Received 11 January 2024; revised 1 March 2024; accepted 21 March 2024. Date of publication 1 April 2024; date of current version 24 April 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3383607

On Robust Optimal Joint Deployment and Assignment of RAN Intelligent Controllers in O-RANs

MOHAMMAD J. ABDEL-RAHMAN[1] (Senior Member, IEEE),
EMADELDIN A. MAZIED[2,3] (Graduate Student Member, IEEE), FAHID HASSAN[4],
KORY TEAGUE[5] (Member, IEEE), ALLEN B. MACKENZIE[6] (Senior Member, IEEE),
SCOTT F. MIDKIFF[5] (Life Senior Member, IEEE), KLEBER V. CARDOSO[7],
AND DIMITRIOS S. NIKOLOPOULOS[2] (Fellow, IEEE)

[1]Data Science Department, Princess Sumaya University for Technology, Amman 11941, Jordan

[2]Computer Science Department, Virginia Tech, Blacksburg, VA 24061, USA

[3]Electrical Engineering Department, Sohag University, New Sohag City 82524, Egypt

[4]Electrical and Computer Engineering Department, Rice University, Houston, TX 77005, USA

[5]Electrical and Computer Engineering Department, Virginia Tech, Blacksburg, VA 24061, USA

[6]Electrical and Computer Engineering Department, Tennessee Tech, Cookeville, TN 38505, USA

[7]Instituto de Informática, Universidade Federal de Goiás, Goiânia 74690, Brazil

CORRESPONDING AUTHOR: M. J. ABDEL-RAHMAN (e-mail: mo7ammad@vt.edu)

This work was supported in part by NSF under Award NSF-CNS-2315851 and Award NSF-CNS-2106634; in part by the Sony Faculty Innovation Award; in part by the Cisco Research Award; and in part by FAPESP/MCTI/MCom/CGI.br under Grant 2020/05127-2.

ABSTRACT The open radio access network (O-RAN) architecture is consolidating the concept of software-defined cellular networks beyond 5G networks, mainly through the introduction of the near-real-time radio access network (RAN) intelligent controller (Near-RT RIC) and the xApps. The deployment of the Near-RT RICs and the assignment of RAN nodes to the deployed RICs play a crucial role in optimizing the performance of O-RANs. In this paper, we develop a robust optimization framework for joint RIC deployment and assignment, considering the uncertainty in user locations. Specifically, our contributions are as follows. First, we develop $\mathrm{C^3P^2}$, a *robust static* joint RIC placement and RAN node-RIC assignment scheme. The objective of $\mathrm{C^3P^2}$ is to minimize the number of RICs needed to control all RAN nodes while ensuring that the response time to each RAN node will not exceed δ milliseconds with a probability greater than β. Second, we develop CPPA, a *robust* joint RIC placement and *adaptive* RAN node-RIC assignment scheme. In contrast to $\mathrm{C^3P^2}$, CPPA enjoys a *recourse* capability, where the RAN node-RIC assignment adapts to the variations in the user locations. We use chance-constrained stochastic optimization combined with several linearization techniques to develop a mixed-integer linear (MIL) formulation for $\mathrm{C^3P^2}$. Two-stage stochastic optimization with recourse, combined with several linearization techniques, is used to develop an MIL formulation for CPPA. The optimal performance of $\mathrm{C^3P^2}$ and CPPA has been examined under various system parameter values. Furthermore, sample average approximation has been employed to design efficient approximate algorithms for solving $\mathrm{C^3P^2}$ and CPPA. Our results demonstrate the robustness of the proposed stochastic resource allocation schemes for O-RANs compared to existing deterministic allocation schemes. They also show the merits of adapting the allocation of resources to the network uncertainties compared to statically allocating them.

INDEX TERMS Open radio access networks (O-RANs), RAN intelligent controller (RIC) placement, RIC assignment, chance-constrained stochastic optimization, two-stage stochastic optimization with recourse, sample average approximation.

I. INTRODUCTION

THE OPEN radio access network (O-RAN) Alliance [1] has emerged as a transformative force in the radio access network (RAN) industry, pioneering a shift towards open, virtualized, interoperable, and intelligent mobile cellular networks. At the forefront of O-RAN's groundbreaking specifications lies its innovative RAN architecture, a pivotal contribution that integrates principles from software-defined networking (SDN) and network functions virtualization (NFV), alongside cloud-native and artificial intelligence (AI) / machine learning (ML) technologies [2]. Drawing inspiration from SDN, the O-RAN architecture [3] incorporates concepts such as the separation of control and data planes, coupled with the adoption of a remote RAN controller. Crucially, this architecture divides the controller into two core blocks: the near-real-time RAN intelligent controller (Near-RT RIC) handling time-sensitive operations, and the non-real-time RAN intelligent controller (Non-RT RIC) catering to operations with more relaxed time constraints. This pioneering approach marks a significant evolution in RAN design and promises to reshape the landscape of mobile cellular networks.

A RIC operates AI/ML-based applications, establishing control loops with the RAN nodes under its jurisdiction. The Non-RT RIC hosts applications, referred to as rApps, capable of handling control loop latency exceeding 1 second. Conversely, the Near-RT RIC manages applications known as xApps, which implement control loops constrained to time intervals between 10 milliseconds and 1 second. The responsiveness of a control loop is intricately tied to the specific RAN function governed by the associated xApp.

In extensive RAN deployments, the NearRT RIC and its latency-sensitive xApps require replication, with a designated allocation of RAN nodes for each RIC. The strategic placement of RICs plays a pivotal role in minimizing response time. This optimization challenge, known as the controller placement problem (CPP), involves distributing a minimum number of RICs to optimal locations to ensure timely completion of control functions. While extensively studied in wired networks (examples include [4], [5], [6], [7], [8], [9], [10], [11], [12]), the CPP presents additional complexities in the dynamic context of mobile wireless networks. These complexities are compounded by considerations such as network latency, reliability, and load balancing, rendering the CPP in cellular networks a multifaceted and challenging undertaking.

User mobility stands out as a pivotal factor in cellular networks, introducing a distinctive challenge to the CPP in software-defined cellular networks (SDCNs). The advent of SDCNs aimed to facilitate flexible cellular network designs, aligning with the evolving requirements of 5G and beyond. User mobility complicates the CPP, especially in the context of SDCNs. This introduces an additional layer of uncertainty, as the distribution of mobile users across RAN nodes becomes stochastic. Consequently, the request rates from the RAN nodes to the RAN controllers become unpredictable, adding a dynamic dimension to the challenge posed by user mobility in the CPP.

Several studies have been proposed to address the congestion-aware[1] CPP in SDCN, with a focus on load balancing as a crucial design metric for the control plane (examples include [13], [14], [15], [16], [17]). However, these endeavors did not incorporate considerations for the uncertainty in users' mobility patterns, which results in variations in the control plane workload. Stochastic programming methods have proven to be powerful mathematical tools for optimizing resource allocation in various types of wireless networks operating under uncertainties (examples include [18], [19], [20], [21]). Therefore, this paper adopts chance-constrained and two-stage stochastic programming [22] to address the CPP, specifically taking into account the uncertainty in users' mobility patterns leading to variations in control workload.

Main Contributions: In this paper, we develop a robust optimization framework for joint RIC deployment and assignment, while considering the uncertainty in the geographical distribution of mobile users (and hence, RAN node request rates). Specifically:

- We develop a *robust static* joint RIC placement and assignment scheme (denoted by C^3P^2): Using chance-constrained stochastic programming, we formulate a static joint stochastic RIC placement and RAN node-RIC assignment problem that is robust to the variations in the mobile user locations (and hence the RAN node request rates). The objective of C^3P^2 is to minimize the number of RICs needed to control all RAN nodes while ensuring that the response time to each RAN node will exceed δ milliseconds with a probability less than $1-\beta$.
- We develop a *robust* joint RIC placement and *adaptive* assignment scheme (denoted by CPPA): Using two-stage stochastic programming, we formulate a joint stochastic RIC placement and adaptive RAN node-RIC assignment problem. In contrast to C^3P^2, where the RAN node-RIC assignment is static, CPPA enjoys a *recourse* capability, where the RAN node-RIC assignment adapts to the variations in the RAN node request rates, resulting from variations in mobile user locations. The goal of the CPPA first stage is to optimally place the minimum number of RICs. Our optimality criteria are: (i) minimizing the number of RICs and (ii) minimizing the response time to various RAN nodes. In contrast to C^3P^2, CPPA does not ensure that the RAN node response time constraints are satisfied with a minimum probability of β. The first-stage problem decision is static and is taken before knowing which realization of RAN node request rates will occur. In the CPPA second stage, the RAN node-RIC assignment is optimized under each realization of RAN node request rates aiming at minimizing the response time to various RAN nodes.

