denoted by $\xi_m[n]$. Such gap is used to measure the performance of the proposed resource sharing algorithm, where the best possible operating point is the one for which the gap is equal to zero.

A. Utility Functions

Even if the quality perceived by the users depends on several elements, we assume in this paper that it can be quantified by using the achieved rate. We therefore consider a generic continuous utility function $U_k(R_k[n])$, function of the average achieved rate $R_k[n]$, as shown in Fig. 2(a). This function is used in our framework to model different utilities for heterogeneous services. Namely, each specific service function is determined by varying six parameters, i.e. U_1, U_2, U_3, R_1, R_2 and R_3. The minimum rate, required to consider a service as active, is assumed to be R_1. When the average achieved rate is lower than R_1, i.e. $R_k[n] < R_1$, the utility function returns the utility value $U_1 \leq 0$. In case the service achieves the average rate of R_1 than the utility returns zero. R_2 represents the standard quality for the services where the utility function provides a utility value equal to U_2. Finally, R_3 indicates the saturation point for the utility function, after which the function becomes non-increasing. The maximum utility for the service type, that is achieved at $R_k[n] = R_3$, is given by U_3. Note that the choice of piece-wise linear functions is mainly due to mathematical tractability, but this does not limit the validity of the proposed sharing platform, which can incorporate also more complex functions.

Using the generic utility function presented above, we defined the specific utility functions for four service types envisioned for 5G: elastic, inelastic, M2M and background services. In particular, prioritization (or fairness) among services (and in particular between critical and non-critical services) can be set by using different (or equal) slopes of the utility functions (e.g. between $R_1 - R_2$ and $R_2 - R_3$). A detail description of the specific utility functions chosen for the four different services is given in the following and reported in Fig. 2(b).

1) Elastic traffic: By definition, elastic services, do not have strict rate or delay constraints. Thus, we consider them to be active as soon as the average achieved rate is greater than zero, $R_k[n] > 0$, meaning $R_1 = 0$ and $U_1 = 0$. Moreover, for elastic users we do not set any upper bound on their rate expectations, meaning $R_3 \to \infty$ and $U_3 \to \infty$. Since the service requirements are quite flexible, the utility function has a smaller slope compared to the one of the other services in any of the same regions.



(a) Generic utility function. (b) Exemplary utility functions.

Fig. 2. Generic utility function (left) and exemplary utility functions per service type (right).

2) Inelastic traffic: Being a demanding service type, inelastic services require a minimum rate to provide service availability, e.g. as in the case of video streaming. For this reason, we set R_1 relatively high, e.g. to provide a continuous service experience to the users. Similar to video streaming, the utility of inelastic services (i.e. perceived quality) is highly affected by the fluctuations of the achieved rate (e.g., the variations in the video quality between 144p and 720p). Therefore, we impose a steep slope between R_1 and R_2 to force a quick increase in the utility as a function of the average achieved rate. However, after reaching a certain quality, the increase in the average achieved rate is less noticeable, and therefore, between R_2 and R_3 we choose a lower slope. As mentioned above, to enforce fairness, the slope of inelastic services between R_2 and R_3 is equal to the one of elastic services between R_1 and R_2.

3) Background traffic: Background services refer to services that usually run in the background and require relatively very low rate and as soon as this is reached, the utility function reaches its saturation point, i.e. $R_2 = R_3$. Furthermore, since they do not have a strict delay constraint, the minimum utility is considered to be zero, i.e. $U_1 = 0$.

4) Machine to machine traffic: We group the heterogeneity of the M2M services envisioned in 5G into three main categories and model the M2M requests as a mixture of all three of them. Namely, the M2M utility function represents three types of services, i.e. emergency, low-rate-delay-sensitive and rate sensitive. Here we assume that M2M incorporates all three services but how the tenant specific resource distribution within M2M is not covered in this work. However, we consider that tenants will prioritize their M2M services and assign resources accordingly. The emergency services, which require low rate but also very high priority, are modeled with the R_1 rate and, since not achieving this rate can have a dramatic impact on the system, we set U_1 to a negative value. Hence, not serving the emergency services results in a big gap for the tenants. The low rate and delay sensitive M2M applications are modeled between R_1 and R_2. As shown in Fig. 2(b), for this type of services, since there is a delay constraint, the utility function characteristic has a relatively large slope. Finally, for the rate constrained services, as the name suggests, achieving higher rates has higher priority than having a low delay. Therefore they are modeled between R_2 and R_3 with a relatively smaller slope.

III. Scheduling problem and analysis

A. Mathematical programming formulation

The scheduler of the shared base station allocates resources by using the optimization model formulated in (1a)−(1h). The proposed techno-economic model runs in real time and controls both the resource allocation and the respective price negotiations in an online manner. Namely, the resource shares of the tenants are dynamically chosen based on their Quality of Service (QoS) expectations (i.e. the achieved rate per user and tenant's time window, W_m), the channel conditions and tenant's market power (i.e. their budget, number of users and traffic mix). The optimizer dynamically assigns resources to each slice per service type and per tenant to minimize the total gap, i.e., as in (1a), $\sum_{m\in M}\xi_m$. By jointly optimizing the resource allocations for all tenants, the scheduler has the flexibility to prioritize the users with the best channel conditions and therefore maximize the utilization of the resources and spectral efficiency.

Constraint (1b) sets the gap of tenant m as the difference between its target utility (i.e. U_{th}) and the sum of the achieved utility over its users (i.e. sum of $U_k(R_k[n])$). Note that within each time window, of length W_m, we evaluate the average by considering the values from the beginning of the time window to the current time slot n, i.e. over $a_m + 1$ time slots, where $a_m \equiv n - 1 \mod W_m$. Therefore, the average achieved rate for user k at time slot n is

$$R_k[n] = \frac{1}{(1+a_m)}\left(\sum_{i=n-a_m}^{n} x_k[i]r_k[i]\right).$$

Furthermore, we assume that all the users have the same importance to the tenants, thus, $U_3 = U_{\text{th},m}/K_m \ \forall k \in K_m$. By selecting the same value of maximum utility, U_3, for all the users, the tenants also claim neutrality in their provided services. However, depending on the agreements between the service providers and the tenants, as well as in accordance to regulatory constraints, this value can be changed, thus allowing our model to include also non-neutral services.

The instantaneous average deviation from the guaranteed resource share, $\epsilon_m[n]$, is given in (1c). Namely, the instantaneous deviation at n for tenant m is given by subtracting the guaranteed resource share S_m from the average assigned resource to the users of m, where the average, as done for the average achieved rate, is evaluated from the beginning of the current time window till time slot n. Constraint (1d) ensures that $\epsilon_m[n]$ is not larger than Δ_m, which by definition is the tenant-specific maximum allowed deviation. Note that ϵ_m can be either positive or negative, i.e. $\epsilon_m \in [-\Delta_m, \Delta_m]$. The former case indicates that the tenant has received – on average and within the current time window – more resources than S_m, while the latter case corresponds to the opposite.

Furthermore, constraint (1e) sets the budget constraint per tenant. The first term of the left-hand-side scales both CAPEX and OPEX according to S_m, which means that in case of no sharing (when $\Delta_m = 0$) the tenant will have to pay for the requested resources. The second term, i.e. $\epsilon_m[n]C_{\text{op}}$, allows tenants to dynamically adjust their total cost according to their resource usage and budget. Namely, if a tenants' actual

$$\min_{x_k[n]} \sum_{m\in M} \xi_m[n] \tag{1a}$$

$$\text{s.t. } U_{th,m} - \sum_{k\in K_m} U_k(R_k[n]) \le \xi_m, \ \forall m \in M, \tag{1b}$$

$$\epsilon_m[n] = \left(\frac{1}{(a_m+1)} \sum_{i=n-a_m}^{n} \sum_{k\in K_m} x_k[i]\right) - S_m, \ \forall m \in M, \tag{1c}$$

$$|\epsilon_m[n]| \le \Delta_m, \quad \forall m \in M, \quad \forall n \in N, \tag{1d}$$

$$\sum_{i=n-a_m}^{n} (S_m(C_{\text{ca}} + C_{\text{op}}) + \epsilon_m[i]C_{\text{op}} + f_{\text{pre}}(C_{\text{pre}}, \ \xi_m)) \le B_m(a_m+1), \forall m \in M, \tag{1e}$$

$$0 \le \Delta_m \le \frac{1}{a_m+1} \sum_{i=n-a_m}^{n} \sum_{k\in K_{m,elastic}} x_k[i], \quad \forall m \in M, \tag{1f}$$

$$\sum_{k\in K} x_k[n] \le 1, \quad x_k[n] \ge 0, \quad \forall k \in K, \tag{1g}$$

$$\sum_{m\in M} S_m \le 1 \ , \quad S_m \ge 0, \quad \forall m \in M, \tag{1h}$$

resource usage is less than the guaranteed resource share (i.e. $\epsilon_m[n] < 0$), then the tenant will not pay for the OPEX cost of the unused resources. The third term in the left-hand-side of the budget constraint is a function, $f_{\text{pre}}(C_{\text{pre}}, \xi_m)$, of the pressure cost unit C_{pre}, defined by the InP, and of the tenant's gap ξ_m. Namely, the gap considered for the evaluation of the pressure cost is the one obtained at the end of the previous time window (i.e. it varies at every time window, but kept constant within the same time window). The effects of the pressure cost term are evident when, e.g., there is a resource demand that exceeds the available resources. In this case, since the resources are limited, the tenants face non-zero gaps, $\xi_m > 0$, which corresponds to an increase of the pressure cost as well as of the total cost of resources. This increase in the cost pushes tenants to increase their Δ_m and decrease S_m. In the extreme case, tenants opt for full sharing, i.e. $\Delta_m = 1$, which allows the scheduler to provide the most spectrum efficient and cost efficient allocation. Moreover, the pressure cost allows the infrastructure provider to accumulate additional revenues not directly used for the current infrastructure, but envisioned to support capacity expansion to meet the tenants' quality requirements. In this respect, scaling the pressure cost by the gap provides an accurate estimation of the capacity needed to satisfy all the tenants.

Constraint (1f) forces the maximum deviation Δ_m to be at maximum equal to the resources assigned to the elastic users of tenant m, which implies that tenants are not willing to trade resources used for critical, i.e. non-elastic, services. By setting $\Delta_m = 0$, tenants indicate that their services are non-elastic and they require the resources they stated by S_m. However, in this case, they also lose the flexibility to adapt to traffic dynamics. Finally, (1g) ensures that the assigned resources do not exceed the total available resources in the system and, similarly, (1h) limits the sum of all S_m to the total amount of resources.

B. Two-step approach

The formulation presented in the previous section is able to capture the dynamics of the resource negotiation, considering both the scheduling aspects as well as the economical constraints (prices and budgets). However, due to its computational complexity, it is not suitable to be used in real-time. Therefore, we decide to split the two decisions that have to be taken, namely on the real time resource allocation and on the negotiations of the sharing parameters.

