CellOS: Zero-touch Softwarized Open Cellular Networks

Leonardo Bonati*, Salvatore D'Oro*, Lorenzo Bertizzolo*,
Emrecan Demirors*, Zhangyu Guan†, Stefano Basagni*, Tommaso Melodia*
*Institute for the Wireless Internet of Things,
Northeastern University, Boston, MA 02115, USA
†Department of Electrical Engineering,
The State University of New York (SUNY) at Buffalo, Buffalo, NY 14260, USA
Email: {l.bonati, s.doro, bertizzolo.l, e.demirors, s.basagni, t.melodia}@northeastern.edu, guan@buffalo.edu

***Abstract*—Current cellular networks rely on closed and inflexible infrastructure tightly controlled by a handful of vendors. Their configuration requires vendor support and lengthy manual operations, which prevent Telco Operators (TOs) from unlocking the full network potential and from performing fine grained performance optimization, especially on a per-user basis. To address these key issues, this paper introduces CellOS, a fully automated optimization and management framework for cellular networks that requires negligible intervention ("zero-touch"). CellOS leverages softwarization and automatic optimization principles to bridge Software-Defined Networking (SDN) and cross-layer optimization. Unlike state-of-the-art SDN-inspired solutions for cellular networking, CellOS: (i) Hides low-level network details through a general *virtual network abstraction*; (ii) allows TOs to *define high-level control objectives* to dictate the desired network behavior without requiring knowledge of optimization techniques, and (iii) automatically generates and executes distributed control programs for simultaneous optimization of heterogeneous control objectives on multiple network slices. CellOS has been implemented and evaluated on an indoor testbed with two different LTE-compliant implementations: OpenAirInterface and srsLTE. We further demonstrated CellOS capabilities on the long-range outdoor POWDER-RENEW PAWR 5G platform. Results from scenarios with multiple base stations and users show that CellOS is platform-independent and self-adapts to diverse network deployments. Our investigation shows that CellOS outperforms existing solutions on key metrics, including throughput (up to 86% improvement), energy efficiency (up to 84%) and fairness (up to 29%).**

***Index Terms*—Software-defined Networking, Zero-touch, 5G.**

I. Introduction

Current, state-of-the-art cellular networks rely on proprietary and inflexible hardware and software solutions produced and maintained by few vendors. These closed architectures generally require manual configuration, preventing Telco Operators (TOs) from being able to fully controlling resources such as spectrum, computing and transmission power to optimize network performance [1–3]. Remedies to this fundamental limitation have been piecemeal, often based on offline solutions for frequency assignment and network planning [4, 5]. Optimizing time-sensitive network functionalities also rests on heuristics often engraved in the hardware fabric [6, 7]. As of today, autonomous optimization of network parameters and swift and flexible control of real-time requirements of lower layer protocols are a territory that is largely uncharted.

Through Software-Defined Networking (SDN), TOs are breaking the imposed vendor lock-in by leaving the static and monolithic Radio Access Network (RAN) architecture in favor of using a dynamically programmable, i.e., *softwarized*, *open RAN* for rapid and innovative network deployments [1, 2, 8–10]. Although the benefits of such an open and multi-vendor approach have been showcased widely [11], how to fully embed softwarization in the future 5G infrastructure remains unsettled, as the highly dynamic and distributed nature of cellular networks is not amenable to be addressed by the centralized SDN approach. This issue is further exacerbated by the increasing densification of cellular deployments and users, which makes non-automated control ineffective, if feasible at all. This is witnessed by recent works on cellular and wireless SDN clearly lamenting that the swift dynamics of these networks generate an overwhelming amount of signaling traffic, hardly bearable by traditional softwarized controllers [12–15]. As a consequence, current hardware implementations and centralized softwarized approaches do not allow timely optimization of network behavior and the increasingly needed superior network performance [16, 17].

TOs are extremely sensitive to these issues. For example, the European Telecommunications Standards Institute (ETSI) formed the Zero-touch Network and Service Management group to define fully-automated—*zero-touch*—paradigms to provide flexibility to the highly decentralized technology of future wireless [18]. Similarly, the latest releases of the 3rd Generation Partnership Project (3GPP) include a functional split of 5G NR[1] base stations (called gNBs) capabilities, so that network control decisions that involve large time

This is the preprint version of L. Bonati, S. D'Oro, L. Bertizzolo, E. Demirors, Z. Guan, S. Basagni, and T. Melodia, "CellOS: Zero-touch Softwarized Open Cellular Networks", which has been accepted for publication on Computer Networks (COMNET).

This work was supported in part by the US National Science Foundation under Grant CNS-1618727 and in part by the US Office of Naval Research under Grants N00014-19-1-2409 and N00014-20-1-2132.

[1]Initially introduced as "New Radio" in [19], the term NR now generically refers to the 5G Radio Access Network, having lost its original meaning in the latest 3GPP specifications [20].

scales are made at the gNB Central Unit (gNB-CU), while lower layer and time-sensitive procedures are executed at the gNB Distributed Units (gNB-DUs) deployed closer to the users [21]. The Linux Foundation and the O-RAN Alliance are promoting and building the Open Network Automation Platform (ONAP) and O-RAN, two automated orchestration frameworks to transition the rigid cellular infrastructure to an elastic and softwarized *open RAN* [22, 23]. We observe that, although these approaches foresee network optimization as pivotal, they do not directly implement it. As of now, this is left to the wits of the TO and to the best of our knowledge there is no *zero-touch* solution yet to perform it dynamically.

This paper contributes to the efforts toward *automated softwarization* and *self optimization* of future 5G networks by proposing CellOS, the first *zero-touch* software framework for next-generation cellular networks. Like an operating system interfacing hardware and software functions (whence the name), CellOS flexibly bridges SDN with cross-layer distributed optimization techniques for the cellular architecture. We push the SDN paradigm beyond the traditional separation of control and data planes, in that we also decouple control from optimization, adding further and unprecedented flexibility. Responding fully to ETSI requirements and industry interests, CellOS enables *zero-touch control and optimization* of low-level network functionalities by providing TOs with an efficient, automated, modular, and flexible network control platform. Specifically, CellOS (i) allows TOs to define centralized and high-level control objectives (e.g., "maximize network throughput") without requiring expertise in cross-layer optimization theory or knowledge of network specifics; (ii) provides a general *virtual network abstraction* that shields the TO from the complexity of a sophisticated framework by abstracting network infrastructure and parameters, including those known at *run-time only* (e.g., user-to-base station associations and channel information); (iii) automatically converts high-level control directives into *distributed cross-layer control programs* to be executed at each network edge element, and (iv) enables *zero-touch optimization of distinct control objectives on different network slices coexisting on the same infrastructure* [24].

Figure 1 illustrates the overall structure of CellOS, exemplified for the 3GPP network architecture.

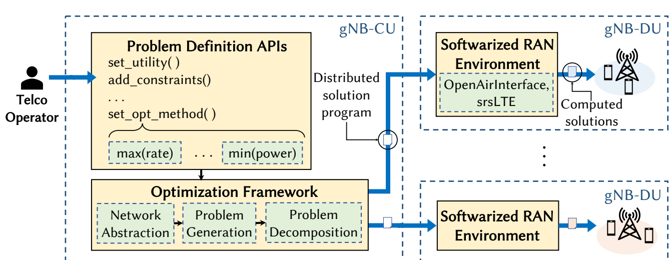


Fig. 1: CellOS at a glance as instantiated for the 3GPP architecture.

The upper-left side of the figure depicts the high level Application Programming Interfaces (APIs) that the TOs can use to define the network control objectives. On the bottom we indicate the components of the framework for automatic generation of the optimization problems and their decomposition into control programs. In a 3GPP scenario this unit corresponds to the gNB-CU, a logical node primarily concerned with control decisions at larger time-scales. On the right, we describe the softwarized RAN that will execute the generated programs. In the 3GPP context, this task would be carried out by the gNB-DU, a logical node that makes time-sensitive decisions involving the lower layers of the protocol stack, and that is interfaced with the gNB-CU.

We have prototyped CellOS on heterogeneous Long Term Evolution (LTE)-compliant testbeds. We have chosen two different implementations of the LTE stack, namely, OpenAirInterface (OAI) [25] and srsLTE [26], to show that our framework is not tied to any specific RAN infrastructure. Our experiments consider a variety of scenarios with multiple base stations and users to show that CellOS optimizes the network performance by swiftly adapting to varying network configurations and settings. We also show the gains in performance that CellOS can bring to RAN implementations for cellular networks, such as OAI and srsLTE, as well as to Medium Access Control (MAC)-layer scheduling algorithms commonly used in cellular networks, i.e., proportional fairness, greedy, and round-robin scheduling algorithms. Results of the comparative performance evaluation of CellOS and prevailing baseline solutions show that using our framework remarkably improves key performance metrics, such as throughput (up to 86%), energy efficiency (up to 84%) and user fairness (up to 29%). We also show that CellOS is transparent to the use of network slicing technologies [27–29], enabling TOs to simultaneously optimize different network functions on distinct network slices. To the best our knowledge this is the first such demonstration, paving the way to the independent management of optimized network slices in 5G systems. Finally, and for the first time, we provide evidence of the potentials of *zero-touch optimization* in a *softwarized RAN* ecosystem by testing CellOS on the long-range open-source POWDER-RENEW PAWR 5G platform [30, 31]. Our results show that CellOS seamlessly interacts with the LTE protocol stack by optimizing resource allocation strategies, successfully increasing the average throughput by 23%.

The remainder of the paper is organized as follows. Section II presents CellOS in the 3GPP context, and a succinct overview of its main components. Details of its architecture are provided in Section III. An example of CellOS operations is given in Section IV. An LTE-compliant prototype of CellOS is illustrated in Section V. Section VI reports the performance evaluation of CellOS on various testbeds, including a lab bench setup, the Arena testbed [32], and the POWDER-RENEW PAWR 5G platform [30, 31], using both the OAI and srsLTE RAN implementations with multiple base stations and users. Work related to our research is surveyed in Section VII. Finally, Section VIII concludes the paper.

II. CellOS in a 5G Flair

This section provides a primer on 5G NR, and an overview of the main CellOS components and on how they can be integrated in the CU/DU functional split introduced by NR.

A. A Brief Overview of 5G NR

Compared to LTE, the 3GPP introduced a series of innovations in NR both in terms of layers of the protocol stack and functionalities, including the support for a wider range of carrier frequencies [33]. The NR frame was endowed with a more flexible structure, which, although still being based on Orthogonal Frequency-division Multiplexing (OFDM), concerns a variable number of symbols per subframe and larger bandwidths with up to 400 MHz per carrier. The 5G RAN can operate in two distinct configurations: *Non-standalone*, i.e., paired with an LTE core network, and *standalone*, i.e., connected to the new 5G Core. Finally, NR base stations, called gNBs, can be deployed in a distributed manner across the network, dividing various parts of the NR protocol stack in different hardware components.