[1] We use "congestion-aware" to describe a network with variations in workload arising from uncertainties in users' activities.

- To evaluate and study our proposed C^{3}P^2 and CPPA schemes, we adopt a sample average approximation (SAA) framework, in which:
  - We use Monte Carlo sampling to generate independent and identically distributed (i.i.d.) samples of the RAN node request rates.
  - We use the generated samples to derive deterministic equivalent programs that represent sampled versions of C^{3}P^2 and CPPA, which only account for the set of generated scenarios (i.e., samples).
  - We use several linearization techniques to convert these deterministic equivalent programs into mixed-integer linear programs (MILPs).
  - We solve the MILPs using IBM ILOG CPLEX optimization studio. CPLEX is a prescriptive analytics solution that enables rapid development and deployment of decision optimization models using mathematical and constraint programming [23].
  - Finally, we statistically estimate the optimality gap of our proposed SAA framework for solving the chance-constrained and the two-stage stochastic programming formulations of C^{3}P^2 and CPPA, respectively.

Paper Organization:

The rest of the paper is organized as follows. The literature review is given in Section II. The considered models and assumptions are stated in Section III, followed by our problem statement. In Section IV, C^{3}P^2 is presented, mathematically formulated, and reformulated as an MILP. An SAA-based algorithm is also developed in Section IV to efficiently solve C^{3}P^2. In Section V, CPPA is presented, mathematically formulated, and reformulated as an MILP. Furthermore, an SAA-based algorithm is developed in Section V to solve CPPA efficiently. C^{3}P^2 and CPPA are extensively evaluated in Section VI. Finally, we conclude the paper in Section VII. The main notations used in this paper are summarized in Table 1. The main abbreviations used in the paper are summarized in Table 7 in Appendix A.

TABLE 1. Notation.

Sets:

Notation	Description
$\mathcal{B}$	Set of RAN node forming the considered mobile network.
$\mathcal{C}$	Set of candidate locations for deploying RICs.

Data:

Notation	Description
k	Request rate of each mobile user.
$\tilde{r}_b$	Request rate of RAN node b.
t_{bc}	Sum of transmission and propagation delays over the link between RAN node b and a RIC at location c.
μ	RIC service rate (a.k.a. processing capacity).
β	Requested minimum probability of delay satisfaction.
α	Minimum probability of delay satisfaction that the sample average approximation approach aims to achieve.
δ	Requested upper-bound on the RICs' response time.
Ω	A set of i.i.d. samples (scenarios) generated from the distribution of the RAN node request rates.
q_{bc}	A design coefficient for the link between RAN node b and a RIC at location c, introduced to balance the tradeoff between minimizing the number of RICs and minimizing the response time to the RAN nodes.

Decision Variables:

Notation	Description
x_{bc}	A binary decision variables; it equals one if a RIC is placed at location c to control RAN node b, and it equals zero otherwise.
y_c	A binary decision variables; it equals one if a RIC is placed at location c, and it equals zero otherwise.
$u_{bc}^{(\omega)}$	A binary decision variable; it equals zero if the response time of RIC c to RAN node b under scenario ω is less than δ, and it equals one otherwise.
$x_{b\acute{b}c}$	$x_{bc}\ x_{\acute{b}c}$.
$z_{b\acute{b}c}^{(\omega)}$	$u_{bc}\ x_{\acute{b}c}$.
v_{bc}	A positive decision variable; it equals max (delay over the link bc, δ).
f_{bc}	A binary decision variable; it equals one if the response time over the link bc is greater than δ, and it equals zero otherwise.
d_{bc}	$1 - f_{bc}$.
$f_{b\acute{b}c}$	$x_{\acute{b}c}\ f_{bc}$.
$e_{b\acute{b}c}$	$x_{\acute{b}c}\ v_{bc}$.

II. LITERATURE REVIEW

Numerous research endeavors have delved into the intricacies of the SDN CPP and the associated node-controller assignment problem. These investigations initially explored the CPP in the context of wired networks design [4], [5], [6], [7], [8], [9], [10], [24], [25], [26], [27], with emphasis on facility location analysis. Emerging trends indicate a growing interest in applying CPP principles to next-generation wireless networks, particularly in the O-RAN domain [28]. In the wireless domain, the CPP has been applied to address challenges in mobile networks with dynamic topologies and high mobility scenarios [13], [14], [15], [16], [17], [28], [29], [30], [31].

The next-generation wireless networks, driven by SDN control for O-RAN design [28], have prompted research on the CPP in wireless networks. Notable works include [13], [14], [15], [16], [17], [28], [29], [30], [31]. In [13], a two-tier CPP, using mixed-integer linear programming, optimized controller placement and node assignment according to the variations in the data rate of traffic flows. Likewise, in [29] the authors introduced two CPP schemes, two-tier leader-based and single-tier leaderless control planes, deploying them interchangeably based on network load. The authors in [31] used simulated annealing for joint controller and satellite gateway placement. An ant colony with an external memory CPP algorithm was proposed in [14] and validated against particle swarm optimization (PSO) for load balancing. In [32], the authors explored the load-balancing CPP with a predetermined number of controllers, addressing proactive and reactive controller-node assignments. In [15], a two-stage framework was developed to optimize energy consumption and task allocation using PSO and deep reinforcement learning. The authors in [16] considered a dynamic topology and developed a branch-and-cut algorithm for optimal controller placement. Clock synchronization was integrated for reliability in a multi-objective CPP in [17]. The work in [28] focused on placing disaggregated SDN-like control functions using a greedy approach. Additionally, the authors in [18] conducted the first study of the wireless CPP, considering uncertainties in wireless links using chance-constrained stochastic optimization.

The aforementioned works have introduced novel placement optimization approaches, using heuristic methods and deterministic (exact) optimization algorithms, to tackle specific requirements of wireless networks. They have considered a spectrum of critical design metrics such as deployment cost (i.e., the number of controllers), latency, load balancing, controller capacity, reliability, resilience, and energy consumption to formulate optimal placement policies.

The majority of CPP research has focused on minimizing the controller deployment cost while achieving low latency [4], [6], [7], [9], [13], [14], [16], [17], [24], [25], [26], [27], [29]. Other works expanded their scope to include load balancing among distributed controllers and considerations of computational and storage power [6], [7], [13], [14], [16], [17], [25], [26]. Conversely, some works assumed a predefined number of controllers and aimed to develop optimal placement policies while minimizing energy consumption, considering delay and load balancing constraints [15].

The CPP research commonly conceptualized the system as an undirected graph $G(V, E)$, where the set of candidate locations for controllers surrounded controlled nodes V. These locations were modeled as a subset of V connected to controlled nodes via wired links represented as the set of edges E in the network graph [4], [5], [6], [7], [8], [9], [10], [13], [16], [24], [25], [26], [27], [29], [31], [32], [28]. Alternatively, some works adopted set theory for system modeling [14], [15], defining pairs of device-to-device wireless links, wireless network entities, and SDN controllers using various sets.

These CPPs were often formulated based on the analysis of facility location problems and their variants, including k-median (k-center) problems [4], [5], [8], [25], [29], capacitated facility location problems, and set covering problems [6], [7]. Additionally, Pareto optimal multi-objective optimization [9], [17], fault-tolerant facility location problems (k-terminal network reliability problem) [10], bin packing problems [13], k-means graph clustering [24], constraint covering graphs [26], integer linear optimization problems [16], [27], and two-stage joint optimization [14], [15], [31] have been employed for CPP formulations.