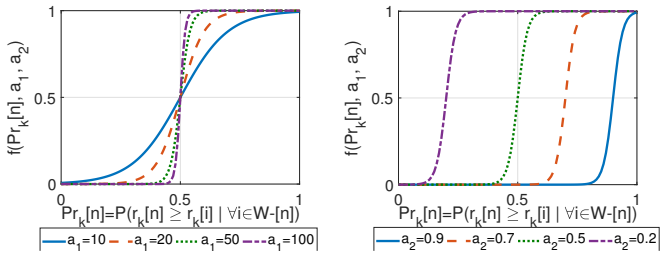
In particular, we separate our model into two sub-problems, P_1 and P_2. The first problem, P_1, focuses on the real time resource allocation with the objective of minimizing the total gap and it is solved at every time sot n. During P_1, the sharing parameters (S_m, Δ_m) are assumed to be constant and, therefore, the constraints that regulates the sharing (i.e. (1f) and (1h)) are inactive. The outcome of P_1 is then given by the allocated resources and corresponding tenants' gaps. The second problem, P_2, is solved at the end of each time window, to update the sharing parameters according to the current users' channel conditions and tenants' targets (i.e. in terms of $U_{th,m}$). In this case, the objective is to find the best sharing parameters so that the total gap of the previous time window is minimized. Namely, P_2 receives as input the achievable rates from the previous time window and derives the optimum sharing parameters S_m^{opt} and Δ_m^{opt} by solving (1a)−(1h).

Note that even if both P_1 and P_2 are derived from the same formulation (1a)−(1h), they are actually different problems since the active variables (and constraints) are different.

C. Exploiting the channel information

The real-time scheduling problem, P_1, myopically focuses on the optimization of the current time slot n without taking into account the upcoming slots. Thus, it is incapable of fully exploiting the transmission opportunities. As a result, P_1 requires a larger amount of resources compared to the one estimated by P_2 in order to provide comparable performance. As a matter of fact, P_2 derives the minimum values of S_m and Δ_m required to minimize the gap, which are overly restricting for P_1. Therefore, to improve the performance of P_1, a channel-aware filter is designed to exploit the statistical information of the channel.

Specifically, we design a channel-aware filter to evaluate the rate expectations for the upcoming time slots of each user, while scheduling the resources for the given time slot n. Even though prediction techniques of the channel characteristics are out of scope of this paper, we assume that the infrastructure provider can derive a statistical profile of the channel behaviors. Therefore, we assume that the infrastructure provider learn a probability density function of the achievable rates for each user $k \in K$, which can be used to evaluate the probability, for that specific user within the given time window, of being in the "best" time slot to assign resources, $Pr_k[n] = P(r_k[n] \ge r_k[i] \ \forall i \in W) \in [0, 1]$, i.e. the slot with best channel conditions compared to the other time slots. In particular, a probability value of 0 indicates that the channel



(a) Varying a_1, when $a_2 = 0.5$. (b) Varying a_2, when $a_1 = 50$.

Fig. 3. Variation of the sigmoid function for different a_1 (left) and a_2 (right) values.

condition at slot n is the worst that can ever be observed, thus, the scheduler should avoid assigning resources, while a value of 1 means that the current channel condition is the best possible and therefore as many resources as possible should be assigned. However, we do not use directly this probability, but we filter it as described below before passing it as input to P_1.

We design a two-step filtering function to map the statistical information onto the assignment decisions. On the first step, the statistical information is scaled using a sigmoid function, i.e. $f(Pr_k[n], a_1, a_2) = 1/(1+e^{-a_1(Pr_k[n]-a_2)})$, as presented in Fig. 3. The characteristic of the sigmoid function can be controlled by using two parameters, i.e. $[a_1, a_2]$ (cf. Fig. 3(a) and Fig. 3(b)). The former parameter, a_1, controls the slope of the linear region of the sigmoid and indirectly controls the resource efficiency. Namely, assuming that the number of users is low, decreasing the slope of the linear region leads to a situation where there exists unassigned resources while the tenants cannot achieve their goals. In contrast, increasing a_1 results in assigning resources also with bad channel conditions, thus decreasing the efficiency of the channel utilization. The latter parameter, a_2, allows the shift of the sigmoid function (c.f. Fig 3(b)). In this case, choosing large values of a_2 gives advantages only to the users with high probabilities. However, when tenants select small time windows, this leads to unassigned resources even in the presence of gaps. In contrast, small values of a_2 equalizes all users does making the filter ineffective.

The output of the sigmoid function, $f(Pr_k[n], a_1, a_2)$, provides an understanding on how good the channel conditions for a specific user are with respect to what such user can achieve in the given time window. However, $f(Pr_k[n], a_1, a_2)$ does not give information about how good the channel is with respect to the other users in that time slot. Therefore, the first step of the filter might not be sufficient to guide the scheduler when there is a significant difference among the distributions of the users' channel.

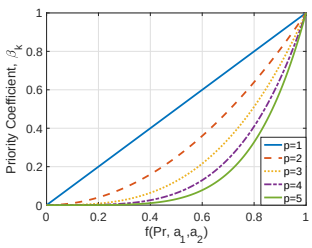


Fig. 4. Changes in the charachteristic of the filter function according to the variations of p.

Consequently, an additional filtering step is introduced to capture this variations among users' channel conditions. More specifically, taking the output of the sigmoid function, $f(Pr_k[n], a_1, a_2)$, the second step outputs $f(Pr_k[n], a_1, a_2)^p$, where p is a scalar. If the variations in the achievable rates among users are negligibly small, e.g. the users have similar pathlosses, p value can be set to 1. In contrast, if the difference is not negligible, a larger value of p should be chosen.

The output of the filter function, referred to as "priority coefficient" and indicated by $\beta_k[n]$, is then used by the scheduler to give priority to the users with the best channel condition (i.e. $\beta_k[n] = 1$) and to discard the users with worst channel conditions (i.e. $\beta_k[n] = 0$). In order to incorporate this information in P_1, the constraint (1b) is updated as

$$U_{th,m} - \sum_{k \in K_m} \beta_k[n] U_k(R_k[n]) \leq \xi_m, \ \forall m \in M. \quad (2)$$

Since the channel information is used to guide the real-time scheduling algorithm, the gap values calculated by P_2 are then derived without priority coefficients, as given in (1b).

Note that the specific values chosen for $[a_1, \ a_2]$ as well as p, combined with the channel conditions, affect the resource allocation. Hereafter, we do not discuss the policies used by the tenants to select those values, but assumed they are given (i.e. we empirically derived those used for the numerical evaluation).

D. Update mechanism

As described above, P_2 derives the optimum sharing parameters, i.e. S_m^{opt} and Δ_m^{opt}, for all the tenants, in order to achieve the minimum total gap $\sum_{m \in M} \xi_m^{\text{opt}}$. However, it is important to remember that the optimization problem is solved by using the achievable rates of the previous time window only, meaning that S_m^{opt} and Δ_m^{opt} are optimal only with respect to the previous window. Therefore, to capture the statistic nature of the channel over a longer time span, the sharing parameters are updated with a weighted approach. Namely, the new values for the sharing parameters, S_m^{new} and Δ_m^{new} to be used in the upcoming time window, are derived as:

$$S_m^{\text{new}} = \alpha_m S_m^{\text{opt}} + (1 - \alpha_m) S_m^{\text{old}}, \quad (3)$$

$$\Delta_m^{\text{new}} = \alpha_m \Delta_m^{\text{opt}} + (1 - \alpha_m) \Delta_m^{\text{old}}. \quad (4)$$

where the feature scaling coefficient, α_m, is calculated as:

$$\alpha_m = \frac{\xi_m - \xi_m^{\text{opt}}}{\xi_m + \xi_m^{\text{opt}}}. \quad (5)$$

By definition α_m measures the difference between the achievable optimum gap and the actual gap observed by the tenant. For instance, when $\xi_m = \xi_m^{\text{opt}} = 0$, the feature scaling coefficient is also 0, which means that the most recently calculated sharing parameters are the optimum values and therefore used also for the upcoming time window without

scaling. In general, with the proposed update mechanism, our framework is able to adapt to the varying channel conditions in a reactive manner. The sharing parameters are automatically updated to provide service quality which is satisfying the tenants requirements while maintaining proportional fairness among them. A thorough study of the α_m selection and its effects on the model's adaptability has been proposed in [8].

IV. Simulation results

In this section, we first present the parameters and the simulation setup considered for the evaluation and then show the effectiveness of the proposed algorithms with some numerical results.

A. Parameters and simulation setup

We consider the downlink of a single base station that is shared among $|M|$ tenants. Unless specified otherwise, each tenant serves $|K_m| = 4$ users and each user is associated with a specific traffic type, i.e. elastic, inelastic, M2M or background. The total set of users, $K = \cup_m K_m$, is distributed homogeneously in the coverage area of the base station and considered to be active for the entire simulation duration, which is set to $N = 5000$ time slots, each of length equal to 1 ms. The presented results are averaged over 100 independently generated instances.

The parameters that are used for the utility functions, reported in Fig. 2, are given in Table I. The utility target is $U_{\text{th},m} = 1, \ \forall m \in M$. Unless specified otherwise, the length of the time window, W_m, is considered to be equal, for all tenants, to the renegotiation interval, assumed to be 80 ms long. The values used for the costs and budgets are $C_{\text{ca}} = 20, \ C_{\text{op}} = 20, \ B_m = 100, \forall m \in M$. As proposed in [8], when tenants have all the same budget, the pressure cost is evaluated as C_{ca} scaled by the number of tenants, i.e. $C_{\text{pre}} = \frac{C_{\text{ca}}}{|M|}$ and $f(C_{\text{pre}}, \xi_m[|W_m|]) = \xi_m[|W_m|] \times C_{\text{pre}}/|W_m|$.

A frequency-flat fading channel is assumed between the base station and the users with i.i.d. Rayleigh coefficients leading to exponential channel gains, $|h_k[n]|^2$. Based on this, the Signal to Interference-plus-Noise Ratio (SINR) is calculated for each user k at each time slot n as:

$$\gamma_k[n] = |h_k[n]|^2 \frac{P d_k^{-\alpha}}{\sigma^2 + I_0}, \tag{6}$$

where P is the transmit power (in Watts), d_k is the distance between the user k and the base station (in meters) and α is the path-loss exponent. In this work, the interference is modeled as the sum of the thermal noise, σ^2 and the average interference, I_0. Therefore, by using (13), the achievable rate of user k at time slot n is expressed by

$$r_k[n] = \log_2(1 + \gamma_k[n]). \tag{7}$$

Finally, the considered filter values (introduced in Section III-C) are set to $a_1 = 10, \ a_2 = 0.5, p = 3$.

TABLE I
Service specific parameters and their values.

Parameter	Elastic	Inelastic	M2M	Background
R_1 (bps/Hz)	0	0.1	0.01	0.05
R_2 (bps/Hz)	1.083	0.225	0.075	0.07
R_3 (bps/Hz)	∞	0.55	0.4	0.07
U_1	0	-0.5	-1	0
U_2	1	0.7	0.7	1
U_3	∞	1	1	1

B. Time complexity analysis

As briefly analyzed in [8], the renegotiation interval, which is set by the InP affects the time complexity of the algorithm. Table II depicts the variation of average computation time of P_1 and P_2 depending on the renegotiation interval in a scenario with $|M| = 3$, $|K| = 12$. The simulations are run in Matlab, whereas the optimization problems P_1 and P_2 are solved by the Gurobi commercial solver [24]. The simulations are run on a Intel 2.4 GHz PC with 6 GB of RAM.

Results show that the longer the renegotiation interval, the longer the time to solve P_2. This is reasonable since the algorithm has to find the optimal sharing parameters over a longer time interval. In contrast, the solving duration of the real time scheduler, P_1, is mainly not effected by the length of the renegotiation interval.

TABLE II
Effects of renegotiation interval on computation time.