One of the main innovations that NR introduces is the split of the layers of the protocol stack of gNBs into distinct units. These, namely gNB Central Unit (gNB-CU) and gNB Distributed Unit (gNB-DU), can be deployed in separate locations across the cellular network [21] (see Figure 1). Specifically, the gNB-CU, which can control multiple gNB-DUs, involves the higher layers of the 3GPP protocol stack (i.e., Packet Data Convergence Protocol (PDCP), Service Data Adaptation Protocol (SDAP) and Radio Resource Control (RRC)) and makes decisions at larger time scales. The gNB-DU, instead, is deployed closer to the edge of the network and executes time-sensitive procedures, which involve the Radio Link Control (RLC), MAC, and Physical (PHY) layers of the protocol stack. Moreover, the PHY layer of the gNB-DU can be additionally be broken down in a standalone gNB Radio Unit (gNB-RU), which performs functions such as power amplification and signal transmission/reception [34].

While proposed by the 3GPP in [35], this separation has received significant attention due to O-RAN [23], which defined a series of interfaces between the aforementioned gNB elements and a RAN Intelligent Controller (RIC), deployed at the edge of the network. The RIC executes different functions of O-RAN, such as radio resource management, higher layers procedures and policy optimization, and control of RAN elements and resources. Moreover, the RIC includes an application layer, which can host third-party components, such as CellOS, that regulate the behavior of the network.

B. CellOS in a Nutshell

A bird's-eye view of the CellOS architecture is shown in Figure 1. In line with the 3GPP *functional split* [21], CellOS is partitioned in gNB-CU and gNB-DU modular units to decouple the definition of network control procedures (at the gNB-CU) from their execution (at the gNB-DU). CellOS main components are the interface to the TO (providing the *Problem Definition APIs*) and the automatic *Optimization Framework* at the gNB-CU, and the *Softwarized RAN Environment* at the gNB-DU.

By means of a rich variety of APIs, the TO sets the network control objective through high level, highly descriptive directives (e.g., "maximize throughput"), providing few key parameters (e.g., the number of base stations). That is all the TO needs to specify, as CellOS abstracts the underlying network structure, hiding lower-level details to the TO and mapping network elements such as base stations and User Equipments (UEs) into virtual ones (*Network Abstraction* block of our Optimization Framework). As soon as the desired control objective is specified, CellOS converts it into a set of mathematical expressions that are used to define a centralized optimization problem, namely, the analytical representation of the optimization objective and of its constraints (*Problem Generation* block in Figure 1). The generated problem is then *automatically* decomposed into a set of distributed sub-problems, one for each of the edge elements (e.g., base stations). This is done by the *decomposition engine*, a core component of the *Problem Decomposition* block. Based on rigorous mathematical techniques, the centralized problem is partitioned both horizontally (decoupling variables belonging to different elements) and vertically (decoupling variables from different layers of each element's protocol stack). The obtained sub-problems are then automatically converted into executable programs that are individually dispatched to each element (*distributed solution programs*, in the *Softwarized RAN Environment*). Finally, each base station updates the distributed solution program with the real-time network parameters gathered from the RAN software stacks, and runs it through its local solver. It is worth mentioning that CellOS is independent of any specific RAN and can be interfaced with any other current or future 5G softwarized cellular stack. Finally, since CellOS edge elements have access to network real-time information by interfacing with the RAN software stacks (e.g., OAI, srsLTE), they update the received distributed control programs, adapting to the network time-varying dynamics, such as user arrival/departure, and mobility.

III. CellOS Architecture

In this section, we describe in details the components of the CellOS architecture, depicted in Figure 2.

A. Problem Definition APIs

CellOS defines a rich set of APIs to specify general high-level information about the desired network configuration and optimization. These APIs include functions to add base stations and for setting per-user requirements (e.g., minimum rate guarantees). The network control objective can be specified through a simple textual string, e.g., *max(rate)* to maximize the network rate, *min(power)* to minimize the overall power consumption.

```python
from cellos import Network, Engine

# Network instantiation
nwk = Network(bs_num)
slices = nwk.get_slices()

# Optimization problem and optional constraints definition
nwk.set_utility('min(power)', slices[0])
nwk.add_constraints({'user_min_rate':
                        [slices[0].get_users(), rate]})

# Optimization engine initialization
eng = Engine()
eng.set_opt_method('sub-gradient')
nwk.initialize_engine(eng)
```

Listing 1: CellOS API example.

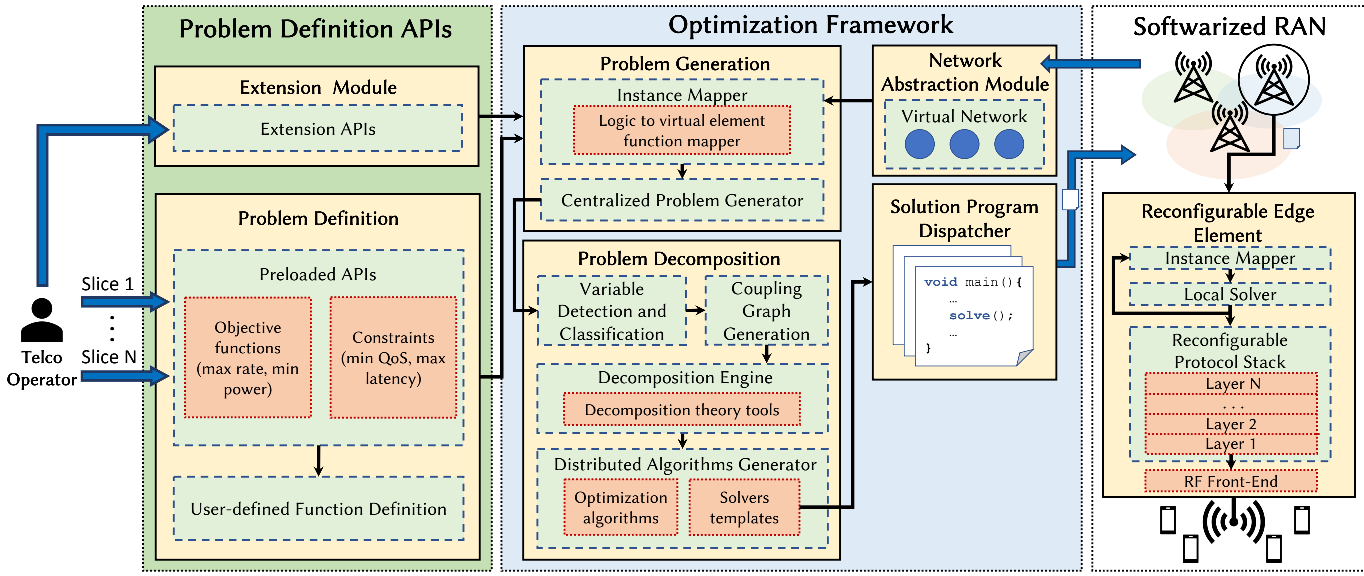


Fig. 2: The CellOS architecture.

An example of CellOS APIs and of the few lines of code needed to program a network objective are shown in Listing 1. In this example, the TO instantiates a new network with a number `bs_num` of base stations (line 2), and gets the network slices instantiated in the network (line 3). An optimization problem aiming at minimizing the transmission power over a specific network slice (`slices[0]`) is then simply set in line 4, with constraints for guaranteeing a minimum rate defined in line 5. It is worth noting that existing slices of the network, active subscribers, and associations of the two, are known a priori by the TO, and stored, for instance, in the cellular core network. We observe that very few lines of code are needed for the TO to set the network goal, after which no further interaction is required. This is because CellOS, dovetailed with the ETSI *zero-touch* principles [18], hides all low-level network details (e.g., channel status, position of mobile users) from the TO through the *network abstraction module* (Section III-B3), and also automatically defines and distributively solves the optimization problem corresponding to the set control objective.

While specifying the objective function in textual form is enough for CellOS to properly work, experienced TOs can define tailor-made custom and more advanced objective functions, optimization techniques, and solvers through an *extension module*. This provides additional APIs for custom mathematical expressions and optimization constraints, and it also allows the TO to select specific optimization techniques and solvers, as well as to achieve fine-grained control of network parameters and functionalities. These are then fed to the optimization framework in a way similar to the preloaded APIs. As of now, CellOS allows to specify functions expressed as linear combination of capacity, Signal-to-Interference-plus-Noise Ratio (SINR), power, and energy efficiency terms, which already enables TOs to formulate a large number of wireless networking optimization problems [36].

B. Optimization Framework

The heart of CellOS resides in its Optimization Framework, which: (i) Converts the high-level centralized code into an optimization problem; (ii) decomposes it into sub-problems; (iii) creates and maintains an abstraction of the network, and (iv) dispatches the solution problems to the Softwarized RAN.

1) Problem Generation: In order to transform high-level specifications into an optimization problem, CellOS first pairs high-level abstraction directives (control objective and constraints) with available network elements (e.g., base stations and users). This is accomplished by the *instance mapper module* that maps physical network elements to their virtual representation, and converts the control objective defined using high-level CellOS APIs (Section III-A) into machine-understandable code. For example, *max(sum(log(rate)))* is converted into $\max \sum_{u \in \mathcal{U}} \log(r_u)$, where $\mathcal{U}$ is the set of UEs and r_u their transmit rate. The generated utility is kept as general as possible by using symbolic placeholders in lieu of parameters whose value will only be known at run-time (e.g., UE-base station associations, channel coefficients, interfering signals, etc.). In so doing, our Optimization Framework is UE-agnostic. It is the base stations that, at run-time, replace the symbolic placeholders with their current value. Specifically, base stations interfaced with CellOS expose parameters and variables that can be tuned and optimized. Thus, placeholders of the generated problems always match physical network capabilities.