Overall, the CPP, categorized as an NP-hard optimization problem, has been approached with various approximation algorithms and heuristic methods across different network architectures, ranging from wired to wireless. The literature underscores two main categories of optimal placement methods: heuristic approaches [4], [5], [7], [9], [10], [14], [15], [17], [25], [26], [27], [28], [30], [31] and deterministic approaches integrating approximation algorithms [6], [8], [13], [16], [24], [29], [31], [32].

The diverse range of methodologies and applications within the CPP literature highlights the importance of its role in optimizing software-defined control plane designs for next-generation networks. We refer the reader to [33], [34], [35] for more details regarding the CPP problems and methods.

Figure 1 presents a comprehensive overview of the recent related studies on the CPP in wireless networks, encompassing diverse design metrics within both single-tier and multi-tier control plane hierarchies. In the single-tier configuration, geo-distributed controllers are modeled, representing one-hop wired connections to controlled nodes and to other controllers. On the other hand, the multi-tier control plane introduces two distinct control domains, namely global and local, as discussed in works such as [13], [29]. The classification of CPP in wireless networks accounts for both wired and wireless connections between the controllers and controlled nodes [18].

To the best of our knowledge, all existing works on CPP, except [18], have not adequately addressed various system uncertainties, particularly in the realm of wireless and mobile SDNs. The study by the authors in [18] marked the pioneering exploration of the ‘wireless CPP,’ where the links between the controllers and controlled nodes are wireless, and they systematically considered uncertainties associated with these links. Employing chance-constrained stochastic optimization [22], they optimized the placement of the controllers and their assignment to the controlled elements, accounting for uncertainties in the availability of the wireless links.

In this paper, we extend our previously proposed framework from [20] for the CPP in SDCNs, where users are mobile, and their distribution across cells in the mobile cellular network is subject to time-varying dynamics. Additionally, we refine the terminology to enhance the clarity and comprehension of our work within the context of O-RANs.

III. MODELS, ASSUMPTIONS, AND PROBLEM STATEMENT

A. SYSTEM MODEL

We consider a set $\mathcal{B} = \{1, 2, \ldots, B\}$ of RAN nodes forming a cellular network, and a set $C = \{1, 2, \ldots, C\}$ of candidate locations for deploying SDN (or RAN) controllers to control the RAN nodes. According to O-RAN standards [36], a

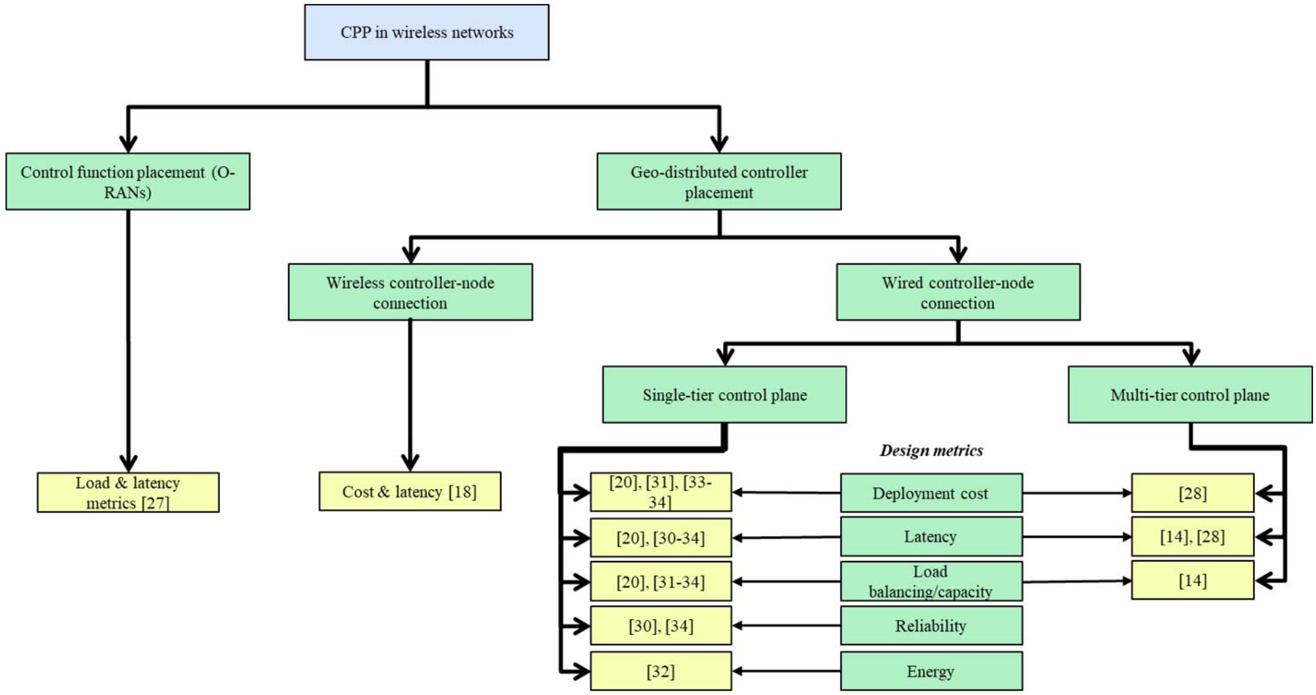


FIGURE 1. Taxonomy of the recent related studies of the CPP in wireless networks.

RAN node can be an O-RAN-compliant LTE evolved node B (eNB) or a component of a next-generation node B (gNB) (i.e., a central unit, CU, or a distributed unit, DU). A RAN node is also commonly called an E2 node due to the E2 interface used to manage the RAN node [36]. A simplified view of our system is depicted in Figure 2 for $B = 9$ and $C = 4$. The RAN controllers can be connected to the RAN nodes through wired or wireless links, as explained in [37]. In this paper, we consider wired links between the RAN nodes and the controllers. Multiple RAN nodes can be controlled by the same RAN controller.

We assume that each mobile user has a request rate of k requests/second. The locations of the mobile users are time-varying and can be modeled as a stochastic process. Hence, at a given time instant, we model the request rate of RAN node b, defined as the number of users served by RAN node b at that time instant multiplied by k, as a stochastic variable, denoted by $\tilde{r}_b$.

B. DISTRIBUTION OF RAN NODE REQUEST RATES

To know the distribution of $\tilde{r}_b, \forall b \in \mathcal{B}$, we want to know the distribution of the mobile user locations. In [38], the authors concluded that the traffic density, defined as the traffic demand per unit area, in a cellular network closely follows a log-normal distribution with spatially correlated characteristics. Supported by this, the authors in [39] proposed a spatial model of scalable, spatially correlated, and log-normally distributed traffic (SSLT). By controlling its parameters, the SSLT model is capable of generating a stochastic traffic distribution over an intended area which captures the spatially-correlated and inhomogeneous characteristics of traffic within a cellular network.

As proposed in [39], the model operates by defining a grid of points, each of which takes a value corresponding to a two-dimensional, spatially-correlated log-normal function. In [39], each point represents a user, or a collection of users within a pixel surrounding the point, with the specified demand. It is assumed in [39] that each user is equally spaced with a log-normally distributed demand. Instead, in this paper, we assume that each user (also represented by a point) has a constant demand and is placed according to the log-normal density function.[2] The model is extended, removing the grid of points and maintaining the model as a continuous density function, denoted by $\lambda(x, y)$, which acts as the parameter for a two-dimensional, non-stationary (inhomogeneous) Poisson point process (PPP). From the model, $\lambda(x, y)$ is defined as:

$$\lambda(x, y) = e^{\sigma \rho^{(S)}(x,y)+\gamma}, \tag{1}$$

where σ and γ are the scaling and location parameters, respectively, of the log-normal distribution, and $\rho^{(S)}(x, y)$ is the standardized version of $\rho^{(G)}(x, y)$, the Gaussian stochastic field. $\rho^{(G)}(x, y)$ is given by:

$$\rho^{(G)}(x, y) = \frac{1}{L} \sum_{l=1}^{L} \cos(i_l\, x + \phi_l) \cos(j_l\, y + \psi_l), \tag{2}$$

where angular frequencies i_l and j_l are uniform stochastic variables between 0 and $\omega_{\max}$, and phases ϕ_l and ψ_l are

[2] Our work can be easily extended to the case when user demands are time-varying.