Renegotiation Interval	P_1 duration (sec)	P_2 duration (sec)
5 ms	0.0015	0.0431
25 ms	0.0012	0.1923
50 ms	0.0016	0.5069
80 ms	0.0011	1.4832
100 ms	0.0015	2.4412

Note that both P_1 and P_2 have time constraints dictated by the system model we proposed. Namely, we need to run P_1 every time slot and P_2 every time window. In order to obtain acceptable computation time for real time implementation two different approaches could be used. From one side, P_1 could be run using more powerful machine to reduce the computation time below 1 ms. On the other side, for cases where the computational time of P_2 becomes too large, an alternative heuristic could be proposed, which is, however, out of the scope of this paper.

C. Value of channel information

In Sec. III-C, we introduce a channel-aware filter to integrate the statistical channel information in the real time scheduler. Basically, we propose to replace constraint (1b) with constraint (9). The proposed channel-aware approach is a simple prediction algorithm, that evaluates current channel conditions taking into account past observations and future expectations.

Hereafter, we want to show the effects of exploiting such channel information on the total achieved gap with respect to: (1) the case without channel information (P_1 solved using

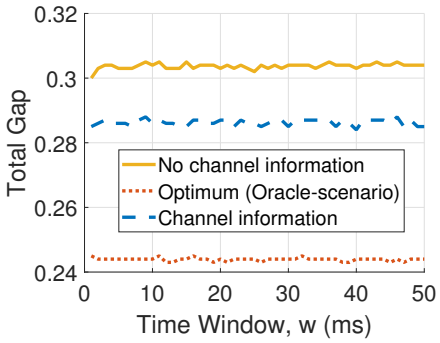


Fig. 5. Effects of integrating channel information on the total gap, for $|M| = 2$, $|K| = 8$.

constraint (1b)) and (2) the case with perfect knowledge of the future channel conditions (oracle scenario). Fig. 5 shows the results for $|M| = 2$ tenants and $|K| = 8$ users. We observe that feeding the model with an estimation of the channel allows the scheduler to better detect the instantaneous opportunities and to increase resource and cost efficiency, by decreasing the total gap.

TABLE III
CHANNEL INFORMATION'S IMPROVEMENT ON THE TOTAL GAP WITH RESPECT TO NO-CHANNEL INFORMATION CASE.

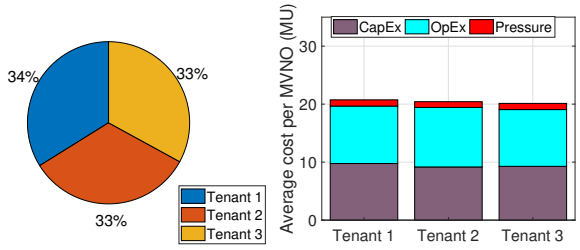
$\|K\|$	Improvement of total ξ_m
8	33.2%
16	38.5%
24	38.6%

Table III shows the effect of increasing the number of users $|K|$ on the total gap, as percentage improvement with respect to the oracle case. Increasing $|K|$ gives the scheduler a higher flexibility in exploiting the transmission opportunities and also higher probability to detect good time slots. In contrast, when $|K|$ is small, the scheduler needs higher accuracy to detect transmission opportunities. However, we can also observe that the performance improvement saturates when further increasing the number of users, which indicates a limit in the improvement that can be obtained by using this approach.

D. Symmetric traffic scenarios

In this section, we report results for the case in which $|M| = 3$ tenants have symmetric traffic (same amount of users per service type).

Due to the symmetry among tenants, we observe an equivalent resource and cost distribution, as shown in Fig. 6. This proves that, as desired, in symmetric cases our model behaves perfectly fair among tenants. Furthermore, Fig. 7 reports the average utility per tenant per service and, as above, we observe that there is a symmetric behavior among tenants, but different prioritization among slices, i.e., services. Namely, due to the utility based prioritization, when the system does not have sufficient resources to fully satisfy all of them, the elastic users are penalized and reach lower utility compared to the other services. Moreover, both inelastic and M2M services



(a) Average resource distribution. (b) Average total cost.

Fig. 6. Average resource distribution and average total cost per tenant for $|K = 12|$.

are achieving an average utility less than 1 due to the utility function used (c.f. Fig. 2(b)). Namely, after reaching the utility value of U_2, all the services have the same slope, that provides fairness between elastic service and the rest of the services.

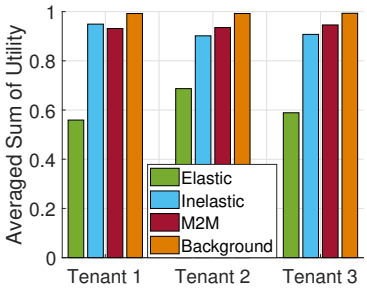
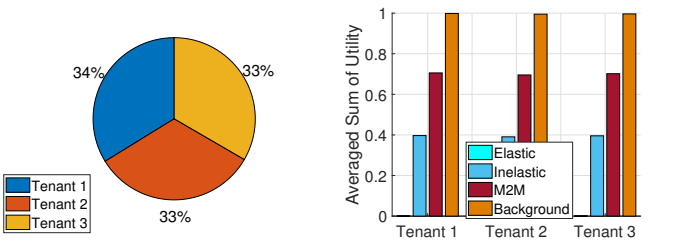


Fig. 7. Average utility per service per tenant.

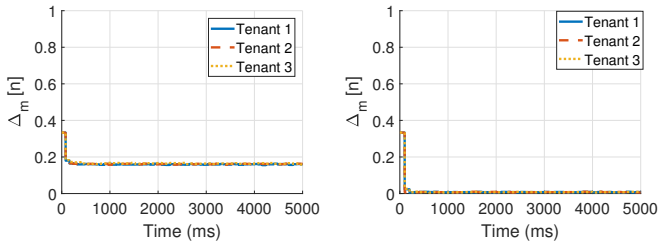
Now, we show how the proposed framework reacts to load changes. In particular, we increase the number of users of each tenant to $|K_m| = 16$ users (i.e. total number of users $|K| = 48$), while keeping fixed the system capacity and utility function parameters. As shown in Fig. 8, despite the strong competition for resources, fairness among tenants is still achieved. Moreover, in Fig. 8(b) even more emphasis is shown on the prioritization given to different services. As expected, the elastic traffic, which has the lowest priority, is being affected mostly from the resource scarcity. In contrast, such prioritization guarantees that the emergency and low-rate-delay-sensitive M2M traffic (i.e. defined in Section II-A4) can achieve the service expectations even in such an extreme scenario (which is proved by the fact that for this service type at least utility equal to U_2 is achieved).

Another interesting effect of the increasing load is shown in Fig. 9. We observe that resource scarcity affects tenant's convenience to trade resources. As a matter of fact, when $|K| = 12$, Δ_m converges to a non-zero value, guaranteeing a certain level of flexibility in resource allocations (c.f. Fig. 9(a)). This flexibility allows the scheduler, and tenants, to adopt an opportunistic behavior thus enhancing cost and resource efficiency. On the other hand, when load drastically increases (c.f. Fig. 9(b)), the inability of serving elastic users pushes $\Delta_m = 0$, $\forall m \in M$ thus reducing the flexibility of sharing.



(a) Average resource distribution. (b) Average utility per service.

Fig. 8. Average resource distribution and average utility per service per tenant for $|K| = 48$.

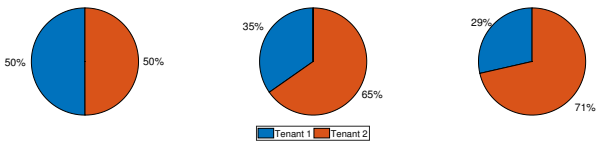


(a) Δ_m variation for $|K| = 12$ (b) Δ_m variation for $|K| = 48$

Fig. 9. Adaptation of Δ_m to the increasing traffic.

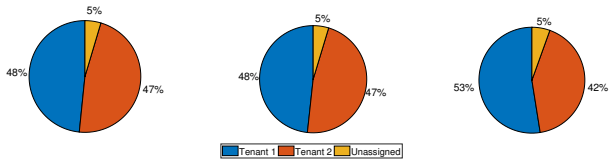
E. Impact of time window

In this section, we analyze the impact of time window differentiation among tenants. Fig. 10 and Fig. 11 shows the effects of varying the time window length on the resource distribution between $|M| = 2$ tenants in case of resource scarcity and resource surplus, respectively.



(a) $W_1 = 80,\ W_2 = 80$ (b) $W_1 = 80,\ W_2 = 40$(c) $W_1 = 80,\ W_2 = 20$

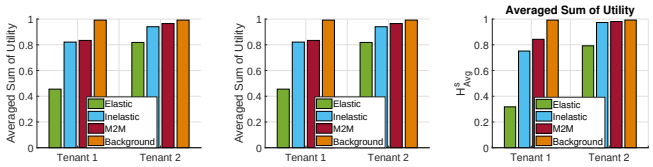
Fig. 10. Effects of window differentiation on average resource distribution per tenant in resource scarcity scenario.



(a) $W_1 = 80,\ W_2 = 80$ (b) $W_1 = 80,\ W_2 = 40$(c) $W_1 = 80,\ W_2 = 20$

Fig. 11. Effects of window differentiation on average resource distribution per tenant in resource surplus scenario.

Generally speaking, smaller time windows indicate that the tenant's requirements need to be satisfied with higher frequency (i.e. within a shorter time frame). Therefore, due to the more stringent delay constraints, the InP has to prioritize the tenant with smaller W_m in order to be able to satisfy its utility target. From one side, this prioritization does not affect the resource distribution among the two tenants, whenever there are sufficient resources to satisfy all the tenants, i.e. resource surplus (cf. Fig. 11). On the other side, however, in case of resource scarcity (cf. Fig. 10), the priority given to the tenant with smaller time window (Tenant 2 in this example) causes an imbalance in the resource allocation, which increases proportionally to the difference between the window lengths. Since choosing a smaller time window corresponds to potentially getting more resources, the selection of this parameter has to be monitored by the InP or regulatory body.



(a) $W_1 = 80,\ W_2 = 40$ (b) $W_1 = 80,\ W_2 = 40$(c) $W_1 = 80,\ W_2 = 20$

Fig. 12. Effects of window differentiation on average utility per service in resource scarcity scenario.

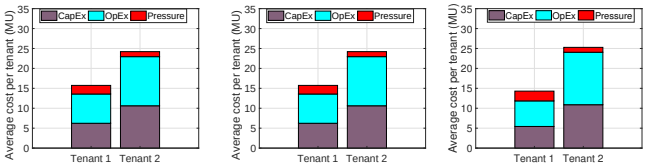
Fig. 12 shows the effects of time window differentiation on the average utility per tenant per service in case of resource scarcity. As expected, the tenant with smaller time window receives a higher priority in the scheduler, which corresponds to a higher average utility with respect to the one achieved by the other tenant. Furthermore, results show that the service which is most penalized by the prioritization is the elastic one. In contrast, non-elastic services are preserved by the utility based prioritization (i.e. the slopes of the utility functions shown in Fig. 2(b)) and experience only marginal decrease in the achieved utility. On the other side, the tenant with smaller time window perceive an increase in utility for all the services, critical as well as elastic. Note that this has the negative effective of reducing the efficiency in resource usage, since more resources are assigned to one of the two tenants, independent of the channel conditions of its users.