2) Problem Decomposition: This component of the Optimization Framework partitions the centralized problem into multiple sub-problems, one for each network element, to be solved distributively at each base station. In general, the centralized network control problem can be formalized as the following network utility maximization problem

$$\underset{\mathbf{x} \in \mathcal{X}}{\text{maximize}} \quad f(\mathbf{x}) \qquad \text{(CEN)}$$

$$\text{subject to} \quad g_i(\mathbf{x}) \leq h_i(\mathbf{x}), \quad \forall i \in \mathcal{I} \qquad (1)$$

where $\mathbf{x}$ represents the optimization variables (e.g., scheduling policies or transmission power levels), $\mathcal{X}$ is the strategy space (i.e., the set of all feasible strategy combinations), $f(\cdot)$ is the network-wide objective function (e.g., the overall capacity or the total energy efficiency of the network). Inequality (1) represents the set $\mathcal{I}$ of constraints (e.g., the transmission power must be bounded by some constant value; each Physical Resource Block (PRB) can be allocated to one UE only, etc.). The biggest challenge in solving (CEN) is that both objective function and constraints are, in general, coupled to different edge elements and to different layers of each element protocol stack. Because of this tight coupling, generating distributed sub-problems that can be locally solved by each base station becomes challenging.

To address this challenge, CellOS first automatically identifies coupled variables and then applies rigorous decomposition to generate new sub-instances of (CEN) that are automatically assembled into uncoupled distributed programs to be executed at each base station. This is accomplished performing the following (Figure 2): *variable detection and classification*, *coupling graph generation*, decomposition (through the *decomposition engine*), and *distributed algorithms generation*.

Variable Detection and Classification: CellOS starts by identifying the optimization variables of the network control problem. This is done by parsing the generated objective function expression looking for symbolic placeholders introduced therein. For instance, in (CEN) CellOS detects $\mathbf{x}$ to be the set of optimization variables of the problem. Then, it determines which layer of the protocol stack houses which variable, e.g., power belongs to the PHY layer, scheduling to the MAC layer, and so on. CellOS then identifies to which base station each variable belongs to. As a result, each variable is assigned to a specific base station and to one of its protocol stack layers.

Coupling Graph Generation: After detecting and classifying problem variables, CellOS organizes their coupling in a graph $G = (V, E)$, where V is the set of variables of the network control problem, which are joined by an edge in E only if they are coupled. Similarly to what done in the previous step, coupling among variables is detected through a symbolic parser. As an example, a coupling graph for $f(\mathbf{x}) = x_2(x_4 + x_5) + x_3(x_4 + \frac{x_1}{x_2})$ is shown in Figure 3a. Variables $\{x_i\}_{i=1,3}$ and $\{x_j\}_{j=2,4,5}$ belong to eNB_1 and eNB_2 (Figure 3b), respectively.

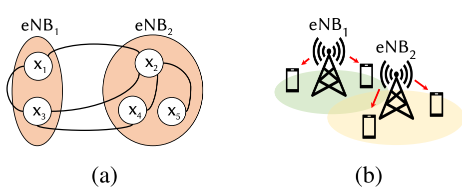


Fig. 3: (a) Coupling graph for $f(\mathbf{x}) = x_2(x_4+x_5)+x_3(x_4+x_1/x_2)$; (b) Network scenario considered in Section IV.

Figure 3a shows that coupling is not limited to variables of a single eNB, but it might also involve those controlled by other eNBs.

Decomposition Engine: Variable detection/classification and coupling graphs are preliminary to automated problem decomposition, which we perform by using well-established techniques, including duality theory [37] and decomposition via partial linearization [16] (additional ones can be implemented through the *extension module* of Figure 2). Decomposability is achieved introducing auxiliary variables (e.g., Lagrangian multipliers, penalization terms, and aggregate interference functions) that remove coupling across optimization variables and generate objective functions and constraints with separable terms in the sense of [37]. Unfortunately, coupling in cellular networks involves heterogeneous network elements and different layers of the protocol stack, resulting in optimization problems whose utility or constraints are rarely separable. For this reason, it is classified into *horizontal coupling* and *vertical coupling*. The former reflects dependencies among different network elements (e.g., among interfering base stations and their subscribers). The latter, instead, concerns *cross-layer* dependencies among different layers of the protocol stack of the same element (e.g., MAC policies affect transmission power and modulation strategies at the PHY layer). Coupling makes centralized control of cellular networks extremely challenging as (i) the number of variables of the problem grows exponentially with the number of network elements, resulting in high computational and time complexity; (ii) the TO needs to be fully aware of the underlying network topology, the traffic demand, and the Channel State Information (CSI) for each individual UE and base station, and (iii) centralized approaches require real-time information exchange between each network element and the centralized controller, imposing high signaling overhead and latency. It is worth to point out that such network real-time information is not known at CellOS controller, but only at the edge elements. Due to the fast changing network dynamics, though, the time required to signal local information to the controller, compute a centralized solution, and adopt it at the edge elements might exceed the coherence time of the found solution. *Such solutions, may refer to an old network state and be obsolete, thus resulting in poor performance.* This makes distributed solutions highly desirable, if not mandatory. Even though distributed algorithms might not always guarantee globally optimal solutions, they usually manage to compute locally optimal ones with significantly lower computational complexity, while ensuring run-time performance [16, 17].

We point out that this work does not focus on proposing new decomposition theories. *Our aim, instead, is to automatically generate distributed optimization programs based on a high-level objective, irrespective of the decomposition method used.*

Distributed Algorithms Generator.: The final step to achieve distributed control of the cellular network is to generate *distributed solution programs* which can be executed and solved by each base station via standard optimization solvers. This task is performed by the *distributed algorithms generator* unit of CellOS Optimization Framework (Figure 2). As mentioned, the Optimization Framework is not cognizant of the value of parameters that are known at run-time only. Accordingly, the *distributed solution programs* contain symbols in place of these parameters. Each base station will then replace these symbols with their actual value at run-time, and associate optimization variables to the served UEs. The *instance mapper* module has been designed to perform this task (Figure 2). This is one of the most important features of CellOS as it makes the solution program generation process (i) fully automated; (ii) independent of network configuration, and (iii) self-adapting to compute parameters at run-time based on current network conditions.

3) Dispatcher and Abstraction Module: The last two components of the Optimization Framework are the *solution program dispatcher* and the *network abstraction module*. The *dispatcher* utilizes sockets to transfer the generated distributed solution programs to each network base station, which will execute and solve them to achieve the desired network objective.

The *network abstraction module* creates a high-level representation of the network infrastructure, hiding low-level, hardware/software details from the TO. This abstraction allows the *problem generation* (Section III-B1) to automatically convert directives and constraints given through the APIs of Section III-A into mathematical expressions and utility functions.

C. Softwarized RAN

The third main component of the CellOS architecture (Figure 2) is in charge of running the distributed solution programs at each network element so as to reach the global network objective requested by the TO. Once the dispatcher has delivered the programs, the *instance mapper* component of the Reconfigurable Edge Element (REE) replaces the symbolic placeholders in the program with their corresponding run-time values. This component is capable of dynamically adapting solution programs to current network conditions, such as arrival/departure of UEs, handovers, and CSI. At the end of this mapping procedure each program is executed by the *local solver* and a solution is computed. As mentioned above, CellOS uses decoupling terms (e.g., Lagrangian multipliers) to allow individual base stations to coordinate with each other. Relevant parameters are iteratively updated and exchanged among the coupled REEs through already available inter-base station interfaces (e.g., X2/Xn interfaces of cellular networks).

Since all the decisions are made locally at the base stations, at most $|\mathcal{U}|\,(|\mathcal{N}|+1)$ variables need to be exchanged at each iteration, where $\mathcal{U}$ is the set of users, $\mathcal{N}$ are the available transmission channels, and $|\cdot|$ denotes the cardinality operator. As we will demonstrate in Section VI-C4 through experimental results, this overhead is negligible if compared to that of centralized approaches, which need to gather local information at the central controller. Because of this very limited signaling overhead, our framework effectively self-adapts to the network fast changing behavior. Upon computing optimal solutions for each local network control problem (e.g., transmission and scheduling policies), these are used by each REE through the Reconfigurable Protocol Stack (RPS), which controls MAC and PHY layers, among others.

IV. CellOS in Action: An Example

We consider the scenario depicted in Figure 3b, where two interfering eNBs in the set $\mathcal{B}$ share two channels and serve two UEs each. Here, $\mathcal{U}_b$ is the set of users u served by eNB $b \in \mathcal{B}$. We consider a downlink cross-layer optimization problem where each eNB has a transmission power budget $P^{\max}$, and that the UEs request a minimum capacity $C^{\min}$. The optimization variables of this problem concern MAC and PHY layers, namely, user scheduling and transmission power allocation. In this example, we assume that the TO uses CellOS to maximize the network capacity. The TO first instantiates a network with two base stations (`nwk = Network(2)`). Then the following network control objective is set on the slice controlled by the TO: `nwk.set_utility('max(capacity)', slices[0])`.

On the other hand, CellOS needs to perform a more complex set of operations to reach the objective specified so succinctly by the TO. Let $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2)$ represent the network scheduling profile, where $\mathbf{y}_b = (y_{b,1,n}, y_{b,2,n})_{n=1,2}$ is the scheduling profile for eNB $b \in \{1, 2\}$. Let $y_{b,u,n}$, instead, represent the scheduling variable such that $y_{b,u,n} = 1$ if user u is scheduled for downlink transmission on channel $n \in \mathcal{N} = \{1, 2\}$ and $y_{b,u,n} = 0$, otherwise. Similarly, $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2)$ represents the network power allocation profile, where $\mathbf{p}_b = (p_{b,1,n}, p_{b,2,n})_{n=1,2}$ is the power allocation profile for eNB b, and $p_{b,u,n}$ represents the downlink transmission power from b to user u on channel n. Let $C_{b,u,n}(\mathbf{y}, \mathbf{p})$ be the capacity for UE u served by eNB b on channel n, expressed as

$$C_{b,u,n}(\mathbf{y}, \mathbf{p}) = B \log_2 \left(1 + \frac{g_{b,u,n} y_{b,u,n} p_{b,u,n}}{N + g_{b',u,n} \sum\limits_{u' \in \mathcal{U}_{b'}} p_{b',u',n} y_{b',u',n}} \right), \quad (2)$$

where B is the employed bandwidth, N is the background noise power, and $g_{b,u,n}$ is the channel gain coefficient computed by u and sent to b, as part of standard cellular networks signaling procedures between user and base station (e.g., LTE Physical Uplink Control Channel (PUCCH)).