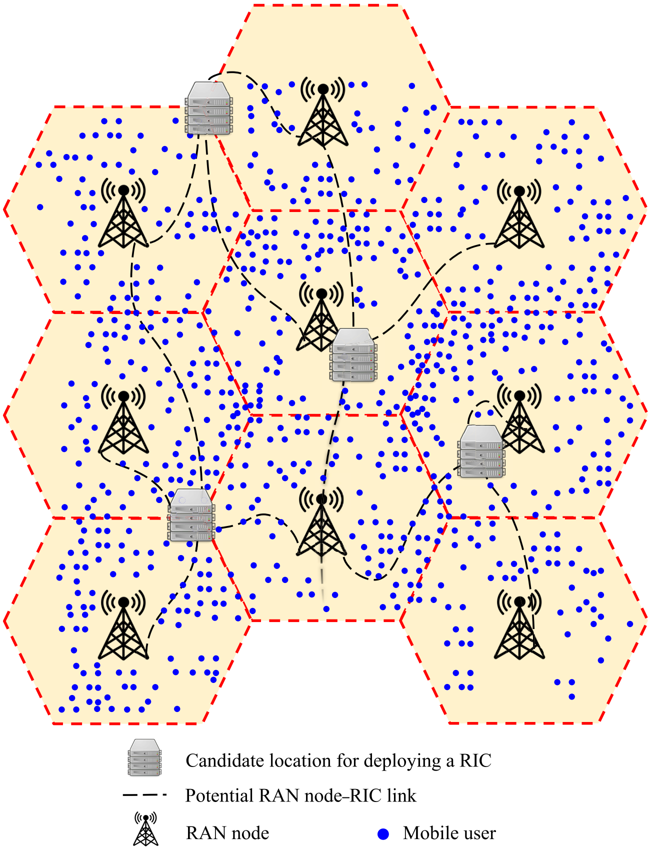

Candidate location for deploying a RIC
Potential RAN node–RIC link
RAN node
Mobile user

FIGURE 2. System model ($B = 9$ and $C = 4$). The locations of the mobile users form a realization of a Poisson point process.

uniform stochastic variables between 0 and 2π. L is the number of stochastic sinusoidal fields used to generate $\rho^{(G)}(x, y)$; for a sufficiently large L, $\rho^{(G)}(x, y)$ can be approximated as a Gaussian stochastic variable.

Each user is positioned according to a non-stationary PPP with parameter $\lambda(x, y)$. The PPP is generated via a trimming method. A stationary PPP with parameter $\lambda_{\max}$, the maximum value of $\lambda(x, y)$ over the considered domain, is first generated. Then, each point (x_i, y_i) of the generated PPP is kept only with probability $\lambda(x_i, y_i)/\lambda_{\max}$. The number of remaining PPP points within the coverage area of RAN node b (after trimming) multiplied by k represents the request rate of RAN node b, i.e., $\tilde{r}_b$.

C. QUEUING DELAY AT THE RICS

In addition to the transmission (and propagation) delays, given by $2t_{bc}$ for the link between RAN node b and RIC c, a RAN node will encounter a queuing delay at the RIC.[3] We model each RIC as an M/M/1 queuing system [40], under which the mean delay (say, at RIC c) can be expressed

TABLE 2. C^{3}P^2 vs. CPPA.

Criterion	C^{3}P^2	CPPA
Delay satisfaction	Response time to each RAN node is guaranteed to be less than or equal to δ milliseconds with a probability greater than β.	Response time is minimized, however no probabilistic guarantees on being equal to δ milliseconds. Also, the response time is prevented from being smaller than δ milliseconds.
Adaptability	The RAN node-RIC assignment and the RIC deployment are fixed.	The RAN node-RIC assignment adapts to the distribution of mobile users (and hence the RAN node request rates). The RIC deployment is fixed.
Complexity	Has lower complexity than CPPA.	Has higher complexity than C^{3}P^2.

as [41]:

$$\mathbb{E}[D_c] = \frac{1}{\mu - \sum_{\left\{\substack{b\in\mathcal{B}:\\ b \text{ assigned to } c}\right\}} \tilde{r}_b}, \tag{3}$$

where μ is the RIC service rate (a.k.a. RIC processing capacity) and $\tilde{r}_b$ is the request rate of RAN node b.[4]

D. PROBLEM STATEMENT

Our objective in this paper is to find *the minimum number of RICs, their optimal locations, and the optimal assignment of these RICs to the RAN nodes, where the optimality criteria are based on satisfying the RAN nodes' delay requirements.* To formulate our problem, we introduce x_{bc}, $b \in \mathcal{B}$, $c \in \mathcal{C}$, as binary decision variables; x_{bc} equals one if a RIC is placed at location c to control RAN node b, and equals zero otherwise.

In the following sections, we develop two schemes for addressing the problem stated above. We refer to our proposed schemes by C^{3}P^2 and CPPA. Table 2 summarizes the main differences between C^{3}P^2 and CPPA.

IV. ROBUST STATIC JOINT RIC PLACEMENT AND RAN NODE-RIC ASSIGNMENT (C^{3}P^2)

In this section, we present our *static* joint RIC placement and RAN node-RIC assignment scheme, referred to as C^{3}P^2.

[3] t_{bc} for the transmission and propagation delays from RAN node b to RIC c and t_{bc} for the transmission and propagation delays from RIC c to RAN node b.

[4] Note that μ of each RIC needs to be greater than the total request rate from all RAN nodes assigned to that RIC, i.e., the denominator of (3) needs to be positive for all $c \in \mathcal{C}$.

A. PROBLEM FORMULATION

C^{3}P^2 aims to find the optimal placement of the minimum number of RICs and the optimal assignment of these RICs to the RAN nodes, to ensure that the response time to each RAN node will exceed δ milliseconds with probability less than $1-\beta$. We mathematically formulate C^{3}P^2 using chance-constrained stochastic optimization [22]. We formulate C^{3}P^2 while considering the uncertainty in $\tilde{r}_b, b \in \mathcal{B}$. We say that C^{3}P^2 is a robust scheme; because it accounts for the variations in $\tilde{r}_b, b \in \mathcal{B}$, resulting from variations in the mobile user locations. Mathematically, C^{3}P^2 can be stated as:

Problem 1 (C^{3}P^2):

$$\underset{\{x_{bc}, b\in\mathcal{B}, c\in\mathcal{C}\}}{\text{minimize}} \sum_{c\in\mathcal{C}} \mathbb{1}_{\{\sum_{b\in\mathcal{B}} x_{bc} \geq 1\}} \tag{4}$$

subject to:

$$\sum_{c\in\mathcal{C}} x_{bc} = 1, \forall b \in \mathcal{B}, \tag{5}$$

$$\Pr\left\{2\ t_{bc}\ x_{bc} + \frac{1}{\mu - \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c}} \leq \delta\right\} \geq \beta, \quad \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \tag{6}$$

$$\Pr\left\{\sum_{b\in\mathcal{B}} \tilde{r}_b\ x_{bc} \leq \mu\right\} = 1, \forall c \in \mathcal{C}, \tag{7}$$

$$x_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \tag{8}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function, which equals one if condition $\{\cdot\}$ is satisfied and equals zero otherwise.