Finally, Fig. 13 and Fig. 14 reports the economic effects of window differentiation. Fig. 13 shows that, according with the resource distribution, the tenant with smaller W_m pays a higher cost, in average, while the tenant with larger time window length decreases the total costs. On the other hand, Fig. 14 reveals that the tenants actual average cost per bps/Hz is similar for all cases. This confirms that the costs paid by the tenants is actually proportional to the resources they get.

F. Adaptation to changes in traffic mix

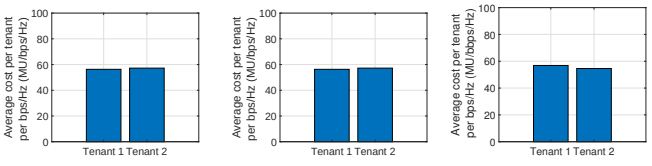
In [15], we analyze the ability of the proposed model to adapt to changes of the wireless environment, and conclude that, in case of resource scarcity, such changes mainly affect the elastic services and our model is able to converge to a new optimal state adapting to the new conditions. Differently, here we consider a resource surplus scenario, and analyze the reaction time and the effects of varying the traffic mix of the tenants.

Fig. 15 shows the adaptation to the changes in the traffic mix. In particular, we assume that till $n = 1920$, the two



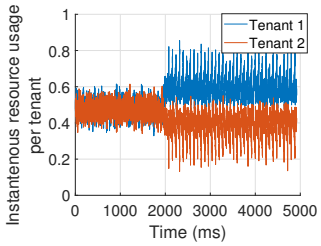
(a) $W_1 = 80,\ W_2 = 40$ (b) $W_1 = 80,\ W_2 = 40$ (c) $W_1 = 80,\ W_2 = 20$

Fig. 13. Effects of window differentiation on average total cost per tenant in resource scarcity scenario.

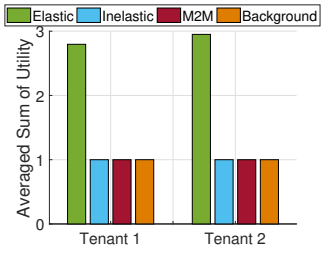


(a) $W_1 = 80,\ W_2 = 40$ (b) $W_1 = 80,\ W_2 = 40$ (c) $W_1 = 80,\ W_2 = 20$

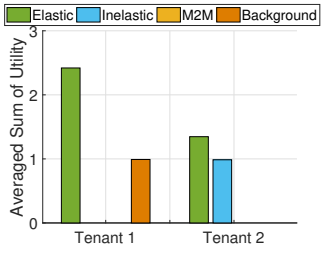
Fig. 14. Effects of window differentiation on average total cost per bps/Hz in resource scarcity scenario.



(a) Variation of resource usage per tenant over time.



(b) Average sum of utility before the change of traffic mix.



(c) Average sum of utility after the change of traffic mix.

Fig. 15. Adaptation to the variations in the traffic mixture.

tenants have symmetric traffic, i.e. 1 user per service type and a total of $|K| = 8$ users. At $n = 1920$, the traffic mix of the tenants changes as follows: the first tenant retains only non-critical services (i.e. it has 2 users with elastic services and 2 users with background services) while the second tenant specializes on critical services (i.e. 3 users with inelastic services and 1 user with elastic service). In Fig. 15(a), we observe, between $n = 1920$ and $n = 2000$, a gradual change in the instantaneous assigned resources. After at least one renegotiation interval, the tenant's sharing parameters are updated, and this leads to a converge of the resource assignment. In Fig. 15(b) and Fig. 15(c), the average utility per service per tenant is shown before and after the traffic mix change, respectively. Note that, after the change, the elastic services achieve in average a smaller utility. This is due to the fact that the number of users per service increases, which means that the resources requested by the non-elastic service (background for tenant 1 and inelastic for tenant 2) also increase.

G. Service specialized tenants

This section investigates the effects of service specialization on the proposed model. More specifically, we analyze the coexistence of four tenants with only one service type and one tenant with multiple service types. This also helps us addressing the question on whether our framework incentives tenants to enter the sharing market as specialized tenants or, in contrast, it is neutral to this choice. Therefore, we consider the scenario with $|M| = 2$ tenants, where the first tenant enters the market as virtually 4 different tenants (one per type). Also, we assume $|K| = 16$ users in total (2 users per service per tenant).

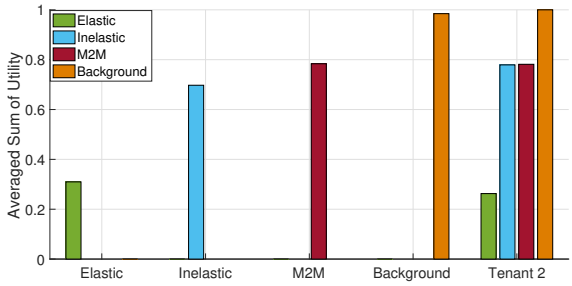


Fig. 16. The average utility per services per tenant

Fig. 16 clearly shows that entering the market as specialized tenant does not provide any advantages in terms of average achieved utility. Furthermore, Fig. 17 shows that a symmetry between the specialized tenants and the tenant with multiple services also exists in terms of the total average costs. Finally, Fig. 18 reports the resource distribution among tenants, which clearly indicates that also resources are split equally (i.e. each tenant gets approximately half of the available resources).

We can conclude that the propose framework and corresponding pricing mechanism are neutral to service specialization. Also, service prioritization (defined in Section II-A) is preserved and fairness is achieved in terms of both resource allocation and costs.

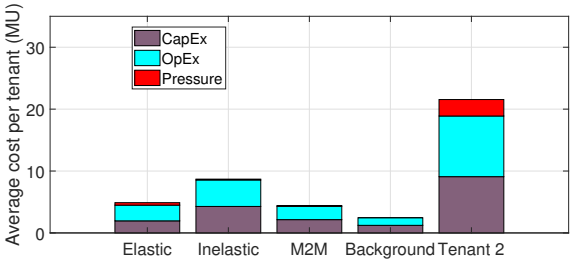


Fig. 17. The average total cost per service per tenant

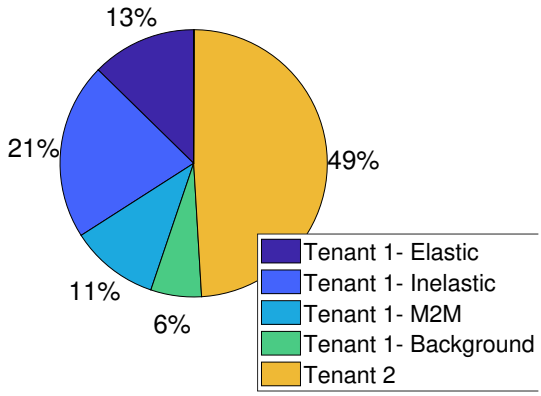
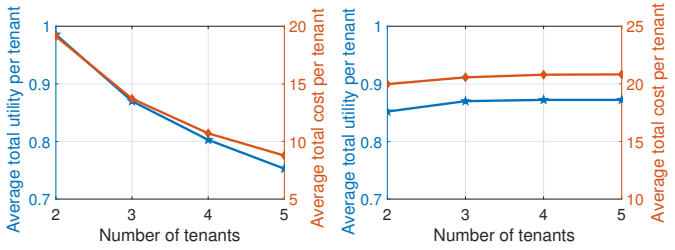


Fig. 18. Effects of service specialization on resource distribution.



(a) Short term (b) Long term

Fig. 19. Effects of increasing number of tenants on the average utility and average costs per tenant.

H. Costs and utility in different sharing scenarios

In this subsection, the effects of the number of tenants $|M|$ on the average cost per tenant and the average utility per tenant per service are investigated. The analysis is conducted considering two different time scales, i.e. short term and long term. In the short term, we assume that the infrastructure provider cannot react to the increase in the number of tenants $|M|$ (and thus users $|K|$), e.g., expanding the available capacity. In contrast, in the long term the capacity is scaled according to the demand.

Fig. 19(a) shows the result for the short term analysis, where the capacity is kept fixed while increasing $|M|$. On the other hand, Fig. 19(b) reports the result for the long term assumption, where the capacity is proportionally increased with $|M|$. Namely, we assume that the increase in capacity is achieved by the infrastructure provider increasing the total bandwidth. Results show that in the short term, Fig. 19(a), as expected, increasing the number of tenants causes a resource scarcity and leads to a decrease of the average utility per tenant. On the other hand, as shown in Fig. 19(a), the increase in $|M|$ also causes a decrease of the individual costs of tenants. In contrast, when considering a longer time scale (in the order of months), the infrastructure provider can react to the changes in $|M|$ and adjust the available capacity according to the needs. In this case, as depicted in Fig. 19(b), the average achieved utility and the average cost are not a function of $|M|$ (i.e. are almost constant when varying $|M|$).

Therefore, from one side we can conclude that, in the long term, if the InP is able to expand the network capacity according to the tenants' needs, the proposed platform provides a sustainable resource sharing even when increasing $|M|$. On the other side, in the short term, we cannot draw any conclusion only looking at Fig. 19(a), since a decrease of the average utility could be compensated by a decreasing in terms of cost (and hence price for the users). To evaluate the tradeoff between utility and cost (price), we use the concept of acceptance probability presented in [25]. In particular, the authors propose to model the acceptance probability as:

$$A_k(p, U_k) = 1 - \exp(-Cp^{-\epsilon}U_k^{\mu}), \tag{8}$$

which basically corresponds to the likelihood of user k to accept a service with price p and a corresponding utility U_k, where μ and ϵ are microeconomic parameters and C is a constant (that we set to the same values suggested in [25]).

To assess the sustainability of the sharing platform, we assume that each tenant aims to keep its profit constant, regardless of the number of tenants, which means that a variation of the costs directly affects the prices (that are computed as the sum of the costs and the profit). Therefore, increasing $|M|$ is accepted by the tenants, if the market share (i.e. the number of users) of each tenant is not decreasing, meaning that the acceptance probability ($A_k(p, U_k)$) should be a non-decreasing function of $|M|$.

By using (15), the condition above can be written, for two generic values $|M_1| \leq |M_2|$, as:

$$A_{k,M_1}(p_{M_1}, U_{k,M_1}) \leq A_{k,M_2}(p_{M_2}, U_{k,M_2}), \tag{9}$$

where

$$A_{k,M_1}(p_{M_1}, U_{k,M_1}) = 1 - \exp(-Cp_{M_1}^{-\epsilon}U_{k,M_1}^{\mu}),$$

$$A_{k,M_2}(p_{M_2}, U_{k,M_2}) = 1 - \exp(-Cp_{M_2}^{-\epsilon}U_{k,M_2}^{\mu}).$$

Assuming that the parameters μ, ϵ, and C are the same for both M_1 and M_2, (16) can be written as

$$\left(\frac{U_{k,M_1}}{U_{k,M_2}}\right)^{\mu} \leq \left(\frac{p_{M_1}}{p_{M_2}}\right)^{\epsilon}. \tag{10}$$

Satisfying (17) means that the variation in the average utility is accepted by the users since it is compensated by the decrease of the service price. In this case, the acceptance probability of $k \in K$ is a non-decreasing function of $|M|$.

Considering the same scenario of Fig. 19, Table IV reports the numerical values for (17). As one can observe, the inequality is always satisfied, which means that the users are paying less for their utility, and they are still willing to accept the service. Therefore, we can conclude that our proposed model provides a cost efficient and sustainable model even in the short term.