The centralized network control problem can be expressed as the following Capacity Maximization Problem (CMP)

$$\underset{\mathbf{y}, \mathbf{p} \in \mathcal{X}}{\text{maximize}} \quad \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} \sum_{n=1}^{2} C_{b,u,n}(\mathbf{y}, \mathbf{p}), \qquad \text{(CMP)}$$

$$\text{subject to} \quad \sum_{n=1}^{2} C_{b,u,n}(\mathbf{y}, \mathbf{p}) \geq C^{\min}, \quad \forall b \in \mathcal{B}, u \in \mathcal{U}_b \quad (3)$$

$$\sum_{u \in \mathcal{U}_b} \sum_{n=1}^{2} p_{b,u,n} \leq P^{\max}, \quad \forall b \in \mathcal{B} \quad (4)$$

$$\sum_{n=1}^{2} y_{b,u,n} \leq 1, \quad \forall b \in \mathcal{B}, \forall u \in \mathcal{U}_b \quad (5)$$

where (3) represents the users' minimum capacity constraint, (4) enforces eNBs' power budget, and (5) guarantees that each eNB allocates each channel to a single UE only.

The main challenges in decomposing (CMP) are: (i) It is a Mixed Integer Non-Linear Programming problem, which is NP-hard in general [38], and (ii) both (2) and (3) are coupled among different eNBs.

CellOS recognizes $\mathbf{y}$ and $\mathbf{p}$ to be the problem optimization variables and associates them to the MAC and PHY layers, respectively. Now, the *problem decomposition* module understands which variables belong to which eNB and creates a coupling graph similar to that in Figure 3a. This is, then, used to detect the aggregate interference term in the capacity expression (2). Accordingly, it defines the following auxiliary function

$$h_{b,u,n}(\mathbf{y}_{-b}, \mathbf{p}_{-b}) = \sum_{b' \in \mathcal{B} \setminus \{b\}} g_{b',u,n} \sum_{u' \in \mathcal{U}_{b'}} p_{b',u'n} y_{b',u',n} \quad (6)$$

where $\mathbf{y}_{-b} = \mathbf{y} \setminus \{\mathbf{y}_b\}$ and $\mathbf{p}_{-b} = \mathbf{p} \setminus \{\mathbf{p}_b\}$ are the scheduling and power allocation variables of the eNBs belonging to $\mathcal{B} \setminus \{b\}$. At this point, new auxiliary variables are introduced to rewrite (CEN) as

$$\underset{\mathbf{y},\mathbf{p},\mathbf{i}}{\text{maximize}} \quad \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} \sum_{n=1}^{2} C_{b,u,n}(\mathbf{y}_b, \mathbf{p}_b, \mathbf{i}_b), \quad \text{(DCMP)}$$

$$\text{subject to} \quad \sum_{n=1}^{2} C_{b,u,n}(\mathbf{y}_b, \mathbf{p}_b, \mathbf{i}_b) \geq C^{\min}, \quad \forall b \in \mathcal{B}, u \in \mathcal{U}_b \quad (7)$$

$$i_{b,u,n} \geq h_{b,u,n}(\mathbf{y}_{-b}, \mathbf{p}_{-b}), \forall b \in \mathcal{B}, u, n = 1, 2 \quad (8)$$

$$\textit{Constraints} \quad (4), (5)$$

CellOS can now use duality optimization tools to generate the following Lagrangian dual function

$$\begin{aligned} L(\boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{i}, \mathbf{y}, \mathbf{p}) &= \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} \sum_{n=1}^{2} C_{b,u,n}(\mathbf{y}_b, \mathbf{p}_b, \mathbf{i}_b) \\ &- \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} \lambda_{b,u} \left(C^{\min} - \sum_{n=1}^{2} C_{b,u,n}(\mathbf{y}_b, \mathbf{p}_b, \mathbf{i}_b) \right) \\ &- \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} \sum_{n=1}^{2} \mu_{b,u,n} \left(h_{b,u,n}(\mathbf{y}_{-b}, \mathbf{p}_{-b}) - i_{b,u,n} \right), \end{aligned} \quad (9)$$

where $\boldsymbol{\lambda} = (\lambda_{b,u,n})$ and $\boldsymbol{\mu} = (\mu_{b,u,n})$ are the non-negative Lagrangian multipliers used in constrained optimization [37].

We observe that problems (CMP) and (DCMP), and the Lagrangian dual function (9) all aim at solving the centralized control problem (CEN). However, the advantage of using (9) is that function $L(\boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{i}, \mathbf{y}, \mathbf{p})$ is written with separable variables, meaning that it can be split into $|\mathcal{B}|$ sub-problems locally solvable by each eNB.

Finally, CellOS dispatches the generated distributed solution programs to the eNBs that populate them with network run-time information (e.g., users' channel coefficients), and compute optimized solutions through their *local solver*.

It is worth noting that the procedures detailed in Sections III-A and III-B need to be executed only once per control problem specified by the TO and that they take very little time to be performed, e.g., 0.03 s for the example of this section (more details on the scalability of CellOS automatic procedures will be given in Section VI-C4).

V. OAI-based CellOS Prototype

In this section, we discuss the prototypes of CellOS, which have been built based on the OAI and srsLTE open-source RAN implementations. The OAI-based prototype is illustrated in Figure 4.

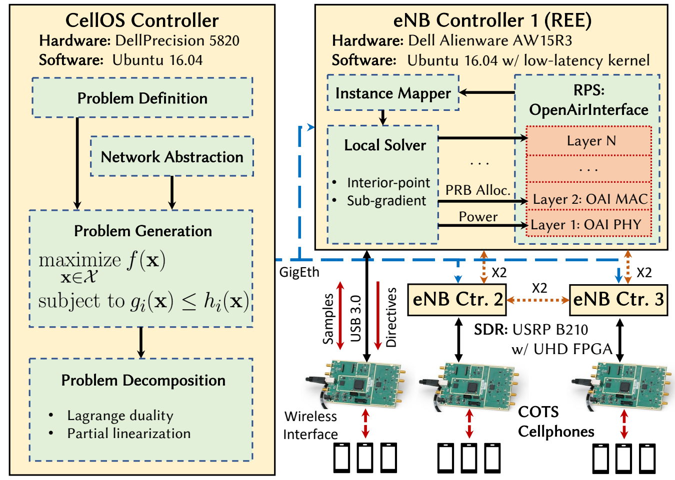


Fig. 4: OAI-based CellOS prototype.

The *CellOS Controller* performs the functionalities of the Problem Definition APIs and of the Optimization Framework. Particularly, it creates and maintains the network abstraction, generates the optimization problem based on the directives from the TO, and performs the problem decomposition. In our experiments the decomposition process is obtained through Lagrangian duality theory [37] and decomposition via partial linearization [16].

Multiple *eNB Controllers*, one for each base station, are connected to the CellOS Controller through a Gigabit Ethernet connection. These controllers use interior-point and subgradient algorithms [37] to solve the received distributed programs, and set the parameters to be used with the RF front-ends they are connected to. Each of these controllers drives an Ettus Research Universal Software Radio Peripheral (USRP) B210, which serves UEs over LTE frequencies. As UEs we used a set of heterogeneous Commercial Off-the-Shelf (COTS) cellular phones (Samsung Galaxy S5, S6 and S7, and Apple iPhone 6s).

In this prototype, CellOS interfaces with the LTE protocol stack implementation offered by *OpenAirInterface*, i.e., an open-source software-based experimental platform for LTE implementations [25]. OAI features LTE RAN applications along with Evolved Packet Core components. As OAI does not directly allow per-user power control, or optimized PRB allocation—key essential requirements of many network control objectives—we have extended its functionalities by significantly modifying its core implementation. Specifically, power control is obtained by amplitude-modulating the downlink data signal intended for a specific UE. PRB allocation, instead, is based on an optimized waterfilling-like fair scheduling algorithm [39], which has low-complexity, thus complying with LTE strict timing requirements. Because of the PRB short time duration it is of utmost importance to compute the PRB allocation very quickly to guarantee compliance with LTE and promptly serve the UEs. According to our scheme, PRBs are allocated only to those UEs whose downlink transmission buffer is not empty.

A similar approach has been followed for the *srsLTE* prototype, which leverages USRPs X310 in place of USRPs B210.

This time, each eNB controller connects to the Software-Defined Radio (SDR) through a $10\,\mathrm{Gbit/s}$ PCI Express network card. In this prototype, CellOS interfaces with the open-source cellular protocol stack offered by srsLTE, which, similarly to what done for OAI, has been extended to perform PHY-layer power control by adjusting the USRPs transmission power, and MAC-layer scheduling by optimally allocating PRBs to UEs.

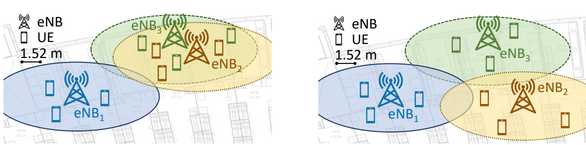
VI. Experimental Evaluation

The effectiveness of CellOS in automatically creating distributed optimization programs from high-level directives, and in managing the network infrastructure to reach different control objectives, is demonstrated via experimentation on various LTE-compliant testbeds. We describe our testbed in Section VI-A, we introduce the investigated performance metrics in Section VI-B, and present our experimental results in Section VI-C.

A. Network Scenarios and Testbed Settings

To demonstrate its platform-independence, we test CellOS over different software and hardware platforms, using OAI and srsLTE, as well as heterogeneous software-defined radios and testbeds.

The OAI-based prototype of Section V has been used in a testbed composed of 3 eNBs and up to 9 UEs. Each eNB uses a $10\,\mathrm{MHz}$ channel bandwidth corresponding to 50 PRBs. For this prototype we consider the two indoor scenarios depicted in Figure 5: (i) A high interference scenario, where two eNBs are in line-of-sight conditions and have largely overlapping coverage areas (Figure 5a), and (ii) a low interference scenario where eNBs are in non-line-of-sight conditions and their coverage areas only partially overlap with each other (Figure 5b).



(a) High interference. (b) Low interference.

Fig. 5: The CellOS lab bench testbed.

The high interference scenario represents those crowded environments (e.g., university campuses, concert halls or convention centers) where several femtocells are deployed in a crowded region to balance the traffic load of a macrocell farther away. In this case, while the interference among macro- and femtocells is small, femtocells with overlapping coverage areas are subject to significant inter-cell interference. In the low interference scenario, instead, eNBs are located far away from each other and, thus, are less subject to inter-cell interference and the subsequent performance degradation.

The srsLTE-based prototype is evaluated on a low-interference setup on the Arena testbed [32]. We instantiated 3 LTE eNBs on USRPs X310 whose antennas are connected to the ceiling of a $208.1\,\mathrm{m}^2$ office space. A set of Dell EMC PowerEdge R340 servers are used to drive the USRPs through 10 Gigabit Ethernet connections. This set of experiments shows that CellOS can *simultaneously obtain different control objectives on multiple network slices.* This represents the scenario in which multiple Mobile Virtual Network Operators (MVNOs) share the same edge elements, or that of a single TO wishing to set diverse control problems on each network slice. To demonstrate the benefits of automatic optimization of the *open RAN*, we finally instantiate CellOS on the long-range open-source 5G POWDER-RENEW platform [31], which is the combination of the Platform for Open Wireless Data-driven Experimental Research (POWDER) and Reconfigurable Ecosystem for Next-generation End-to-end Wireless (RENEW), and part of the Platforms for Advanced Wireless Research (PAWR) [30].