B. C^{3}P^2 WITH SAMPLED REQUEST RATE DISTRIBUTION

Chance-constrained stochastic programs are largely intractable due to the difficulty in checking the feasibility of a particular solution [42]. In other words, for a given $x_{bc}, b \in \mathcal{B}, c \in \mathcal{C}$, computing $\Pr\{2t_{bc}x_{bc} + \frac{1}{\mu - \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}} x_{\acute{b}c}} \leq \delta\}$ accurately is hard. One standard technique for addressing this difficulty in solving chance-constrained stochastic programs is sampling. The basic idea is to approximate the true distribution of stochastic variables with an empirical distribution by sampling. We generate a set Ω of i.i.d. samples (scenarios) from the distribution of the RAN node request rates, described in Section III-B, using Monte Carlo simulation. After generating the scenarios, the chance constraint can be estimated using an indicator function as $|\Omega|^{-1} \sum_{\omega\in\Omega} \mathbb{1}_{\{2\ t_{bc}\ x_{bc} + \frac{1}{\mu - \sum_{\acute{b}\in\mathcal{B}} r_{\acute{b}}^{(\omega)}\ x_{\acute{b}c}} \leq \delta\}} \geq \beta$, where $r_b^{(\omega)}$ is the request rate of RAN node b under scenario ω. C^{3}P^2 with sampled request rate distribution is given by:

C^{3}P^2 With Sampled Request Rate Distribution:

$$\underset{\{x_{bc}, b\in\mathcal{B}, c\in\mathcal{C}\}}{\text{minimize}} \sum_{c\in\mathcal{C}} \mathbb{1}_{\{\sum_{b\in\mathcal{B}} x_{bc} \geq 1\}} \tag{9}$$

subject to:

$$\sum_{c\in\mathcal{C}} x_{bc} = 1, \forall b \in \mathcal{B} \tag{10}$$

$$\frac{1}{|\Omega|} \sum_{\omega\in\Omega} \mathbb{1}\left\{2\ t_{bc}\ x_{bc} + \frac{1}{\mu - \sum_{\acute{b}\in\mathcal{B}} r_{\acute{b}}^{(\omega)}\ x_{\acute{b}c}} \leq \delta\right\} \geq \alpha, \quad \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \tag{11}$$

$$\sum_{b\in\mathcal{B}} r_b^{(\omega)}\ x_{bc} \leq \mu, \forall c \in \mathcal{C}, \forall \omega \in \Omega, \tag{12}$$

$$x_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \tag{13}$$

where $\alpha \in (0, 1]$ and it can be different from β. There are several advantages for solving the sampled version of C^{3}P^2, as summarized in [20]. One of these advantages is to get a lower bound on the required number of RICs. Specifically, if o_β^* and $\hat{o}_\alpha$ are the optimal objective function values (i.e., the minimum number of RICs) of C^{3}P^2 and its sampled version, respectively. Then, it has been shown that $\hat{o}_\alpha \leq o_\beta^*$ with probability at least $1-\eta$ if $|\Omega| \geq \frac{1}{2(\beta-\alpha)^2} \ln(\frac{1}{\eta})$ and $\alpha < \beta$ [43].[5] Note that the minimum number of scenarios needed for the result to hold depends on η and $\beta - \alpha$. It increases as η decreases, and it also increases as α gets closer to β.

In the next subsection, we use several linearization techniques to convert the sampled version of C^{3}P^2 into a mixed-integer linear program (MILP), in order to solve it using CPLEX.

C. MIXED-INTEGER LINEAR REFORMULATION

The mathematical formulation of C^{3}P^2 developed in the previous subsections is non-linear. In this subsection, we present an *equivalent* linear reformulation of C^{3}P^2. The mathematical details of deriving this linear reformulation of C^{3}P^2 are explained in Appendix B.

The sampled version of C^{3}P^2 can be equivalently written as an MILP as follows:

MILP Reformulation of the Sampled Version of C^{3}P^2:

$$\underset{\left\{\substack{x_{bc}, y_c, x_{b\acute{b}c}, \\ z_{b\acute{b}c}^{(\omega)}, u_{bc}^{(\omega)}, \\ b, \acute{b}\in\mathcal{B}, c\in\mathcal{C}, \omega\in\Omega}\right\}}{\text{minimize}} \sum_{c\in\mathcal{C}} y_c \tag{14}$$

subject to:

$$\sum_{c\in\mathcal{C}} x_{bc} = 1, \forall b \in \mathcal{B}, \tag{15}$$

$$\sum_{b\in\mathcal{B}} x_{bc} \leq B\ y_c, \forall c \in \mathcal{C}, \tag{16}$$

$$\begin{aligned} &2\ \mu\ t_{bc}\ x_{bc} - 2\ t_{bc} \sum_{\acute{b}\in\mathcal{B}} r_{\acute{b}}^{(\omega)}\ x_{b\acute{b}c} - \mu\ N_{bc}^{(\omega)}\ u_{bc}^{(\omega)} \\ &\quad + N_{bc}^{(\omega)} \sum_{\acute{b}\in\mathcal{B}} r_{\acute{b}}^{(\omega)}\ z_{b\acute{b}c}^{(\omega)} + (\delta - \epsilon) \sum_{\acute{b}\in\mathcal{B}} r_{\acute{b}}^{(\omega)}\ x_{\acute{b}c} \\ &\quad \leq \mu(\delta - \epsilon) - 1, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega, \end{aligned} \tag{17}$$

[5] For example, if we select α to be smaller than β by 0.15, then solving the sampled version of C^{3}P^2 with $|\Omega| \geq \frac{1}{2\times 0.15^2} \ln \frac{1}{0.1} = 51$ scenarios will provide a lower bound on the required number of RICs for C^{3}P^2 with probability at least 0.9. In Section VI, we solve the sampled version of C^{3}P^2 with $|\Omega| = 100$ scenarios.

$$\sum_{\omega\in\Omega}\left(1-u_{bc}^{(\omega)}\right) \geq \alpha\ |\Omega|, \forall b \in \mathcal{B}, \forall c \in C, \tag{18}$$

$$x_{b\acute{b}c} \leq x_{bc}, \forall b, \acute{b} \in \mathcal{B}, \forall c \in C, \tag{19}$$

$$x_{b\acute{b}c} \leq x_{\acute{b}c}, \forall b, \acute{b} \in \mathcal{B}, \forall c \in C, \tag{20}$$

$$x_{b\acute{b}c} \geq x_{bc} + x_{\acute{b}c} - 1, \forall b, \acute{b} \in \mathcal{B}, \forall c \in C, \tag{21}$$

$$z_{b\acute{b}c}^{(\omega)} \leq u_{bc}^{(\omega)}, \forall b, \acute{b} \in \mathcal{B}, \forall c \in C, \forall \omega \in \Omega, \tag{22}$$

$$z_{b\acute{b}c}^{(\omega)} \leq x_{\acute{b}c}, \forall b, \acute{b} \in \mathcal{B}, \forall c \in C, \forall \omega \in \Omega, \tag{23}$$

$$z_{b\acute{b}c}^{(\omega)} \geq u_{bc}^{(\omega)} + x_{\acute{b}c} - 1, \quad \forall b, \acute{b} \in \mathcal{B}, \forall c \in C, \forall \omega \in \Omega, \tag{24}$$

$$\sum_{b\in\mathcal{B}} r_b^{(\omega)}\ x_{bc} \leq \mu, \forall c \in C, \forall \omega \in \Omega, \tag{25}$$

$$x_{bc}, y_c \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in C, \tag{26}$$

$$u_{bc}^{(\omega)} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in C, \forall \omega \in \Omega, \tag{27}$$

$$x_{b\acute{b}c}, z_{b\acute{b}c}^{(\omega)} \geq 0, \forall b, \acute{b} \in \mathcal{B}, \forall c \in C, \forall \omega \in \Omega. \tag{28}$$

This MILP formulation can be solved optimally using CPLEX.

D. COMPUTATIONAL COMPLEXITY

An MILP is, in general, NP-complete (i.e., there is no known algorithm that has a finishing time that is polynomial in the problem size)[6] [44]. However, MILPs can certainly be solved in a time that is exponential in the problem size using the branch and bound algorithms implemented in CPLEX.