A further insight is given in Table V, where Eq. (17) is evaluated for all the slice types (where 'yes' means that the Eq. (17) holds). In this case, we can see that, by increasing the number of tenants from $|M| = 4$ to $|M| = 5$, the acceptance probability of the elastic users decreases, whereas always increases for non-elastic services. This means that the tenants have a risk of losing some of the elastic traffic.

TABLE IV
VARIATION OF AVERAGE UTILITY AND TOTAL COSTS PER TENANT WITH THE NUMBER OF TENANTS IN SHORT TERM.

$\lvert M_1\rvert \rightarrow \lvert M_2\rvert$	$\left(\frac{U_{k,M_1}}{U_{k,M_2}}\right)^{\mu}$	$\left(\frac{p_{M_1}}{p_{M_2}}\right)^{\epsilon}$
$2 \rightarrow 3$	1,2834	3,7822
$3 \rightarrow 4$	1,1744	2,6893
$4 \rightarrow 5$	1,1372	2,2142

TABLE V
EVALUATION OF THE USERS' ACCEPTANCE PROBABILITY FOR ALL SLICE TYPES. WE USE 'YES' TO INDICATE THAT EQ. (17) HOLDS, 'NO' OTHERWISE.

$\lvert M_1\rvert \rightarrow \lvert M_2\rvert$	Elastic	Inelastic	M2M	Background
$2 \rightarrow 3$	Yes	Yes	Yes	Yes
$3 \rightarrow 4$	Yes	Yes	Yes	Yes
$4 \rightarrow 5$	No	Yes	Yes	Yes

The decrease in elastic services acceptance probability can be handled by an accurate and timely capacity expansion. The proposed pressure cost allows the infrastructure provider to accurately estimate the capacity needs and the expansion time. Even though increasing $|M|$ leads to lower utilities, since the collected pressure cost proportionally increases with the utility decrease, higher $|M|$ also implies faster capacity expansions.

V. CONCLUSION

We have shown that dynamic network slicing offers an efficient way of exploiting variable traffic and channel conditions to share resources among tenants with different characteristics and strategies. Our proposed scheme defines a new platform where tenants can acquire resources over a short time scale, negotiating through a set of network and economic parameters. Numerical results show that the proposed approach provides fairness among both tenants and services and can improve the efficiency of resource allocation up to 40% by exploiting simple prediction mechanisms. Despite the tenants share a common infrastructure, results have also demonstrated that it is possible for them to differentiate their services by tuning model parameters. We have also shown that the pricing model can allocate economic resources for capacity expansion and that this is crucial to keep infrastructure sharing convenient for the tenants.

ACKNOWLEDGMENT

This work is funded by the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 643002.

REFERENCES

[1] Cisco, "The zettabyte era: trends and analysis," 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.pdf

[2] W. Lemstra, "Leadership with 5G in Europe: Two contrasting images of the future, with policy and regulatory implications," *Telecommunications Policy*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0308596118300491

[3] China Mobile Communications Corporation, Huawei Technologies, Deutsche Telekom, and Volkswagen, "5G service-guaranteed network slicing white paper," 2017.

[4] OECD, "Wireless market structures and network sharing," 2014. [Online]. Available: http://dx.doi.org/10.1787/5jxt46dzl9r2-en

[5] N. C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 954 – 1001, 2017.

[6] D. Zhang, Z. Chang, and T. Hamalainen, "Reverse combinatorial auction based resource allocation in heterogeneous software defined network with infrastructure sharing," in *IEEE Vehicular Technology Conference (VTC Spring)*, 2016.

[7] P. Cramton and L. Doyle, "An open access wireless market supporting competition, public safety, and universal service," 2016. [Online]. Available: http://www.cramton.umd.edu/papers2015-2019/cramton-doyle-open-access-wireless-market.pdf

[8] O. U. Akgül, I. Malanchini, V. Suryaprakash, and A. Capone, "Dynamic resource allocation and pricing for shared radio access infrastructrue," in *IEEE International Conference on Communications (ICC)*, 2017.

[9] G. S. Kasbekar, S. Sarkar, K. Kar, P. K. Muthuswamy, and A. Gupta, "Dynamic contract trading in spectrum markets," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2856 – 2862, 2014.

[10] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462 – 476, 2016.

[11] X. Ting, P. Zhiwen, L. Nan, and Y. Xiaohu, "Inter-operator resource sharing based on network virtualization," in *International conference on Wireless Communication Signal Processing (WCSP)*, 2015, pp. 1–6.

[12] M. I. Kamel, L. B. Le, and A. Girard, "LTE wireless network virtualization: Dynamic slicing via flexible scheduling," in *IEEE Vehicular Technology Conference (VTC Fall)*, Sept 2014, pp. 1–5.

[13] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management & prioritization in 5G mobile systems," in *22 European Wireless 2016; 22th European Wireless Conference*. IEEE, May 2016, pp. 1–6.

[14] A. Gran, S.-C. Lin, and I. F. Akyildiz, "Towards wireless infrastructure-as-a-service (WIaaS) for 5G software-defined cellular systems," in *2017 IEEE International Conference on Communications (ICC)*.

[15] O. U. Akgül, I. Malanchini, V. Suryaprakash, and A. Capone, "Service-aware network slice trading in a shared multi-tenant infrastructure," in *IEEE Global Communications Conference (GLOBECOM)*, 2017.

[16] J. Pérez-Romero, O. Sallent, S. Ferrús, and R. Agustí, "Admission control for multi-tenant radio access networks," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017.

[17] R. Berry, M. Honig, T. Nguyen, V. Subramanian, H. Zhou, and R. Vohra, "On the nature of revenue-sharing contracts to incentivize spectrum-sharing," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2013.

[18] L. Cano, A. Capone, G. Carello, M. Cesana, and M. Passacantando, "Cooperative infrastructure and spectrum sharing in heterogeneous mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2617 – 2629, 2016.

[19] L. Zheng, J. Chen, C. Joe-Wong, C. W. Tan, and M. Chiang, "An economic analysis of wireless network infrastructure sharing," in *15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2017.

[20] I. Malanchini and M. Gruber, "How operators can differentiate through policies when sharing small cells," in *IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015.

[21] A. P. Avramova and V. B. Iversen, "Radio access sharing strategies for multiple operators in cellular networks," in *IEEE International Conference on Communication Workshop*, June 2015, pp. 1113–1118.

[22] J. S. Panchal, R. Yates, and M. M. Buddhikot, "Mobile network resource sharing options: Performance comparisons," *IEEE Transactions on Wireless Communications,*, vol. 12, no. 9, pp. 4470–4482, 2013.

[23] I. Malanchini, S. Valentin, and O. Aydin, "Wireless resource sharing for multiple operators: Generalization, fairness, and the value of prediction," *Computer Networks*, vol. 100, pp. 110 – 123, 2016.

[24] Gurobi Optimization Inc., "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

[25] L. Badia, M. Lindstrom, J. Zander, and M. Zorzi, "Demand and pricing effects on the radio resource allocation of multimedia communication systems," in *IEEE Global Telecommunications Conference, GLOBECOM '03*, vol. 7, Dec 2003, pp. 4116–4121 vol.7.

Özgür Umut Akgül holds a M.Sc. in Computer Engineering from Istanbul Technical University in Turkey. He is currently pursuing the Ph.D. degree with Department of Electronics and Information, Politecnico di Milano in Italy. He is also involved in the EU H2020 ACT5G project. His main research interests are mostly related to techno-economic modeling and analysis of network slicing in multi-tenant networks based on game theoretical models.

Ilaria Malanchini Ilaria Malanchini is a Senior Research Engineer and has been with Bell Labs Stuttgart since 2012. She received B.S. and M.S. degrees in telecommunications engineering from Politecnico di Milano, Italy, in 2005 and 2007, respectively, and a Ph.D. in electrical engineering from Drexel University, Philadelphia, and Politecnico di Milano in 2011. Ilaria was awarded the Meucci-Marconi Award and the Chorafas Foundation Prize for her Master and PhD thesis, respectively. She published more than 25 peer reviewed journal and conference papers and has more than 10 granted or filed patents. Her research interests focus on optimization models, mathematical programming, game theory, and machine learning, with the application of these techniques to wireless network problems such as wireless resource allocation, anticipatory network optimization, infrastructure and resource sharing, and network slicing.

Antonio Capone is currently a Full Professor with the Politecnico di Milano (Technical University of Milan), where he is also the Director of the ANTLab. His expertise is on networking and his main research activities include radio resource management in wireless networks, traffic management in software defined networks, network planning, and optimization. On these topics, he has published over 250 peer-reviewed. He was an Editor of the ACM/IEEE TRANSACTIONS ON NETWORKING from 2010 to 2014. He serves in the TPCs of major conferences in networking, he is an Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING, *Computer Networks*, and *Computer Communications*.

Anticipatory Resource Allocation and Trading in a Sliced Network

Özgür Umut Akgül*†, Ilaria Malanchini†, and Antonio Capone*
*Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milano, Italy
Email: oezguerumut.akguel, antonio.capone@polimi.it
†Nokia Bell Labs, Stuttgart, Germany
Email: ilaria.malanchini@nokia-bell-labs.com

***Abstract*—The inefficiency of the real-time network slicing and resource allocation algorithms are widely researched in literature. In a shared network, this inefficiency becomes more critical due to the economical impacts of the network decisions. Extending our previous reactive network slicing model, in this paper, we propose an anticipatory network slicing framework that can exploit the prediction information in order to increase the spectral efficiency. The proposed model provides an efficient and sustainable trading framework for all parties, namely, a higher average achieved utility for a lower cost for the tenants and enhanced capacity management for infrastructure provider.**

I. Introduction

The saturation in the consumer market and the decreasing profitability of the network provisioning make the cost efficiency as the dominant factor in the transition to the next generation wireless networks. As the cost reduction becomes a key challenge, in parallel, the network operators begin searching for alternative revenue sources. The recent researches reveal that serving to the specialized industry segments can boost the revenues as high as 36% [1]. On the other hand, in order to accommodate these new markets, the prevalent network infrastructure has to support a multitude of vertical applications with diverse requirements that pushes the technical limits. Moreover, a key aspect while increasing these technical capabilities is to preserve the existing infrastructure as much as possible [2].

A relatively new idea to solve the heterogeneous requirements of the different services is to provide both storage and processing resources along with the respective network resources [3]. Consequently, this approach lead to logically slicing the network resources and optimizing each slice based on the requirements of the respective service. Thus, through network slicing, the network resources (in terms of spectrum, network functions and computational power) can be customized to achieve the maximum quality of service (QoS) per service type. [4]

A. Related Works

The simplest way of slicing a network is statically dividing the network resources according to the slice templates and the stedy state conditions, whose advantanges are analysed in [5]. Contrastively, the lack of flexibility in static slicing often results in cases where the operators lose their capability to compensate the variations in the channel and the traffic demand which eventually leads lower spectral efficiency and higher costs. On the other hand, by exploiting the transient conditions, the performance of the slicing can be enhanced. In [6], a dynamic slicing and trading framework has been proposed in multi-tenant networks, where the tenants can update their shares in short time scales (i.e. in the order of millisecond) and follow the most efficient (both in terms of cost and performance) resource allocations for themselves. The concept of network slicing in order to accommodate multiple services leads to the idea of serving multiple tenants using the same infrastructure in order to decrease total cost, namely *infrastructure sharing* [7]. Despite its well investigated structure, the infrastructure sharing literature is mostly based on well defined service level agreements (SLAs) that covers long time intervals (e.g. in scale of months). Following the expectation of cost reduction, majority of the works in literature focus on either the economical impacts of sharing, such as [8] [9], or the technological enablers of sharing, like [10] [11]. However, the changing landscape of mobile networks requires a through analysis of techno-economic aspects of sharing.