We assess CellOS performance by letting UEs download a file stored on our local server for $60\,\mathrm{s}$. *It is worth mentioning that it only took CellOS* $1.43\,\mathrm{s}$ *and 8 lines of code (see Listing 1) to automatically generate the evaluated distributed control programs* (more details on the scalability of these operations will be given in Section VI-C4)

B. Performance Metrics

CellOS has been evaluated against the following metrics.

- *Sum throughput of the network*, defined as

$$S = \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} S_{b,u}, \quad \forall b \in \mathcal{B}, u \in \mathcal{U}_b \tag{10}$$

  where $\mathcal{B}$ and $\mathcal{U}_b$ are the sets of the eNBs b and of UEs u they are serving, and $S_{b,u}$ is the throughput offered to $u \in \mathcal{U}_b$ by b.
- *Normalized transmission power* of the base stations to the UEs. To analyze the impact of power control policies on the transmission power of eNBs, we show the transmission power of the base stations normalized to their maximum transmission power. Let $P_b^{\max}$ and $P_b^{\min}$ be the maximum and minimum power levels of base station b, the normalized transmission power is defined as

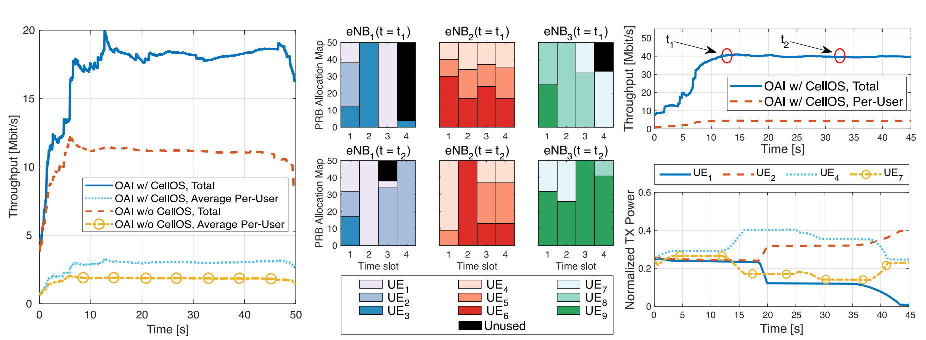
$$P_{b,u}^N = \frac{P_{b,u} - P_b^{\min}}{P_b^{\max} - P_b^{\min}}, \quad \forall b \in \mathcal{B}, u \in \mathcal{U}_b \tag{11}$$

  where $P_{b,u} \in \{P_b^{\min}, P_b^{\max}\}$ is the power used by eNB $b \in \mathcal{B}$ to transmit to its user $u \in \mathcal{U}_b$.
- *Global energy efficiency*, defined as the amount of information per unit of energy the eNBs transmit to their subscribers:

$$EE = \frac{\sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} S_{b,u}}{\sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} P_{b,u}}, \quad \forall b \in \mathcal{B}, u \in \mathcal{U}_b \tag{12}$$

  where $P_{b,u}$ is the power used by eNB b to transmit to its user u.
- *System fairness*, measured through Jain's equation [40]. Given users $u \in \mathcal{U} = \bigcup_{b \in \mathcal{B}} \mathcal{U}_b$, Jain's fairness index $\mathcal{J}$ is defined as

$$\mathcal{J} = \frac{(\sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} S_{b,u})^2}{|\mathcal{U}| \sum_{b \in \mathcal{B}} \sum_{u \in \mathcal{U}_b} S_{b,u}^2}, \quad \forall b \in \mathcal{B}, u \in \mathcal{U}_b. \tag{13}$$



(a) OAI w/ and w/o CellOS. (b) PRBs at time t_1 and t_2. (c) Throughput and power.

Fig. 6: Throughput maximization in the high interference scenario on the OAI-based prototype.

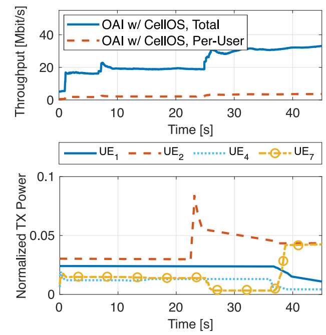


Fig. 7: Power minimization in the high interference scenario on the OAI-based prototype.

TABLE I: Summary of experimental setup.

Figure	**Optimization Problem**	**Scenario**	**RAN Software**	**Testbed**
Fig. 6	max(rate)	High Interference	OAI [25]	Lab Bench Setup
Fig. 7	min(power)			
Fig. 8	max(sum(log(rate)))			
Fig. 9	max(sum(log(rate)))	Low Interference		
Fig. 10	max(rate)			
Fig. 11a	max(rate)	Slicing	srsLTE [26]	Arena [32]
Fig. 11b	min(power)			
Fig. 12	max(rate), min(power), max(sum(log(rate)))	Controller Time	N/A	
Fig. 13		Local Solver Time		
Fig. 14		Signaling Overhead		
Fig. 15a	max(rate)	Long-range	srsLTE [26]	POWDER-RENEW [30, 31]

C. Experimental Results

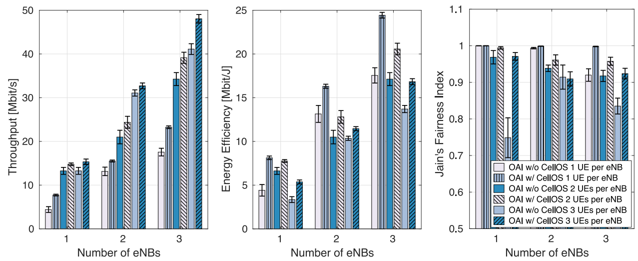
CellOS has been evaluated against the metrics of Section VI-B in a variety of network configurations (i.e., high and low interference, with and without network slicing), and on different testbeds, including a lab bench setup, the Arena testbed [32], and the POWDER-RENEW PAWR 5G platform [30, 31].

To fully appreciate the effects of the automatic optimization procedures introduced by CellOS, we consider a cellular network implemented through OAI and srsLTE and we compare the achieved network performance with and without CellOS. Moreover, we also compare the performance achieved by state-of-the-art scheduling algorithm commonly used in commercial cellular networks, *i.e., proportional fairness, greedy, and round-robin,* to that achieved by CellOS-managed networks. A summary of our experimental setup is shown in Table I.

1) High Interference Scenario: Figure 6 presents results obtained when optimizing throughput (network control objective of *max(rate)*) in the high interference scenario in Figure 5a. We start by evaluating the throughput gains brought to OAI by CellOS zero-touch approach. Average total and per-user throughput are shown in Figure 6a. We observe that CellOS brings significant benefits to the network performance, with improvements as high as 75% (63% on average). This is because of the interplay between the optimized per-user power control and scheduling determined by CellOS and executed locally by the Softwarized RAN. Indeed, CellOS automatic optimization procedures allow the eNBs to serve UEs with an optimized resource allocation and power-controlled signals, which significantly reduces the inter-cell interference while guaranteeing a minimum rate to UEs. To provide further insights on the resource allocation procedures automatically executed by each eNB, we investigated the network throughput, and power and PRBs allocated to the users during an experiment run of the *max(rate)* solution program (Figures 6b and 6c, respectively). For clarity, only the power for four users is shown. As time progresses, the throughput (both total and per-user) plateaus out to a stable value, which is a consequence of local optimality of the solution program that successfully limits interference. Power is changed for the individual user in time, also responding to optimization requirements and reflecting current network conditions. Figure 6b depicts the PRBs allocated to UEs at time instants t_1 and t_2 of Figure 6c. We observe that the eNBs adapt the PRB allocation in *real-time* to satisfy user requests while achieving the set network objective. In fact, time slots with unassigned PRBs may even occur, without compromising the eNB ability of satisfying its subscribers requirements.

To show that different network control objectives produce different results, we investigate throughput and power determined by CellOS for power minimization (control objective of *min(power)*), while guaranteeing a minimum per-user data rate of $1\,\mathrm{Mbit/s}$ (Figure 7). As expected, the achieved throughput is lower than that of the *max(rate)* control program (Figure 6c). This is due to the normalized transmission power of the eNBs being remarkably lower than that in Figure 6c (up to one order of magnitude). We notice, though that UEs achieve an average throughput of $2.63\,\mathrm{Mbit/s}$, which satisfies the constraint on their minimum rate.

The next set of experiments concerns the performance of OAI with and without CellOS in scenarios with varying number of eNBs and UEs. The network control objective requires to maximize throughput while explicitly accounting for fairness, namely, is set to *max(sum(log(rate)))*. Scenarios with one base station consider only eNB_3, while Scenarios with two base stations concern eNB_2 and eNB_3, i.e., the base stations with overlapping cells (see Figure 5a). Results concerning sum throughput, energy efficiency and fairness are shown in Figure 8.



(a) Throughput. (b) Energy efficiency. (c) Fairness.

Fig. 8: Sum-log-rate maximization in the high interference scenario on the OAI-based prototype w/ and w/o CellOS.

The throughput comparison is shown in Figure 8a, where we can see that OAI with CellOS always outperforms OAI without CellOS. In Figure 8b, we evaluate energy efficiency, pivotal in large-scale networks [41]. As expected, since our framework achieves a higher throughput with a lower power expenditure, the network is more energy efficient when managed by CellOS. System fairness is shown in Figure 8c. We notice that, in general, CellOS improves user fairness, with increases up to 29%. Improvements are more evident in scenarios with higher number of eNBs and UEs, as optimization techniques are more effective in those more dense scenarios with higher interference. Specifically, since in these scenarios suboptimal algorithm solutions generate inefficient resource allocation policies, optimal ones are required the most. Indeed, CellOS optimized resource allocation, and its ability to fine-tune the power directed to the served UEs allows the base stations to contain the interference directed to other eNBs, thus increasing the network performance.