The MILP formulation of C^{3}P^2 with sampled request rate distribution has $BC + C + BC|\Omega| = \mathcal{O}(BC|\Omega|)$ binary variables and $B^2C + B^2C|\Omega| = \mathcal{O}(B^2C|\Omega|)$ continuous variables. It also has $B+C+BC|\Omega|+BC+3B^2C+3B^2C|\Omega|+C|\Omega| = \mathcal{O}(B^2C|\Omega|)$ constraints.

A pure integer program (IP) is a special case of a mixed integer program (MIP). (An IP is an MIP with zero continuous variables.) Hence, an MIP is at least as hard as its corresponding IP. Therefore, the complexity of solving C^{3}P^2 can be expressed as $\Omega(2^{BC|\Omega|})$. This motivates us to develop, in the next subsection, a more efficient approach for solving C^{3}P^2.

E. SAMPLE AVERAGE APPROXIMATION (SAA) ALGORITHM FOR C^{3}P^2

In the previous subsection, we converted the *sampled* version of C^{3}P^2 into a form that can be solved optimally using the branch-and-bound and branch-and-cut algorithms implemented in CPLEX. In this subsection, we present an algorithm that provides lower and upper statistical bounds for the optimal objective function value of C^{3}P^2 (Problem 1). This algorithm, called the sample average approximation (SAA) algorithm, is summarized in Algorithm 1. The SAA algorithm consists of the following four main processes:

- *Scenario generation.* The uncertainty in C^{3}P^2 is in the RAN node request rates. The first process in the SAA algorithm is to generate the scenarios (realizations of the RAN node request rates) as described in Section III-B. Let Ω be the set of generated scenarios.
- *Solution of C^{3}P^2 with sampled request rate distribution.* The second process in the SAA algorithm is to solve the sampled version of C^{3}P^2, considering the scenarios that have been generated in the first process.
- *Verification of the solution feasibility.* Assume that $\bar{\mathbf{x}} \overset{\text{def}}{=} [\bar{x}_{bc}, \forall b \in \mathcal{B}, \forall c \in C]$ is an optimal solution for the sampled version of C^{3}P^2. In the third process of the SAA algorithm, we verify the feasibility of $\bar{\mathbf{x}}$ to C^{3}P^2. To do this, we first estimate the true probability function $q(\bar{\mathbf{x}})$ (defined in (29)) by $\hat{q}_{|\Omega|}(\bar{\mathbf{x}})$ (defined in (30)) using the set of scenarios Ω generated in the first process.

$$q(\bar{\mathbf{x}}) \overset{\text{def}}{=} \Pr\left\{2\ t_{bc}\ x_{bc} + \frac{1}{\mu - \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c}} > \delta\right\}, \tag{29}$$

$$\hat{q}_{|\Omega|}(\bar{\mathbf{x}}) \overset{\text{def}}{=} \frac{1}{|\Omega|}\sum_{\omega\in\Omega} \mathbb{1}_{\left\{2\ t_{bc}\ x_{bc} + \frac{1}{\mu - \sum_{\acute{b}\in\mathcal{B}} r_{\acute{b}}^{(\omega)}\ x_{\acute{b}c}} > \delta\right\}}. \tag{30}$$

Next, following the method described in [42] and [45], we construct a $(1-\xi)$-confidence upper bound on $q(\bar{\mathbf{x}})$, given by:

$$U(\bar{\mathbf{x}}) \overset{\text{def}}{=} \hat{q}_{|\acute{\Omega}|}(\bar{\mathbf{x}}) + z_\xi \sqrt{\frac{\hat{q}_{|\acute{\Omega}|}(\bar{\mathbf{x}})\left(1 - \hat{q}_{|\acute{\Omega}|}(\bar{\mathbf{x}})\right)}{|\acute{\Omega}|}}, \tag{31}$$

where $\acute{\Omega}$ is a set of new scenarios generated for the verification of the feasibility of $\bar{\mathbf{x}}$, $\hat{q}_{|\acute{\Omega}|}(\bar{\mathbf{x}})$ is the estimated value of $q(\bar{\mathbf{x}})$ for the set of scenarios $\acute{\Omega}$, $z_\xi \overset{\text{def}}{=} \Phi^{-1}(1-\xi)$, and $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution.[7] If $U(\bar{\mathbf{x}})$ is less than the risk level $(1-\beta)$, then $\bar{\mathbf{x}}$ is feasible with confidence level $(1-\xi)$.
- *Computation of the statistical lower and upper bounds.* If $\bar{\mathbf{x}}$ is a feasible solution to C^{3}P^2 (with confidence level $(1-\xi)$), then the corresponding objective function value constitutes a statistical upper bound on the optimal objective function value of C^{3}P^2. To get a lower bound for the optimal objective function value, we take I iterations. For each iteration, we run the sampled version of C^{3}P^2 with $|\Omega|$ scenarios J times.[8] For these J runs, we follow the same scheme as the one described in [42] to pick the lth smallest optimal value. Specifically, we first compute $\theta_{|\Omega|}$, as follows:

$$\begin{aligned}\theta_{|\Omega|} &\overset{\text{def}}{=} B(\lfloor(1-\alpha)\ |\Omega|\rfloor; 1-\beta, |\Omega|)\\ &\overset{\text{def}}{=} \sum_{i=0}^{\lfloor(1-\alpha)\ |\Omega|\rfloor}\binom{|\Omega|}{i}(1-\beta)^i\beta^{|\Omega|-i}.\end{aligned} \tag{32}$$

[6] Unless they have a special structure, such as the totally unimodular integer programs.

[7] $|\acute{\Omega}|$ is typically significantly larger than $|\Omega|$.

[8] We pick a different set of $|\Omega|$ scenarios for every iteration of $1 \leq i \leq I$ and $1 \leq j \leq J$.

Algorithm 1: SAA Algorithm for C^{3}P^2

```
Result: Lower and upper statistical bounds on the optimal objective function value.
$i \leftarrow 1$ $j \leftarrow 1$
while $i \leq I$ do
    while $j \leq J$ do
        a. Generate a new set $\Omega$ of scenarios.
        b. Using CPLEX, solve the sampled version of C$^3$P$^2$ with the new set $\Omega$ of scenarios. Denote the solution by $\bar{\mathbf{x}}_{ij}$ and the optimal value by $\bar{o}_{ij}$.
        c. Generate a new set $\acute{\Omega}$ of scenarios.
        d. Estimate $q(\bar{\mathbf{x}}_{ij})$ in (29) using the set $\acute{\Omega}$ of scenarios, as in (30). Denote the estimated value of $q(\bar{\mathbf{x}}_{ij})$ by $\hat{q}_{|\acute{\Omega}|}(\bar{\mathbf{x}}_{ij})$. Use (31) to compute $U(\bar{\mathbf{x}}_{ij})$.
        if $U(\bar{\mathbf{x}}_{ij}) \leq 1 - \alpha$ then
            Go to step e.
        else
            Continue to iteration $j + 1$.
        end
        e. Estimate the corresponding upper bound for C$^3$P$^2$ using (14).
    end
    Pick the smallest upper bound as the approximated upper bound and denote it by $a_i$.
    Sort the optimal solutions in non-decreasing order, pick the $l$th optimal value from (32)–(33), and denote it by $\bar{o}_{il_i}$.
end
$\frac{1}{I}\sum_{i=1}^{I} \bar{o}_{il_i}$ is a lower statistical bound for the optimal objective function value.
$\min_{1 \leq i \leq I} a_i$ is an upper statistical bound for the optimal objective function value.
The optimality gap is estimated to be $\frac{\min_{1 \leq i \leq I} a_i - \frac{1}{I}\sum_{i=1}^{I} \bar{o}_{il_i}}{\frac{1}{I}\sum_{i=1}^{I} \bar{o}_{il_i}} \times 100\%$.
```

where $B(\cdot;\cdot,\cdot)$ is the CDF of the binomial distribution. Then, l is computed as the largest integer such that:

$$B\big(l-1;\theta_{|\Omega|},J\big) \overset{\text{def}}{=} \sum_{i=0}^{l-1} \binom{J}{i} {\theta_{|\Omega|}}^{i} \big(1-\theta_{|\Omega|}\big)^{J-i} \leq \xi. \tag{33}$$

Finally, taking the average of the lth smallest optimal values across the I iterations provides a lower bound for the optimal objective function value.