Furthermore, it is a well known fact that the real time scheduling is always wasteful due to its local focus. The full potential of network slicing can be attained by anticipating the evolution of the system dynamics using past observations and the current state of the network. Although the anticipatory networking has been well-investigated, its applicability to the field of network slicing and slice trading is novel. [12] investigates the advantages of anticipatory network slicing considering static SLAs and a prediction algorithm with high accuracy. However, due to time complexity, such a complicated approach cannot be used in real time algorithms. In [6], we integrate a simple prediction algorithm to the network slicing problem in order to overcome the inefficiency of the real time resource scheduler. [2] outlines the major prediction methods to various networking problems and gives some insight about the optimization techniques in anticipatory networking. Among the methods presented in [2], auto-regressive integrated moving average (ARIMA) and the feed forward neural networks (FFNN) are the most eligible candidates for our problem of real time network slicing and slice trading due to their accuracy level and the time complexity. Despite some studies, e.g. [13], investigate the impact of prediction

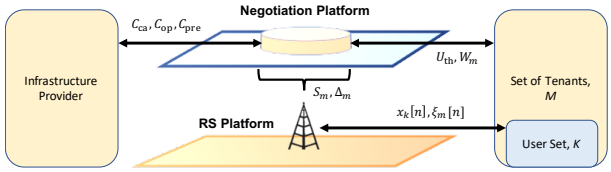
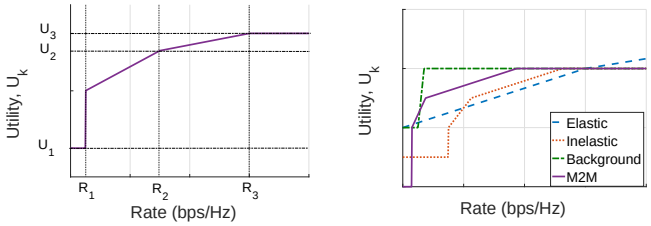


Fig. 1. Proposed negotiation and resource scheduling (RS) platform.



(a) Generic utility function. (b) Exemplary utility functions.

Fig. 2. Generic utility function (left) and exemplary utilities (right).

errors on resource allocation, the economic implications of the prediction errors on a shared network are still unclear.

B. Contributions and Organization

To summarize, the main contributions of this work are as follows:

- An overview of the anticipation techniques and an investigation of their suitability to our model,
- Techno-economic analysis on the effects of anticipation in a sliced and shared network,
- A novel filtering approach designed in order to exploit the advantages of prediction when the prediction accuracy is high and to filter out the disadvantages when the prediction accuracy is low.

The remainder of the paper is organized as follows. Following the outline of the considered system model and the considered prediction models in Section II, Section III presents the proposed anticipatory network slicing and trading framework and details the exploitation of the predicted data. The numerical analysis of the proposed model is presented in Section IV and Section V concludes the paper.

II. System Model

As an extension of our previous work [6], we have considered the dynamic negotiation platform given in Fig. 1 which summarizes the interaction between the key stakeholders in our model, namely an infrastructure provider, the set of tenants (M) and the set of users (K). Lowercase letters k and m are used to indicate a specific user and tenant respectively. For the sake of simplicity, we assumed that each user is related to one service type and the total amount of users, $|K|$, is distributed equivalently among the tenants and shown as $|K_m|$ where $\cup_{m \in M} K_m = K$. Following the general approach in resource allocation literature, the simulation horizon, N, is discretized and divided into time slots that are indexed with n.

The SLAs between tenants and the infrastructure provider that controls the resource sharing are mapped to the model using three parameters, i.e. S_m, Δ_m and W_m. The guaranteed resource share, represented by $S_m \in (0, 1)$, shows the average resource share that the tenant m receives in average. In order to exploit the dynamic nature of the wireless environment, the tenants define a maximum deviation from SLA, represented by Δ_m, for a given time window W_m. Therefore, the delay constraint per tenant is indirectly integrated using W_m. The tenant specific sharing parameters, S_m and Δ_m, are updated at each renegotiation interval (RI).

As given in Fig. 1, tenants set their utility targets, U_{th}, and their respective budgets, B_m. The total cost of wireless resources is modeled as the summation of the operational costs (C_{op}), capital costs (C_{ca}) and the pressure cost (C_{pre}) which is used to regularize the resource consumption and to collect the necessary revenue in order to expand the network. More specifically, in line with any demand based market, the pressure cost scales the unit cost according to the instantaneous demand; if there are insufficient resources to satisfy all the users, pressure cost becomes greater than zero, making it more expensive to buy the resources. Otherwise, namely if there are sufficient resources to satisfy all the users, then it will be zero. Thus, the accumulated pressure cost also indicates the necessary additional capacity to fully satisfy all the users.

Based on the decided sharing parameters per tenant and the achievable rate of each user k, $r_k[n]$, the real time scheduler assigns resources per user $x_k[n]$. The actual achieved rate of k at any time slot n is calculated as $r_k[n]x_k[n]$ and used in order to calculate the utility of each user $U_k[n]$. We assumed that each user has an equivalent weight in the tenant's utility target, indicating that the total achieved utility of the operator is $\sum_{k \in K_m} \frac{U_k[n]}{K_m}$.

A. Utility Functions

As detailed in [6], the utility of each user is measured based on the average achieved rate of the respective user in the respective time window. In order to model the utility, we designed a piece-wise linear function (c.f. Fig. 2(a)) that are determined using six parameters, i.e. R_1, R_2, R_3, U_1, U_2 and U_3. If the actual achieved rate is smaller than the minimum rate requirement, R_1, it is considered to be not active and the service produces a utility value of $U_1 \leq 0$. R_2 indicates the necessary rate in order to consider the service to receive a standard quality and produces the utility of U_2. The region between $R_1 - R_2$ is designed to have a steep slope due to the high visibility of the enhancement in the achieved QoS. Finally, R_3 indicates the achieved rate that produces the maximum utility, U_3. Note that any further increase in the achieved rate after R_3 does not effect the utility.

Simmilar to our previous work, [6], the heterogeneity in envisioned 5G services is captured by considering four major service types, i.e. *elastic services, inelastic services, machine to machine services (M2M)* and *background services* and their

utility functions are designed to be as given in Fig. 2(b). As the name suggests, elastic services do not have strict delay or rate constraints, thus $R_1 = 0,\ U_1 = 0$. Moreover, it is assumed that they do not have any upper limits for their rate expectations, meaning $R_3 \to \infty$. Inelastic and M2M services assume to contain all three regions indicated by $R_1,\ R_2$ and R_3. For both of these services, U_1 is assumed to be lower than zero, indicating that not serving these services would even further decrease the total utility. For M2M service, it is assumed that each piece-wise linear region captures a different type of device group, namely, emergency, low-rate-delay-sensitive and rate sensitive. Finally, the background service is assumed to need a very low rate and reaches directly to U_3 when it is satisfied, consequently, $R_2 = R_3$ and $U_1 = 0$.

B. Auto Regressive Integrated Moving Average

Auto-Regressive Integrated Moving Average (ARIMA) is widely applied in the field of wireless networks, due to its simplicity, locality and relatively high performance. Unlike most of the deep learning mechanisms (e.g. [14]), ARIMA does not require a long history of the observed function's outputs [13]. Through a relatively smaller set of instances, ARIMA determines a few parameters that would represent the function best and anticipates the possible behaviors of the observed function in the upcoming instances.

ARIMA contains five major parameters, namely, the prediction window W_p, learning window W_l, the number of auto-regressive terms p, the number of nonseasonal differences d and finally the number of moving average terms q. In the first step of the algorithm, the past observations within W_l are used in order to estimate the correlation within the past and current time slots and the (p, d, q) parameters are chosen according to these correlations. On the second phase of the algorithm, using the converged ARIMA parameters (p, d, q), the upcoming values of the series are predicted. Note that this values are not only time series depended but also time dependent, therefore, this analysis over the parameters are require to be renewed at every renegotiation interval.

C. Feed Forward Neural Networks

The feed forward neural networks (FFNN) or feed forward multilayer perceptron is widely applied in the time series prediction due to its high precision and capability to approximate complex functions. Unlike ARIMA, FFNN requires a relatively long learning period, during which it learns the correlation among the instances and updates the weights of the neural network accordingly. FFNN is defined using four major parameters, namely, the learning window (W_l), prediction window (W_p), the number of nodes in the hidden layer (D_N), the number of hidden layers and finally the number of delays. The two layer FFNN is considered to be the general approximator [2], thus in this study we limit the number of hidden layers to one.

In the implemented structure, the FFNN predicts the $D_N + 1^{\text{th}}$ instance based on the input of D_N. Afterwards, the D_N window is shifted one time, resulting in a delay set that would also include the newly predicted instance and this new delay set is used to predict $D_N + 2^{\text{nd}}$ instance. The FFNN continues shifting delay set and making the prediction for the next instance until it reaches W_p.

$$\min_{x_k[n], S_m, \Delta_m} \quad \sum_{m \in M} \xi_m[n] \tag{1a}$$

$$\text{s.t.} \ \ U_{th,m} - \sum_{k \in K_m} U_k(R_k[n]) \leq \xi_m, \ \forall m \in M, \tag{1b}$$

$$\epsilon_m[n] = \left(\frac{1}{(a_m + 1)} \sum_{i=n-a_m}^{n} \sum_{k \in K_m} x_k[i] \right) - S_m, \ \forall m \in M, \tag{1c}$$

$$|\epsilon_m[n]| \leq \Delta_m, \quad \forall m \in M, \quad \forall n \in N, \tag{1d}$$

$$\sum_{i=n-a_m}^{n} (S_m(C_{\text{ca}} + C_{\text{op}}) + \epsilon_m[i] C_{\text{op}} + f_{\text{pre}}(C_{\text{pre}}, \ \xi_m)) \leq B_m(a_m + 1), \forall m \in M, \tag{1e}$$

$$0 \leq \Delta_m \leq \frac{1}{a_m + 1} \sum_{i=n-a_m}^{n} \sum_{k \in K_{m,elastic}} x_k[i], \quad \forall m \in M, \tag{1f}$$

$$\sum_{k \in K} x_k[n] \leq 1, \quad x_k[n] \geq 0, \quad \forall k \in K, \tag{1g}$$

$$\sum_{m \in M} S_m \leq 1 \ , \quad S_m \geq 0, \quad \forall m \in M, \tag{1h}$$

III. Anticipatory Resource Scheduling Problem And Analysis

A. Mathematical programming formulation

The proposed mathematical programming formulation in (1a)-(1h) distributes the network resources in real time while enabling a market driven pricing mechanism according to the QoS requirements of services, achievable rates of the users, tenants' budgets and the utility goals. The continuous objective function, (1a) minimizes the total gap of the tenants ξ_m, namely the difference between tenant's expected utility and the actually obtained utility, and maximizes the resource efficiency.

(1b) sets the gap definition per tenant, namely, the difference between the expected utility and the achieved utility, $U_k(R_k[n])$, that is calculated based on the previously defined utility functions. During the calculation of the achieved utility at any time slot n, we use the average achieved rate up to the current time slot through the expanding time window, $a_m + 1$, where $a_m \equiv n - 1 \mod W_m$. Thus the average achieved rate through the expanding time window is calculated using

$$R_k[n] = \frac{1}{(1 + a_m)} \left(\sum_{i=n-a_m}^{n} x_k[i] r_k[i] \right). \tag{2}$$

The maximum instantaneous deviation per tenant, i.e. defined in (1c), is limited by Δ_m in (1d).