2) Low Interference Scenario: These experiments concern 3 eNBs and 9 UEs in low interference conditions (Figure 5b). Results on throughput and on the allocated

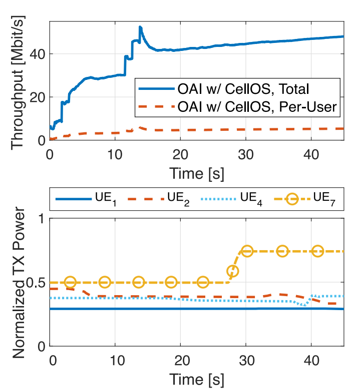


Fig. 9: Sum-log-rate maximization in the low interference scenario on the OAI-based prototype w/ CellOS.

normalized power are shown in Figure 9. In this scenario CellOS is required to optimize the network control objective *max(sum(log(rate)))*. As expected, performance is better than in the high interference scenario because of the lower interference level, that allows the eNBs to use higher power without disrupting each other transmissions. In Figure 10, we compare CellOS rate maximization with two well-known state-of-the-art scheduling algorithms: The *proportional fairness* algorithm, that is the *de facto standard* in cellular networks [7, 42], and the *greedy* algorithm [43]. We notice that CellOS

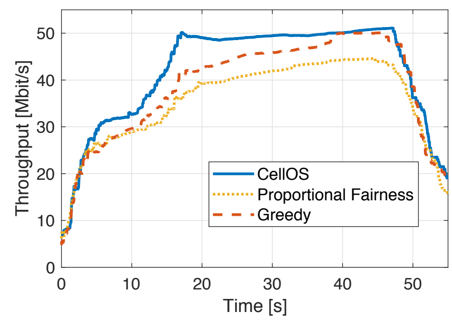


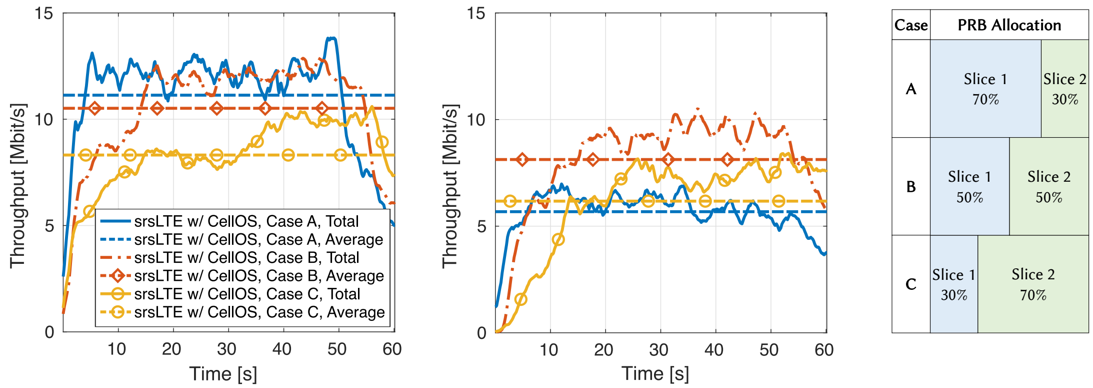
Fig. 10: Rate maximization in the low interference scenario: OAI w/ CellOS vs. OAI w/ *proportional fairness* [42] and OAI w/ *greedy* [43] scheduling policies.

outperforms the proportional fairness algorithm because of this overarching optimization approach to network management. The greedy approach, instead, obtains throughput levels similar to those of CellOS, albeit with a significant delay. Indeed, because of its optimized MAC-layer procedures, which allow the network base stations to mindfully allocate resources to the served UEs, CellOS achieves said throughput level after only few seconds from the system start and maintains it until the UEs finish downloading data.

3) Network Slicing: This set of experiments concerns 3 eNBs instantiated on the USRPs X310 of the Arena testbed [32] through srsLTE. The eNBs serve 9 COTS UEs. The antennas of the USRPs are hung off the ceiling of a $208.1\ \mathrm{m}^2$ office space.

We target a scenario in which multiple MVNOs lease infrastructure resources from an Infrastructure Provider (IP). The IP, which owns the physical equipment (e.g., the base stations), allocates slices of the network to MVNOs following, for instance, the approach described in [29]. Since MVNOs act independently from one another, with different subscribers and requirements (e.g., quality of service), they may need to optimize different control programs on their slice of the network. Considering this, and cognizant of current 5G cellular networks trends, we designed CellOS to handle different network slicing configurations.

Figure 11 showcases the unique ability of CellOS in implementing different control strategies for different network slices, *simultaneously optimizing different control programs on different network slices,* namely, Slice 1 and Slice 2, on each eNB. Specifically, Slice 1, which is allocated to MVNO 1, aims at maximizing the network throughput, while Slice 2, allocated to MVNO 2, minimizes the power consumption. The network sum and average throughput achieved by this per-slice behavior are shown in Figure 11. In our experiments, the two slices were allocated different percentages of the available PRBs (see Figure 11c): First 70% to Slice 1 and 30% to Slice 2 (Case A of Figure 11), then 50% to each slice (Case B), and finally a 30%–70% allocation was used (Case C). Figure 11a shows the throughput of Slice 1 in the three



(a) (b) (c)

Fig. 11: Optimization of different control programs on different slices on the srsLTE-based prototype instantiated on the Arena testbed [32]: (a) Throughput of Slice 1 (*max(rate)*); (b) throughput of Slice 2 (*min(power)*); (c) PRB allocation.

cases. Figure 11b presents that of Slice 2. As expected, the throughput of the *max(rate)* control program instantiated by MVNO 1 on Slice 1 increases with the resources allocated to the slice. On the contrary, the throughput performance of the *min(power)* control program instantiated by MVNO 2 on Slice 2 does not increase with the resources allocated to the slice. All three configurations of Figure 11b converge toward $7\ \mathrm{Mbit/s}$. This is due to the fact that this control problem aims at reaching the minimum per-user rate constraint set by the TO without consuming all available network resources. By looking at Figure 11, we notice that CellOS managed to independently optimize different control problems on different slices of the network (*max(rate* on Slice 1, and *min(power)* on Slice 2). This demonstrates that CellOS provides softwarized MVNOs with independent control of all resources in their leased network slice while sharing the same physical network infrastructure.

4) CellOS Scalability: In this section, we evaluate the scalability of CellOS in terms of time and operations required by the controller to generate distributed solution programs, and by the REEs to solve them. Finally, we compare the overhead generated by CellOS REEs to that of state-of-the-art solutions, such as FlexRAN [13] and Orion [44]. The results presented in this section have been obtained by executing CellOS on a single CPU of a Dell EMC PowerEdge R340 server of the Arena testbed [32]. The server is equipped with an Intel Xeon E-2146G processor with $3.5\ \mathrm{GHz}$ base frequency and $32\ \mathrm{GB}$ DDR4-2666 RAM.

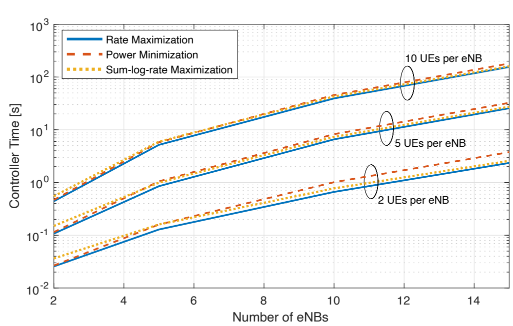


Fig. 12: Scalability of CellOS controller operations as a function of the number of eNBs, UEs and for different network control problems.

Figure 12 shows the time needed by CellOS controller to generate the distributed solution programs starting from the TO directives as a function of the number of network eNBs, UEs, and for different network control problems. This includes the time to perform: (i) The problem definition procedures, which interpret the TO high level directives; (ii) the generation of the centralized version of the problem based on an abstraction of the network, and (iii) the problem decomposition operations, which divide the centralized problem into sub-problems to be distributively solved by the softwarized RAN. We notice that, even though the computation time increases with the number of users and base stations, these operations are executed once per control problem. Also, recall that the generated problems utilize symbolic placeholders and do not require knowledge of real-time parameters. For this reason, all operations can be performed offline, and computation times are thus negligible if compared to the typical service times of cellular networks.

Figure 13 shows the time needed by CellOS REEs to solve the distributed problems automatically generated by the controller (Section III) for different numbers of base stations and UEs in the network. Different control problems require different solution times. For instance, the power minimization problem, whose objective function is a linear function in the transmission power variables, is solved more rapidly than the rate and sum-log-rate maximization problems, whose

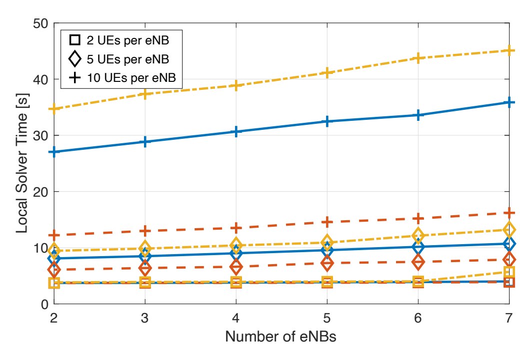


Fig. 13: Scalability of CellOS local solver operations as a function of the number of eNBs, UEs and for different network control problems: (i) Rate maximization (solid lines); (ii) sum-log-rate maximization (dot-dashed lines), and (iii) power minimization (dashed lines).

utility functions are non-linear because of logarithmic and fractional terms, which increase the problem complexity. As a consequence, the execution time of each problem strongly depends on the complexity of the underlying objective function to be optimized. It is worth noticing that the times of both Figures 12 and 13 can be considerably reduced if executed on high-performance equipment, as the one typically used in commercial cellular network deployments.

The signaling overhead generated by each CellOS REE is evaluated in Figure 14 against that generated by other well-established software-defined cellular control frameworks such as FlexRAN [13] and Orion [44]. Since CellOS executes the optimization problems locally at each REE, its overhead stems from the REEs exchanging $|\mathcal{U}|\,(|\mathcal{N}|+1)$ optimization variables and Lagrangian multipliers. These are the only information required to converge to a distributed problem solution (Section III-C). These variables are represented by real numbers encoded as 32-bit floating point numbers. Figure 14

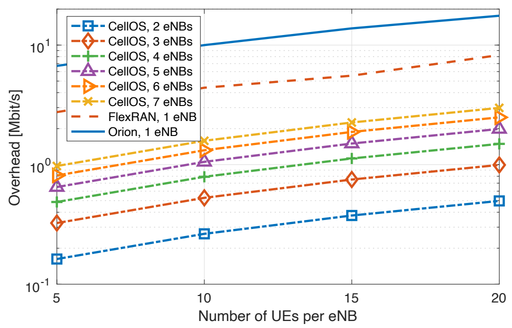
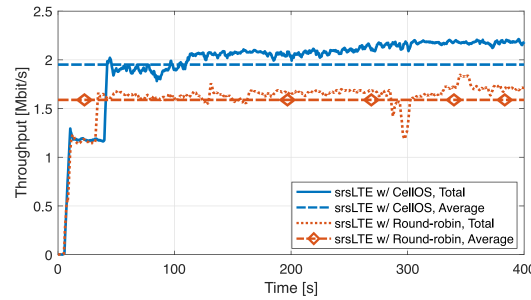


Fig. 14: Signaling overhead: CellOS vs. FlexRAN [13, Figure 7] and Orion [44, Figure 13a].