As explained in [42], if the optimal value of the sampled version of C^{3}P^2 is denoted by $\bar{o}$, then $\bar{o}_{ij}, j \in \{1, 2, \ldots, J\}$ for a given i (in Algorithm 1), can be viewed as an i.i.d. sample of the random variable $\bar{o}$. If $\bar{o}_{ij}$ are sorted in a non-decreasing order, i.e., $\bar{o}_{i1} \leq \ldots \leq \bar{o}_{iJ}$. Then, it can be shown that with probability at least $1 - \xi$, the random quantity $\bar{o}_{il_i}$ is a lower bound of the optimal objective function value of C^{3}P^2 [43], [46], [47].

V. ROBUST JOINT RIC PLACEMENT AND ADAPTIVE RAN NODE-RIC ASSIGNMENT (CPPA)

In this section, we consider the joint stochastic RIC placement and *adaptive* RAN node-RIC assignment problem, referred to as CPPA.

A. PROBLEM FORMULATION

Using two-stage stochastic programming [22], we formulate CPPA under the uncertainty of $\tilde{\mathbf{r}} \overset{\text{def}}{=} [\tilde{r}_b, \forall b \in \mathcal{B}]$. In contrast to C^{3}P^2, in CPPA the RAN node-RIC assignment adapts to the variations in the RAN node request rates.

The goal of the first-stage problem is to optimally place the minimum number of RICs, knowing the distribution of $\tilde{\mathbf{r}}$. Our optimality criteria are: (i) minimizing the number of RICs and (ii) minimizing the response time to various RAN nodes, without decreasing it below δ. In contrast to C^{3}P^2, CPPA does not ensure that the RAN node response time constraints are satisfied with a minimum probability of β. The first-stage problem decision is static and is taken before knowing which realization of $\tilde{\mathbf{r}}$ will occur. In the second-stage problem, the RAN node-RIC assignment is optimized under each realization of $\tilde{\mathbf{r}}$ aiming at minimizing the response time to various RAN nodes, without decreasing it below δ. Our two-stage stochastic optimization problem can be formulated as follows:

Problem 2 (CPPA):

$$\underset{\{y_c, c \in C\}}{\text{minimize}} \left\{ \sum_{c \in C} y_c + \mathbb{E}\big[h(\mathbf{x}, \tilde{\mathbf{r}})\big] \right\} \tag{34}$$

subject to:

$$y_c \in \{0, 1\}, \forall c \in C, \tag{35}$$

where $h(\mathbf{x}, \tilde{\mathbf{r}})$ is the optimal value of the second-stage problem, which is given by:

$$\underset{\left\{\substack{x_{bc},\\ b \in \mathcal{B}, c \in C}\right\}}{\text{minimize}} \left\{ \sum_{b \in \mathcal{B}} \sum_{c \in C} q_{bc} \times \max\left(2\, t_{bc}\, x_{bc} + \frac{1}{\mu - \sum_{\acute{b} \in \mathcal{B}} \tilde{r}_{\acute{b}}\, x_{\acute{b}c}}, \delta \right) \right\} \tag{36}$$

subject to:

$$y_c = \mathbb{1}_{\{\sum_{b \in \mathcal{B}} x_{bc} \geq 1\}}, \forall c \in C, \tag{37}$$

$$\sum_{c \in C} x_{bc} = 1, \forall b \in \mathcal{B}, \tag{38}$$

$$\Pr\left\{ \sum_{b \in \mathcal{B}} \tilde{r}_b\, x_{bc} \leq \mu \right\} = 1, \forall c \in C, \tag{39}$$

$$x_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in C, \tag{40}$$

where $q_{bc}, b \in \mathcal{B}, c \in \mathcal{C}$, are design coefficients introduced to balance the tradeoff between minimizing the number of RICs and minimizing the response time to the RAN nodes.

B. CPPA WITH SAMPLED REQUEST RATE DISTRIBUTION

A key source of difficulty in solving two-stage stochastic programs is in evaluating $\mathbb{E}[h(\mathbf{x}, \tilde{\mathbf{r}})]$. One standard technique for addressing this difficulty is sampling. The basic idea, as described in Section III-B, is to approximate the true distribution of stochastic variables with an empirical distribution by sampling. We generate a set Ω of i.i.d. samples (scenarios) from the distribution of the RAN node request rates using Monte Carlo simulation. After generating the scenarios, $\mathbb{E}[h(\mathbf{x}, \tilde{\mathbf{r}})]$ can be estimated as $\frac{1}{|\Omega|}\sum_{\omega\in\Omega} h(\mathbf{x}, \mathbf{r}^{(\omega)})$, where $\mathbf{r}^{(\omega)} \overset{\text{def}}{=} [r_b^{(\omega)}, \forall b \in \mathcal{B}]$ is the vector of the RAN node request rates under scenario ω. CPPA with sampled request rate distribution is given by:

CPPA With Sampled Request Rate Distribution:

$$\underset{\left\{\substack{y_c, x_{bc}^{(\omega)},\\ b\in\mathcal{B}, c\in\mathcal{C},\\ \omega\in\Omega}\right\}}{\text{minimize}} \left\{\sum_{c\in\mathcal{C}} y_c + \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left(\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}} q_{bc} \times \max \right.\right.$$
$$\left.\left. \times \left(2\, t_{bc}\, x_{bc}^{(\omega)} + \frac{1}{\mu - \sum_{\acute{b}\in\mathcal{B}} r_{\acute{b}}^{(\omega)}\, x_{\acute{b}c}^{(\omega)}}, \delta\right)\right)\right\} \tag{41}$$

subject to:

$$y_c = \mathbb{1}_{\left\{\sum_{\omega\in\Omega}\sum_{b\in\mathcal{B}} x_{bc}^{(\omega)} \geq 1\right\}}, \forall c \in \mathcal{C}, \tag{42}$$

$$\sum_{c\in\mathcal{C}} x_{bc}^{(\omega)} = 1, \forall b \in \mathcal{B}, \forall \omega \in \Omega, \tag{43}$$

$$\sum_{b\in\mathcal{B}} r_b^{(\omega)}\, x_{bc} \leq \mu, \forall c \in \mathcal{C}, \forall \omega \in \Omega, \tag{44}$$

$$x_{bc}^{(\omega)} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega, \tag{45}$$

$$y_c \in \{0, 1\}, \forall c \in \mathcal{C}. \tag{46}$$

It has been shown that a solution to the CPPA with sampled request rate distribution is an optimal solution to the CPPA with probability approaching one exponentially fast as $|\Omega|$ increases [48], [49]. Specifically, if $\mathbf{y} \overset{\text{def}}{=} [y_c, \forall c \in \mathcal{C}]$, and $\mathbf{y}_\epsilon$ and $\hat{\mathbf{y}}_\epsilon$ denote the sets of ϵ-optimal solutions of CPPA and its sampled version, respectively. Then, for any $\epsilon > 0$ and $\delta \in [0, \epsilon]$, there exists a constant $\zeta(\delta, \epsilon) \geq 0$ such that $\Pr\{\hat{\mathbf{y}}_\delta \subset \mathbf{y}_\epsilon\} \geq 1 - 2^{C} e^{-\zeta(\delta,\epsilon)|\Omega|}$.

In the next subsection, we use several linearization techniques to convert the sampled version of CPPA into a mixed-integer linear program (MILP), in order to solve it using CPLEX.

C. MIXED-INTEGER LINEAR REFORMULATION

The mathematical formulation of CPPA developed in the previous subsections is non-linear. In this subsection, we present an *equivalent* linear reformulation of CPPA. The mathematical details of deriving this linear reformulation of CPPA are explained in Appendix C.