(1e) reflects the economic implications of technical decisions. The left-hand-side of (1e) calculates the total cost of a tenant, i.e. the summation of capital expenditures (CapEx), operational expenditures (OpEx), and pressure cost. The first term, i.e. $S_m(C_{\text{Ca}} + C_{\text{Op}})$, reflects the fact that tenants are required to pay both the CapEx and the OpEx in parallel to their resource share in the network. On the other hand, a flexibility has been provided by the second term, i.e. $\epsilon_m[i]C_{\text{op}}$, that is scaled according to the tenants actual resource usage. More specifically, this term indicates that the tenants are only obliged to pay OpEx for the resources that are obtained from the resource pool. This second term can also provides an economic incentive to share resources rather than absolute ownership. The third term, i.e. $f_{\text{pre}}(C_{\text{pre}}, \xi_m)$, is the pressure cost. By its definition, the pressure cost is a tool for the infrastructure provider to regulate the price of the resources. In this paper, we assumed that the infrastructure provider has no profit constraints and reinvests all the obtained revenue. Therefore, in order to calculate this cost, we used a multiplication of operators' gap and the unit pressure cost, i.e. $\xi_m C_{\text{pre}}$. However, since ξ_m is a dynamic entity, in order to give a level of predictability in pricing, the average gap in the previous renegotiation interval is used to calculate the pressure cost in the upcoming renegotiation interval. The right-hand-side of (1e) is the budget of the tenant, B_m. Note that the B_m is defined per time slot. Therefore, by multiplying it with the expanding time window length, $a_m + 1$, and summing the total costs on the left-hand-side, the tenant is given a chance to use the unused budget from the previous time slots in the forthcoming time slots within the same time window.

(1f) sets the upper-bound for the delta, i.e. the total amount of assigned resources to the elastic services through the previous time window. This approach is chosen to indicate that the tenants would not be risking to lose the resources to be assigned to the non-elastic services. (1g) and (1h) are set to reflect the physical restrictions of the network, namely, the total assigned resources cannot be more than available resources and the infrastructure provider cannot sell non-existing resources.

B. Integration of the prediction data

The mathematical model given in (1a)-(1h) receives the predicted achievable rates per user, i.e. $r_k^{\text{pre}}[n]$, and provides an optimal resource distribution for the given $r_k^{\text{pre}}[n]$. Therefore, the simplest way to integrate the prediction data is the direct implementation of predicted rates in the model. On the other hand, it is clear that, with this straight forward approach, any error in the prediction will have a direct impact on the performance of the scheduler. In order to guarantee the optimality of the predicted resource distribution for the actual achievable rates $r_k[n]$, it requires a prediction algorithm with a very high accuracy level, therefore making the $r_k^{\text{pre}}[n]$ as close as to the actual achievable rates $r_k[n]$. Nevertheless, such a prediction algorithm usually requires a complicated model and have a relatively small prediction horizon which is overly restrictive and is not suitable for a real time application. Consequently, a simpler but less accurate prediction algorithm is more feasible in our scenario.

In order to prevent the prediction errors from affecting the resource distribution, the proposed mathematical model is logically separated into two parts, namely P_1 and P_2. Using the complete mathematical model, (1a)-(1h), P_2 is the part where the prediction data is used to determine the predicted minimum gap, i.e. ξ_m^{pre}, sharing parameters, i.e. S_m^{pre}, Δ_m^{pre}, and the resource distribution among users, i.e. x_k^{pre}. Note that in case of perfect prediction, i.e. the oracle scenario, x_k^{pre} is the optimum resource distribution while ξ_m^{pre} is the minimum achievable gap in the given network. However, due to the issues discussed above, x_k^{pre} and ξ_m^{pre} are not used directly, but instead, they are implemented as the guidance parameters to the real time scheduling problem P_1.

P_1 receives ξ_m^{pre}, x_k^{pre}, S_m^{new}, Δ_m^{new} and r_k^{pre} from P_2 and using (1a),(1b),(1c),(1d), (1e) and (1g), determines the real time resource allocations, $x_k[n]$. The predicted resource distribution x_k^{pre} is used as an upper-bound to the real time resource scheduling, i.e.

$$x_k^{\text{pre}} \geq x_k[n].$$

In this way, the real time scheduler can make small adjustments on the resource allocations in order to tune with the errors in prediction, however, it is not possible to fully avoid from the prediction errors since such a solution would require not only the errors in the prediction but the affects of these errors on resource distribution.

C. Anticipating the sharing parameters

As previously detailed, P_2 derives the optimum sharing parameters (S_m^{pre}, Δ_m^{pre}) based on the predicted channel conditions. On the other hand, depending on the prediction accuracy, the predicted shares can be very inaccurate, resulting in the cases where the tenants choose wrong sharing parameters. Therefore, the update process of the sharing parameters is build on a weighted approach simmilar to one we applied in [6]. In line with our past approach, at the beginning of each new renegotiation interval, the sharing parameters of the tenants are updated using a feature scaling coefficient (α_m), as shown below:

$$S_m^{\text{new}} = (1 - \alpha_m)S_m^{\text{pre}} + \alpha_m S_m^{\text{old}}, \tag{3}$$

$$\Delta_m^{\text{new}} = (1 - \alpha_m)\Delta_m^{\text{pre}} + \alpha_m \Delta^{\text{old}}, \tag{4}$$

where α_m is formalized as:

$$\alpha_m = \frac{|\xi_m - \xi_m^{\text{pre}}|}{\xi_m + \xi_m^{\text{pre}}}. \tag{5}$$

In the best case scenario which can be outlined as the case with perfect prediction, the achieved gap will equal to the predicted gap ($\xi_m = \xi_m^{\text{pre}}$), pulling α_m to zero. In this case, the sharing parameters are directly equal to the predicted sharing parameters. On the other hand, in worst case, where the prediction is totally wrong, the measured gap is much higher than the predicted gap ($\xi_m >> \xi_m^{\text{pre}}$), pushing α_m to one. In this scenario, the wrongly predicted sharing parameters are

not considered at all and the scheduler will keep on using the previous sharing parameters.

D. Active filtering

In the earlier sections, we outlined a scheme how to decrease the impact of prediction errors with bi-level execution of the proposed model. However, it is also clear that this bi-level approach is not sufficient to provide the necessary robustness. In a competitive market, the business decisions (particularly the tenants' incentive to involve in an shared network) are directly affected by the efficiency of the scheduling algorithm. Thus, in this section, a simple yet efficient filter approach is proposed to confine the impacts of prediction errors and enhance the algorithm's robustness to the prediction errors. The proposed filter is formulated as,

$$F(x_k^{\text{pre}}[n], E_k[n]) \; = \; x_k^{\text{pre}}[n] + \frac{E_k[n]}{1 + e^{-a_{1,k}(E_k[n] - a_{2,k})}} \quad (6)$$

where $x_k^{\text{pre}}[n]$ is the calculated optimum resources for the predicted rates, $E_k[n]$ is the prediction error, $a_{1,k}$ and $a_{2,k}$ are filter parameters. In order to calculate the prediction error, we used the euclidean distance between the predicted achievable rate and the measured achievable rate, i.e. $E_k[n] = |r_k^{\text{pre}}[n] - r_k[n]|$. Note that in case of perfect prediction, a.k.a. the oracle scenario, the output of the proposed filter mechanism is equal to $x_k^{\text{pre}}[n]$. The filter's sensitivity to the prediction error depends on $a_{1,k}$ and $a_{2,k}$. Since the prediction error and its effects on the resource allocation can vary over time and the user mix, $a_{1,k}$ and $a_{2,k}$ are chosen to be dynamic. More specifically, these values are calculated using,

$$a_{1,k} = \mu_{n \in W_m}(E_k), \quad (7)$$

$$a_{2,k} = \frac{10}{\sigma_{n \in W_m}(E_k)}, \quad (8)$$

and are updated at the end of every RI based on the error observed during the completed RI.

The output of the filter function $F(x_k^{\text{pre}}[n], E_k[n])$ sets an upper limit to the assignable resources in P_1, namely,

$$F(x_k^{\text{pre}}[n], E_k[n], \beta_k[n]) \geq x_k[n]. \quad (9)$$

Depending on the prediction error ($E_k[n]$), the assignable resources vary within $x_k^{\text{pre}}[n] \leq x_k[n] \leq 1$.

IV. Simulation Results

During our simulations, we assume that the proposed model has been integrated to a base station with a coverage radius of 500 m, and $|K| = 12$ users are sharing the downlink of this base station. The set of users (K) is distributed uniformly to the coverage area and are registered among $|M| = 3$ tenants equivalently, namely $|K_m| = 4$. The presented results are averaged over 50 independent instances and each one of this instances covers a simulation horizon of 5000 transmission time intervals (TTIs), i.e. $N = 5000$ TTIs. Moreover, the simulation horizon is discritized into time slots with a length of 1 TTI.

TABLE I
Comparison between ARIMA and FFNN in terms of their accuracy levels and time complexities.

	ARIMA	FFNN
Time complexity for training process (sec)	-	75.03
Time complexity for prediction process (sec)	1.15	0.598
Prediction error for $\|W_P\| = 10$ms (MAPE)	7.61 %	7,14 %
Prediction error for $\|W_P\| = 50$ms (MAPE)	160.8 %	216,8 %
Adaptation	Yes	No

The simulations are set on Matlab 2017a while the proposed mathematical formulation is run using Gurobi solver [15]. Users movement has been assumed to be in a straight line towards a random direction with the walking speed, i.e. $v = 1.5$ m/s. The users achievable rate is calculated using the Shannon-Hartley theorem, i.e. $r_k[n] = log2(1 + SINR_k[n])$. For each user k, the SINR is calculated under constant transmission power, P_{Tx}, and constant inter-cell interference, I_0 assumptions, using the equation $SINR_k[n] = |h_k[n]|^2 P_{Tx} d_k^{-\alpha}/\sigma^2 + I_0$ where d_k indicates the distance of the user, α is the path loss exponent and σ^2 is the sum of the thermal noise. Moreover, a frequency-flat fading channel with Rayleigh coefficients, i.e. $h_k[n]$, is assumed to be between the user and the base station. The maximum Doppler spread is calculated using $F_d = v f_c/c$ where f_c indicates the carrier frequency of 2 GHz, c shows the speed of light and v is the walking speed of 5.4 km/hr.

A. Comparison between different prediction methods

Our previous studies have revealed that a perfect prediction (i.e. with zero prediction error) can increase the efficiency of the sharing more than 30%. However, the well-known inverse proportion between prediction accuracy and the time complexity challenges the proposed real-time model. More specifically, in order to have performance improvements as high as 30%, the prediction accuracy has to be close to the perfect prediction which requires a complex algorithm that is impossible to run in real-time on a commercially available machine. On the other hand, the decrease in the prediction accuracy can result in efficiency levels lower than the scenario where the prediction is not implemented (i.e. no prediction).