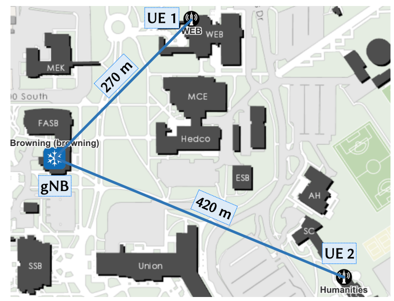
shows that the signaling overhead generated by CellOS REEs is significantly lower than that of prevailing state-of-the-art centralized approaches. Even when managing a single network base station, as it is the case of Figure 14, previous approaches must exchange a massive amount of local information with the central controller, thus generating large signaling and latency.

5) Experiment of POWDER-RENEW PAWR Platform: We demonstrate the platform- and RAN-independence of CellOS by running *long-range* experiments on one of the PAWR wireless platforms [30]. Specifically, we leverage POWDER-RENEW [31] and the 5G implementation of srsLTE to deploy a NR gNB and 2 UEs in an authentic outdoor wireless environment. The gNB employs a USRP X310 located on the rooftop of a 28.75 m-tall building, while we use ground-level USRPs B210 as UEs. The gNB utilizes a reduced channel bandwidth of 15 PRBs (corresponding to 3 MHz) to reach the two UEs distant 270 m and 420 m, respectively (see Figure 15b). In this case, the UEs download a file from a local server for 400 s.

Figure 15a shows the throughput gains achievable by running CellOS rate maximization on top of srsLTE, which uses a *round-robin* scheduler when instantiated without CellOS. Albeit the reduced bandwidth and increased gNB-UEs distance result in a lower total throughput than that of the previous experiments, we notice that CellOS significantly improves the network performance because of its zero-touch approach to



(a)



(b)

Fig. 15: Long-range experiments on the POWDER-RENEW PAWR platform [30, 31]: (a) srsLTE w/ CellOS rate maximization vs. srsLTE w/ *round-robin*; (b) long-range experiment area.

optimization, which allows to optimize the resources allocated to the UEs, and bring gains as high as 86% (23% on average). To the best of our knowledge, this is the first demonstration of *zero-touch optimization* on a long-range open-source 5G testbed. Such instantiation gives evidence of the potential of the *softwarized Open RAN* approach cellular networks are moving toward.

VII. Related Work

Recent years have heralded SDN as the technology that would inherently endow the *monolithic* Internet architecture with much needed *flexibility*. The largest part of SDN work focuses on the *programmability* of wired networks, with few works exploring scenarios comprising wireless devices [12–14, 45–48]. To the best of our knowledge, there is no solution aimed at integrating a zero-touch, flexible, and dynamic optimization framework to the fabric of cellular networks. Therefore, this section reviews SDN-based solutions for wireless networking.

Guan et al. proposed WNOS, a wireless network operating system featuring network virtualization and distributed solution of optimization problems [14]. Although this work is the most similar to ours, it only focuses on infrastructure-less ad hoc networks with static nodes. For this reason, it is not suitable to handle mobile and dynamic cellular scenarios. An effort to explicitly take mobility into account is made by Bertizzolo et al. with SwarmControl [49], a distributed control framework for the self-optimization of drone networks.

ONAP and O-RAN are two infrastructure-oriented automation platforms with the ambition of "orchestrating" many network functions [22, 23]. They offer TOs network abstractions to specify system details and traffic policies. However, optimization policies and algorithms must be explicitly programmed.

Adaptations of the SDN paradigm to cellular networks have been proposed by Li et al. (CellSDN [47]), Bernardos et al. (SDWN [50]), and by Bradai et al. (CSDN [48]). CellSDN proposes a control-oriented operating system focused on cellular network management and subscriber policies rather than on performance optimization. Works like SDWN and CSDN, instead, describe general frameworks to optimize network utilization and performance leveraging edge network information.

Few works have addressed the interplay between the SDN architecture and that of networks including LTE explicitly. Gudipati et al. envision SoftRAN as an abstraction of all eNBs in a geographical area as a single virtual base station to perform operations including metrics optimization [46]. This centralized approach, however, can hardly address heterogeneous optimization problems in the dense, flexible and rapidly growing architecture of 5G cellular networks. Foukas et al. propose FlexRAN [13] and Orion [44] as centralized controllers coordinating various LTE agents, and supporting network slicing, respectively. These systems, though, neglect optimization, and their centralized nature may result in limited scalability and reduce the performance in dense scenarios. Finally, OpenRadio, by Bansal et al., develops a programmable wireless data plane providing programming interfaces on PHY and MAC layers [45]. Optimization, however, is left to the wits of the TO.

Finally, we notice that all the mentioned solutions for cellular networks propose programmable protocol stack implementations where the optimization procedures need to be *manually designed* and there is no way to perform them *dynamically* or *automatically*.

VIII. Conclusions

We presented CellOS, the first zero-touch optimization and management framework for next-generation cellular open RANs. CellOS enables TOs to automatically optimize the *network behavior* through high-level directives without requiring knowledge of optimization theory or of network specifics. CellOS *automatically* generates distributed solution programs to be run at the base stations to simultaneously optimize heterogeneous objectives on different network slices. We prototyped CellOS by using the LTE-compliant OpenAirInterface and srsLTE software, and demonstrated its capabilities through a experimental campaign under varying indoor settings, characterized by different interference conditions and heterogeneous devices. Results indicate that CellOS remarkably improves key performance metrics when compared with existing solutions, including throughput (up to 86%), energy efficiency (up to 84%), and user fairness (up to 29%). Finally, we evaluated CellOS in the outdoor environment of the POWDER-RENEW PAWR 5G platform, providing long-range links. Results from those experiments confirm the effectiveness of CellOS in obtaining superior performance and indicate a new way of managing and optimizing softwarized cellular networks.

References

[1] Radisys. (2019) Open RAN - Enabling the O-RAN of the Future Today. hub.radisys.com/white-papers/open-ran-enabling-the-o-ran-future.

[2] Telecom Infra Project. (2019) OpenRAN: The Next Generation of Radio Access Networks. https://telecominfraproject.com/wp-content/uploads/OpenRAN-v11082019-vFinal.pdf.

[3] S. Kumar, E. Hamed, D. Katabi, and L. E. Li, "LTE Radio Analytics Made Easy and Accessible," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 211–222, August 2014.

[4] D. Gonzalez G., M. Garcia-Lozano, S. Ruiz Boqué, and D. S. Lee, "Optimization of Soft Frequency Reuse for Irregular LTE Macrocellular Networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2410–2423, May 2013.

[5] I. Siomina and D. Yuan, "Analysis of Cell Load Coupling for LTE Network Planning and Optimization," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2287–2297, June 2012.

[6] L. Korowajczuk, *LTE, WiMAX and WLAN Network Design, Optimization and Performance Analysis*. Chichester, United Kingdom: John Wiley & Sons, 2011.

[7] R. Margolies, A. Sridharan, V. Aggarwal, R. Jana, N. K. Shankaranarayanan, V. A. Vaishampayan, and G. Zussman, "Exploiting Mobility in Proportional Fair Cellular Scheduling: Measurements and Algorithms," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 355–367, February 2016.

[8] S. Mair. (2019) AT&T 2019 5G Recap: New Year, New Ways for AT&T Customers to Connect. https://about.att.com/innovationblog/2020/01/2019_5g_recap.html.

[9] O-RAN Alliance. (2020) The O-RAN Alliance and the Telecom Infra Project (TIP) Reach New Level of Collaboration for OpenRadio Access Networks. https://static1.squarespace.com/static/5ad774cce74940d7115044b0/t/5e54df89a726e63147f0b569/1582620555237/2020-02-25_O-RAN-TIP-PR-v1.0.pdf.

[10] ——. (2020) O-RAN Alliance Continues to Grow as Global Operators and Suppliers Reach Across Borders to Collaborate on Open Innovation in Radio Access Networks. https://static1.squarespace.com/static/5ad774cce74940d7115044b0/t/5e4ed59178b98159a8b1f881/1582224786007/2020-02-20_O-RAN+progress+PR_v1.0.pdf.

[11] ——. (2019) O-RAN Alliance Conducts First Global Plugfest to Foster Adoption of Open and Interoperable 5G Radio Access Networks. https://static1.squarespace.com/static/5ad774cce74940d7115044b0/t/5dfba8fb1326ae1bcf4a8b6f/1576773884092/O-RAN-2019.12.19-EC-C-PR-on-2019-Plugfest-v1.0.pdf.

[12] D. Thembelihle, M. Rossi, and D. Munaretto, "Softwarization of Mobile Network Functions Towards Agile and Energy Efficient 5G Architectures: A Survey," *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–21, November 2017.

[13] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks," in *Proc. of ACM SIGCOMM Conf. on emerging Networking EXperiments and Technologies (CoNEXT)*, Irvine, CA, USA, December 2016.

[14] Z. Guan, L. Bertizzolo, E. Demirors, and T. Melodia, "WNOS: An Optimization-based Wireless Network Operating System," in *Proc. of ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Los Angeles, CA, USA, June 2018.

[15] D. Lynch, M. Fenton, D. Fagan, S. Kucera, H. Claussen, and M. O'Neill, "Automated Self-optimization in Heterogeneous Wireless Communications Networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 419–432, February 2019.

[16] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J. S. Pang, "Decomposition by Partial Linearization: Parallel Optimization of Multi-agent Systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 641–656, February 2014.

[17] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and Distributed Methods for Constrained Nonconvex Optimization—Part I: Theory," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 1929–1944, April 2016.

[18] ETSI ZSM. (2019, August) Zero-touch Network and Service Management Reference Architecture. https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf.

[19] 3GPP, "Study on Scenarios and Requirements for Next Generation Access Technologies," TS 38.913, V14.1.0, 2018.

[20] ——, "NR and NG-RAN Overall Description," TS 38.300, V15.0.0, 2018.

[21] 3GPP, "Release description; Release 15," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 21.915, 10 2019, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/21915.htm

[22] The Linux Foundation. (2018) Open Network Automation Platform Architecture. https://www.onap.org/wp-content/uploads/sites/20/2018/11/ONAP_CaseSolution_Architecture_112918FNL.pdf.
[23] O-RAN Alliance. (2018, October) O-RAN: Towards an Open and Smart RAN. https://www.o-ran.org/s/O-RAN-WP-FInal-181017.pdf.
[24] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, March 2018.
[25] F. Kaltenberger, A. P. Silva, A. Gosain, L. Wang, and T.-T. Nguyen, "OpenAirInterface: Democratizing Innovation in the 5G Era," *Computer Networks*, no. 107284, May 2020.
[26] I. Gomez-Miguelez, A. Garcia-Saavedra, P. Sutton, P. Serrano, C. Cano, and D. Leith, "srsLTE: An Open-source Platform for LTE Evolution and Experimentation," in *Proc. of ACM Intl. Workshop onWireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, New York City, NY, USA, October 2016.
[27] S. D'Oro, F. Restuccia, and T. Melodia, "Toward Operator-to-Waveform 5G Radio Access Network Slicing," *IEEE Communications Magazine*, vol. 58, no. 4, April 2020.
[28] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia, "The Slice is Served: Enforcing Radio Access Network Slicing in Virtualized 5G Systems," in *Proc. of Proc. of IEEE Intl. Conf. on Computer Communications (INFOCOM)*, Paris, France, April 2019.
[29] S. D'Oro, L. Bonati, F. Restuccia, M. Polese, M. Zorzi, and T. Melodia, "Sl-EDGE: Network Slicing at the Edge," in *Proc. of ACM Intl. Symp. on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, Shanghai, China, October 2020.
[30] Platforms for Advanced Wireless Research. https://advancedwireless.org.
[31] Platform for Open Wireless Data-driven Experimental Research. https://www.powderwireless.net.
[32] L. Bertizzolo, L. Bonati, E. Demirors, and T. Melodia, "Arena: A 64-antenna SDR-based Ceiling Grid Testbed for Sub-6 GHz Radio Spectrum Research," in *Proc. of ACM Intl. Workshop onWireless Network Testbeds, Experimental evaluation & CHaracterization (WiNTECH)*, Los Cabos, Mexico, October 2019.
[33] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-End Simulation of 5G mmWave Networks," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2237–2263, April 2018.
[34] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead," *arXiv preprint arXiv:2005.10027 [cs.NI]*, May 2020.
[35] 3GPP, "Study on CU-DU Lower Layer Split for NR," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.816, 01 2018, version 15.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/38816.htm
[36] S. D'Oro, P. Mertikopoulos, A. L. Moustakas, and S. Palazzo, "Interference-Based Pricing for Opportunistic Multicarrier Cognitive Radio Systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 12, pp. 6536–6549, December 2015.
[37] D. P. Bertsekas, *Convex Optimization Algorithms*. Nashua, NH, USA: Athena Scientific Belmont, 2015.
[38] J. Lee and S. Leyffer, *Mixed Integer Nonlinear Programming*. New York, NY, USA: Springer Science & Business Media, 2011.
[39] G. Scutari, D. Palomar, and S. Barbarossa, "The MIMO Iterative Waterfilling Algorithm," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 1917–1935, May 2009.
[40] R. K. Jain, D. M. W. Chiu, and E. R. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System," *Eastern Research Laboratory, Digital Equipment Corporation: Hudson, MA, USA*, pp. 2–7, September 1984.
[41] 3GPP, "Telecommunication management;Study on system and functional aspects of energy efficiency in 5G networks," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 32.972, 9 2019, version 16.1.0. [Online]. Available: http://www.3gpp.org/DynaReport/32972.htm
[42] T. Bu, L. E. Li, and R. Ramjee, "Generalized Proportional Fair Scheduling in Third Generation Wireless Data Networks," in *Proc. of Proc. of IEEE Intl. Conf. on Computer Communications (INFOCOM)*, Barcelona, Spain, April 2006.
[43] X. Wu, R. Srikant, and J. R. Perkins, "Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 595–605, June 2007.
[44] X. Foukas, M. Marina, and K. Kontovasilis, "Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture," in *Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, Snowbird, UT, USA, October 2017.
[45] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "OpenRadio: A Programmable Wireless Dataplane," in *Proc. of ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, Helsinki, Finland, August 2012.
[46] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software Defined Radio Access Network," in *Proc. of ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, Hong Kong, China, August 2013.
[47] L. E. Li, M. Z. Mao, and J. Rexford. (2012) CellSDN : Software-defined Cellular Networks. ftp://ftp.cs.princeton.edu/techreports/2012/922.pdf.
[48] A. Bradai, K. Singh, T. Ahmed, and T. Rasheed, "Cellular Software Defined Networking: A Framework," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 36–43, June 2015.
[49] L. Bertizzolo, S. D'Oro, L. Ferranti, L. Bonati, E. Demirors, Z. Guan, T. Melodia, and S. Pudlewski, "SwarmControl: An Automated Distributed Control Framework for Self-Optimizing Drone Networks," in *Proc. of Proc. of IEEE Intl. Conf. on Computer Communications (INFOCOM)*, Toronto, ON, Canada, July 2020.
[50] C. J. Bernardos, A. de la Oliva, P. Serrano, A. Banchs, L. M. Contreras, H. Jin, and J. C. Zuniga, "An Architecture for Software Defined Wireless Networking," *IEEE Wireless Communications Magazine*, vol. 21, no. 3, pp. 52–61, June 2014.

Leonardo Bonati received his B.S. in Information Engineering and his M.S. in Telecommunication Engineering from University of Padova, Italy in 2014 and 2016, respectively. He is currently pursuing a Ph.D. degree in Computer Engineering at Northeastern University, MA, USA. His research interests focus on 5G cellular networks, network slicing, software-defined networking for wireless networks, and unmanned aerial vehicles networks.

Salvatore D'Oro received received his Ph.D. degree from the University of Catania in 2015. He is currently an Associate Research Scientist at Northeastern University. In 2015, 2016 and 2017 he organized the 1st, 2nd and 3rd Workshops on COmpetitive and COoperative Approaches for 5G networks (COCOA). He also served on the Technical Program Committee (TPC) of the IEEE Conference on Standards for Communications and Networking (CSCN'18), Med-Hoc-Net 2018 and the CoCoNet8 workshop at IEEE ICC 2016. He serves on the TPC of Elsevier Computer Communications journal. Dr. D'Oro is also a reviewer for major IEEE and ACM journals and conferences. Dr. D'Oro's research interests include game-theory, optimization, learning and their applications to telecommunication networks. He is a Member of the IEEE.

Lorenzo Bertizzolo is a candidate for Ph.D. in Computer Engineering and research assistant at the Institute for the Wireless Internet of Things at Northeastern University and a collaborator of AT&T Labs Research, working on the integration of Unmanned Aerial System into the next generations' cellular networks. He earned his B.S. and his M.S. in Computer and Communication Networks Engineering from Politecnico di Torino, Italy in 2014 and 2015, respectively. His research focuses on 5G, software-defined networking for wireless networks, distributed optimization, and Unmanned Aerial Networks.

Emrecan Demirors is a Research Assistant Professor with the Department of Electrical and Computer Engineering at Northeastern University. He is conducting research at the Wireless Networks and Embedded Systems Laboratory. Previously, he was an Associate Research Scientist with the Department of Electrical and Computer Engineering at Northeastern University, from 2017 to 2019. He received my Ph.D. degree in Electrical and Computer Engineering from Northeastern University in 2017, under the supervision of Professor Tommaso Melodia. He had previously received my B.S. and M.S degrees in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey in 2009 and 2011, respectively, under the supervision of Professor Hayrettin Koymen. From 2010 to 2011, he was a Systems Engineer at Meteksan Defence Industry Inc., Ankara, Turkey.

Zhangyu Guan is an Assistant Professor with the Department of Electrical Engineering (EE) at The State University of New York at Buffalo (SUNY Buffalo). He received his Ph.D. in Communication and Information Systems from Shandong University in China in 2010. Dr. Guan was a visiting Ph.D. student with the Department of EE, SUNY Buffalo, from 2009 to 2010. He also worked at UB as a Postdoctoral Research Associate from 2012 to 2015. After that, he worked as an Associate Research Scientist with the Department of ECE at Northeastern University in Boston, MA, from 2015 to 2018. He directs the Wireless Intelligent Networking and Security (WINGS) Lab at SUNY Buffalo, with research interests in modeling, control, and system design toward next-generation, intelligent and secure wireless networking.

Stefano Basagni is with the Institute for the Wireless Internet of Things and an associate professor at the ECE Department at Northeastern University, in Boston, MA. He holds a Ph.D. in electrical engineering from the University of Texas at Dallas (December 2001) and a Ph.D. in computer science from the University of Milano, Italy (May 1998). Dr. Basagni's current interests concern research and implementation aspects of mobile networks and wireless communications systems, wireless sensor networking for IoT (underwater and terrestrial), definition and performance evaluation of network protocols and theoretical and practical aspects of distributed algorithms. Dr. Basagni has published over nine dozen of highly cited, refereed technical papers and book chapters. His h-index is currently 44 (June 2020). He is also co-editor of three books. Dr. Basagni served as a guest editor of multiple international ACM/IEEE, Wiley and Elsevier journals. He has been the TPC co-chair of international conferences. He is a distinguished scientist of the ACM, a senior member of the IEEE, and a member of CUR (Council for Undergraduate Education).

Tommaso Melodia is the William Lincoln Smith Chair Professor with the Department of Electrical and Computer Engineering at Northeastern University in Boston. He is also the Founding Director of the Institute for the Wireless Internet of Things and the Director of Research for the PAWR Project Office. He received his Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 2007. He is a recipient of the National Science Foundation CAREER award. Prof. Melodia has served as Associate Editor of IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing, Elsevier Computer Networks, among others. He has served as Technical Program Committee Chair for IEEE Infocom 2018, General Chair for IEEE SECON 2019, ACM Nanocom 2019, and ACM WUWnet 2014. Prof. Melodia is the Director of Research for the Platforms for Advanced Wireless Research (PAWR) Project Office, a $100M public-private partnership to establish 4 city-scale platforms for wireless research to advance the US wireless ecosystem in years to come. Prof. Melodia's research on modeling, optimization, and experimental evaluation of Internet-of-Things and wireless networked systems has been funded by the National Science Foundation, the Air Force Research Laboratory the Office of Naval Research, DARPA, and the Army Research Laboratory. Prof. Melodia is a Fellow of the IEEE and a Senior Member of the ACM.