Therefore, the two considered prediction algorithms are compared both in terms of their prediction accuracy and their time complexity. The time complexity of each algorithm is collected from a commercially available computer equipped with i7-4510U CPU and 16 GB ram. Following the general approach in literature, the total time to run the prediction algorithms is divided into two parts, i.e. the learning time and the prediction time (c.f. Table I). The considered learning time is the total duration of building a machine learning model to perform the predictions, whereas, the prediction time compose of the total time spent on making the predictions for the upcoming renegotiation interval. FFNN works in a larger data set in order to reach a generic model that can be used over N, while ARIMA produces a local model that can be used to predict the upcoming RI. Due to its local focus, ARIMA's total time complexity is much lower than FFNN's. In order to

evaluate the accuracy of the prediction algorithm, we used the well-known mean average percentage error (MAPE) and the mean square error (MSE) concepts from the literature, that are calculated using

$$\text{MAPE}(\%) = \frac{100}{N \times |K|} \sum_{k \in K} \sum_{n \in N} \frac{|r_k^{\text{pre}}[n] - r_k[n]|}{r_k[n]}, \quad (10)$$

$$\text{MSE} = \frac{1}{N \times |K|} \sum_{k \in K} \sum_{n \in N} (r_k^{\text{pre}}[n] - r_k[n])^2. \quad (11)$$

Table I proves that the efficiency of FFNN for shorter prediction horizon, while ARIMA is more successful for longer W_P. Morever, due to its locality, ARIMA can adapt itself to the changing correlations over time, whereas, FFNN requires to be retrained in order to maintain the prediction performance. Thus, we concluded that ARIMA is more suitable for our problem.

TABLE II
TO BE ADDED

Scenario (W_P, W_L)	MAPE (%)	MSE
(10,10)	7.61	0.101
(10,50)	7.34	2.14
(10,90)	12.81	0.69
(25,25)	76.64	1.10
(25,50)	29.86	0.84
(25,75)	28.30	0.77
(50,50)	165.9	3.70

Table II outlines the accuracy of ARIMA for different W_P and W_L values. Note that in the given simulation scenario, the correlation window of the achievable rates is 100 TTIs, therefore, the analysis is limited to the $W_P \leq 100$ TTIs. The results underline the importance of W_L as it has direct effect on both MAPE and MSE. Despite the usual approach of choosing $W_P + W_L = 100$ TTIs (i.e. the correlation window), our analysis showed that for smaller W_P, having a relatively too big W_L results in over-fitting and drastically decreases the accuracy. Moreover the first two columns in Table II have similar MAPE values while they show a clear differentiation in MSE. Due to the exponential of the prediction error in (11), the bigger prediction errors are more visible in (11) compared to (10), meaning that if two simulations' MAPE values are identical, in the one with the smaller MSE the prediction error will be more uniformly distributed. Our simulations show that the proposed model performs approximately 3% better with the usage of $|W_L| = 50$ compared to the case where $|W_L| = 10$ despite the huge differentiation in terms of MSE. This indicates that the average value of the error has greater impact on our algorithm than instantaneous errors.

B. Analysis of robustness to the prediction errors

The performance of the proposed anticipatory framework strongly depends on the prediction accuracy. In this part, an analysis of the dependency to the prediction errors and the effects of using the filtering approach is investigated. Fig. 3 reports the variation of average total utility over all the users for $|M| = 2$ with the prediction horizon for three different scenarios, namely, no prediction, no filter and with filter. In the no prediction scenario, the reactive model in [6] is implemented.

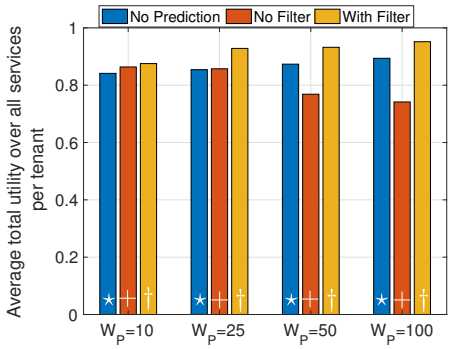


Fig. 3. Comparison of the average achieved utility for different W_P lengths and different scenarios, i.e. no prediction (blue bar marked with " $\star$ "), no filter (orange bar marked with " $+$ ") and application of filter (yellow bar marked with " $\dagger$ ")

Regardless of the prediction horizon, the algorithm with filter implementation performs better than both no prediction and direct prediction implementations. Moreover, increasing the prediction horizon (and the RI) decreases the prediction accuracy which results in lower average achieved utility for the 'no filter' scenario. As can be seen in Fig. 3, for both the 'no prediction' and 'with filter' scenarios, an increase in the total average utility is measured in parallel with the RI. This symmetric increase indicates that the proposed filter mechanism can filter out the negative effects of bad prediction accuracy in the model and exploits the good prediction information.

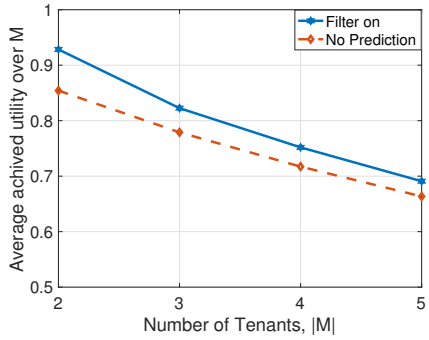


Fig. 4. Variation of average achieved utility over $|M|$ with the number of tenants $|M|$ for a constant network capacity.

Note that in Fig. 3, the differentiation between average achieved utilities per scenario is very low. This is due to the fact that with smaller $|RI|$ the advantage of prediction is being lost as the negotiation platform converges to a real time negotiation algorithm. As in the extreme case of $RI = 1$ TTI the two algorithms (i.e. no prediction and with prediction) performs identical since the resource negotiations are done per time slot. Fig. 4 analysis the effects of increasing $|M|$ (and proportionally $|K|$) on the average achieved utility for

TABLE III
VARIATION OF AVERAGE UTILITY AND TOTAL COST PER TENANT WITH THE NUMBER OF TENANTS IN SHORT TERM WITHOUT PREDICTION

$\lvert M_1\rvert \rightarrow \lvert M_2\rvert$	$(\frac{U_{k,M_1}}{U_{k,M_2}})^{\mu}$	$(\frac{p_{M_1}}{p_{M_2}})^{\epsilon}$	Status
$2 \rightarrow 3$	1.1598	3.1820	YES
$3 \rightarrow 4$	1.1736	1.6810	YES
$4 \rightarrow 5$	1.1901	1.1802	NO

no prediction and with filter cases. In order to have a direct understanding on the results, $RI = W_P = 25$ TTI is chosen while $W_L = 75$ TTI. In this scenario, each tenant serves $|K_m| = 4$ users, thus, the increase in $|M|$ shows the performance of the algorithm for more crowded networks. The results show that the advantage of having a prediction will fade as the network becomes more crowded due to the increase in the non-elastic users. Therefore, in order to exploit the full potential of prediction, the network capacity should be parallel to the network demand.

C. Techno-economic analysis of prediction

In order to analyze the economical impacts of anticipatory network slicing on the envisioned market model, a comparison in terms of tenant's acceptance of the given service quality to the given costs are investigated using the service acceptance inequality in [6], i.e.

$$(\frac{U_{k,M_1}}{U_{k,M_2}})^{\mu} \leq (\frac{p_{M_1}}{p_{M_2}})^{\sigma}. \tag{12}$$

The tenants are considered to accept the given service from the given price if (12) holds. Else it is assumed that the tenants are unlikely to accept the price of service, and consequently would leave the proposed sharing framework. Table III and Table IV shows a comparison between two cases where the prediction is used and not used for $W_P = 25$ TTI. In order to indicate the cases where (12) holds, we used 'YES' and for the other cases we used 'NO'.

TABLE IV
VARIATION OF AVERAGE UTILITY AND TOTAL COST PER TENANT WITH THE NUMBER OF TENANTS IN SHORT TERM WITH FILTER

$\lvert M_1\rvert \rightarrow \lvert M_2\rvert$	$(\frac{U_{k,M_1}}{U_{k,M_2}})^{\mu}$	$(\frac{p_{M_1}}{p_{M_2}})^{\epsilon}$	Status
$2 \rightarrow 3$	1.2555	2.7378	YES
$3 \rightarrow 4$	1.2247	1.6242	YES
$4 \rightarrow 5$	1.1815	1.2001	YES

A comparison between these two tables concurs that the application of anticipation would increase the resource efficiency, allowing tenants to achieve higher average utilities with relatively lower costs. Moreover it increases the capacity of the market, and lets the infrastructure provider to serve more tenants using the same infrastructure.

V. CONCLUSION

In this paper, we have shown that the efficiency of the network slicing can be improved by integrating prediction tools to anticipate the varying traffic and channel conditions. Our investigation has shown that between two popular prediction algorithms in literature, ARIMA is more suitable for our problem. A novel filter has been used to integrate the prediction information in order to dynamically filter out the disadvantages of low prediction accuracy. Numerical results have underlined the importance of a timely capacity expansion in order to exploit the full potential of anticipatory network slicing. Lastly, our mathematical analysis has shown that the increased resource efficiency via anticipation lets the infrastructure providers to serve more tenants using the same infrastructure.

ACKNOWLEDGMENT

This work is funded by the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 643002.

REFERENCES

[1] M. Patzold, "5g readiness on the horizon [mobile radio]," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 6–13, March 2018.

[2] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1790–1821, thirdquarter 2017.

[3] D. Sahinel, C. Akpolat, M. A. Khan, F. Sivrikaya, and S. Albayrak, "Beyond 5g vision for iolite community," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 41–47, January 2017.

[4] P. L. Vo, M. N. H. Nguyen, T. A. Le, and N. H. Tran, "Slicing the edge: Resource allocation for ran network slicing," *IEEE Wireless Communications Letters*, pp. 1–1, 2018.

[5] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management & prioritization in 5g mobile systems," in *European Wireless 2016; 22th European Wireless Conference*, May 2016, pp. 1–6.

[6] O. U. Akgul, I. Malanchini, and A. Capone, "Dynamic resource trading in sliced mobile networks," *submitted to: IEEE Transactions on Network and Service Management,*. [Online]. Available: http://dx.doi.org/10.1787/5jxt46dzl9r2-en

[7] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5g infrastructure markets: The business of network slicing," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9.

[8] I. Malanchini and M. Gruber, "How operators can differentiate through policies when sharing small cells," in *IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015.

[9] D.-E. Meddour, T. Rasheed, and Y. Gourhant, "On the role of infrastructure sharing for mobile network operators in emerging markets," *Computer Networks*, vol. 55, no. 7, pp. 1576–1591, 2011.

[10] J. Luo, J. Eichinger, Z. Zhao, and E. Schulz, "Multi-carrier waveform based flexible inter-operator spectrum sharing for 5G systems," in *Dynamic Spectrum Access Networks (DYSPAN), 2014 IEEE International Symposium on*, April 2014, pp. 449–457.

[11] I. Malanchini, S. Valentin, and O. Aydin, "Wireless resource sharing for multiple operators: Generalization, fairness, and the value of prediction," *Computer Networks*, vol. 100, pp. 110 – 123, 2016.

[12] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5g network slicing resource utilization," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9.

[13] I. Malanchini and V. Suryaprakash, "Minimizing the impact of prediction errors during anticipatory resource allocation," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, June 2018, pp. 1–6.

[14] S. Anbazhagan and N. Kumarappan, "Day-ahead deregulated electricity market price forecasting using recurrent neural network," *IEEE Systems Journal*, vol. 7, no. 4, pp. 866–872, Dec 2013.

[15] Gurobi Optimization Inc., "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com