Algorithm 2: SAA Algorithm for CPPA

```
Result: Lower and upper statistical bounds on the
        optimal objective function value.
$j \leftarrow 1$
while $j \leq J$ do
    a. Generate a new set $\Omega$ of scenarios.
    b. Solve the sampled version of CPPA with the new
       set $\Omega$ of scenarios. Denote the solution by $\bar{\mathbf{x}}_j$ and
       the optimal value by $\bar{o}_j$.
    c. Generate a new set $\acute{\Omega}$ of scenarios.
    d. Estimate the corresponding upper bound for CPPA
       based on the set $\acute{\Omega}$ of scenarios, which is given by
       $a_j \overset{\text{def}}{=} \sum_{c\in\mathcal{C}} y_c + \frac{1}{|\acute{\Omega}|}\sum_{\omega\in\acute{\Omega}} h\big(\bar{\mathbf{x}}, \mathbf{r}^{(\omega)}\big)$.
end
$\frac{1}{J}\sum_{j=1}^{J} \bar{o}_j$ is a lower statistical bound for the optimal
objective function value.
$\min_{1\leq j\leq J} a_j$ is an upper statistical bound for the optimal
objective function value.
The optimality gap is estimated to be
$\frac{\min_{1\leq j\leq J} a_j - \frac{1}{J}\sum_{j=1}^{J}\bar{o}_j}{\frac{1}{J}\sum_{j=1}^{J}\bar{o}_j} \times 100\%$.
```

The sampled version of CPPA can be equivalently written as an MILP as follows:

MILP Reformulation of the Sampled Version of CPPA:

$$\underset{\left\{\substack{y_c, x_{bc}^{(\omega)},\\ v_{bc}^{(\omega)}, f_{bc}^{(\omega)},\\ d_{bc}^{(\omega)}, x_{b\acute{b}c}^{(\omega)},\\ f_{b\acute{b}c}^{(\omega)}, e_{b\acute{b}c}^{(\omega)},\\ b,\acute{b}\in\mathcal{B},\\ c\in\mathcal{C},\omega\in\Omega}\right\}}{\text{minimize}} \left\{\sum_{c\in\mathcal{C}} y_c + \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left(\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}} q_{bc}\, v_{bc}^{(\omega)}\right)\right\} \tag{47}$$

subject to:

(15), (16), (25), (63)−(69), $\forall\omega \in \Omega$

linearization constraints of $x_{b\acute{b}c}^{(\omega)}$ and $f_{b\acute{b}c}^{(\omega)}$, similar to (61) and (62).

This MILP formulation can be solved optimally using CPLEX.

D. COMPUTATIONAL COMPLEXITY

The MILP formulation of CPPA with sampled request rate distribution has $C + 3BC|\Omega| + 2B^2C|\Omega| = \mathcal{O}(B^2C|\Omega|)$ binary variables and $BC|\Omega| + B^2C|\Omega| = \mathcal{O}(B^2C|\Omega|)$ continuous variables. It also has $4B^2C|\Omega| + 6BC|\Omega| + 2C|\Omega| + B|\Omega| = \mathcal{O}(B^2C|\Omega|)$ constraints. Hence, the complexity of solving CPPA can be expressed as $\Omega(2^{B^2C|\Omega|})$. This motivates us to develop, in the next subsection, a more efficient approach for solving CPPA.

E. SAMPLE AVERAGE APPROXIMATION (SAA) ALGORITHM FOR CPPA

Similar to C^{3}P^2, in this subsection we present an SAA algorithm that provides lower and upper statistical bounds for the optimal objective function value of CPPA. This algorithm, summarized in Algorithm 2, consists of three main processes: (i) Scenario generation, (ii) solution of CPPA with sampled request rate distribution, and (iii) computation of the statistical lower and upper bounds. Processes (i) and (ii) are similar to those of the SAA algorithm of C^{3}P^2, described in Section IV-E. If $(\bar{\mathbf{y}} \overset{\text{def}}{=} [\bar{y}_c, \forall c \in C], \bar{\mathbf{x}})$ is a feasible solution to CPPA, then $\sum_{c \in C} \bar{y}_c + \frac{1}{|\acute{\Omega}|} \sum_{\omega \in \acute{\Omega}} h(\bar{\mathbf{x}}, \mathbf{r}^{(\omega)})$ constitutes a statistical upper bound on the optimal objective function value of CPPA, where $\acute{\Omega}$ is a new set of scenarios that is different from the set generated in the first process of the SAA algorithm. To get a lower bound for the optimal objective function value of CPPA, we take J iterations. For each iteration, we run the sampled version of CPPA with $|\Omega|$ scenarios.[9] Taking the average of the optimal values across the J iterations provides a lower bound for the optimal objective function value.

VI. PERFORMANCE EVALUATION

In this section, we evaluate our stochastic joint RIC placement and RAN node-RIC assignment schemes and compare them with sequential optimization and deterministic optimization schemes.

A. EVALUATION SETUP

We used the grid topology shown in Figure 3 with $B = 9$ and $C = 4$. The RIC processing capacity (μ) was set to 20000 requests/second. k in Section III was set to 0.2. The packet size of the flow setup request was set to 1500 Bytes [50] and the channel data rate was set to 100 Mbps (hence, the transmission time equals $\frac{1500 \times 8}{100 \times 10^6} = 0.12$ milliseconds). α in (11) equals β in (6). We ran our experiments on an Intel core i5 3.3 GHz core duo with 64 GB RAM.

In addition to the number of RICs, we introduce two performance evaluation metrics to assess the ability of the RIC deployment and assignment schemes to provide on-time responses to the RAN node requests. These metrics are:

- *The average probability of RAN node satisfaction*: In general, the response time to different RAN nodes will be $\leq \delta$ with different probabilities. In this metric, we compute the average of these probabilities.
- *The average RAN node delay dissatisfaction*: If the response time to a RAN node is $\leq \delta$ with probability κ, this means that the response time to this RAN node exceeds δ for $(1-\kappa) \times 100\%$ of the scenarios. The excess delay under these scenarios varies from one scenario to another. In this metric, we compute the average excess delay, averaged over the scenarios where the response time exceeds δ and over the different RAN nodes.

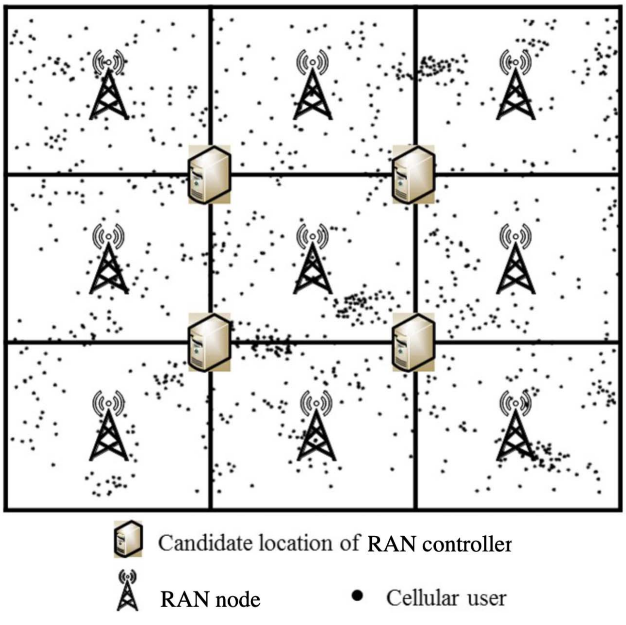


FIGURE 3. Considered network topology in our performance evaluation. The locations of the mobile users form a realization of a Poisson point process.

To explain how the average probability of RAN node satisfaction and the average RAN node delay dissatisfaction are computed, let $I_{b\omega}$ and $d_{b\omega}$ be two variables defined for each $b \in \mathcal{B}$ and $\omega \in \Omega$. Then, the average probability of RAN node satisfaction and the average RAN node delay dissatisfaction can be computed following Procedure .

We studied the effects of δ, β, and $q_{bc} \overset{\text{def}}{=} q, \forall b \in \mathcal{B}, \forall c \in C$, on the above performance metrics.

B. DETERMINISTIC PLACEMENT AND ASSIGNMENT: SEQUENTIAL VS. JOINT

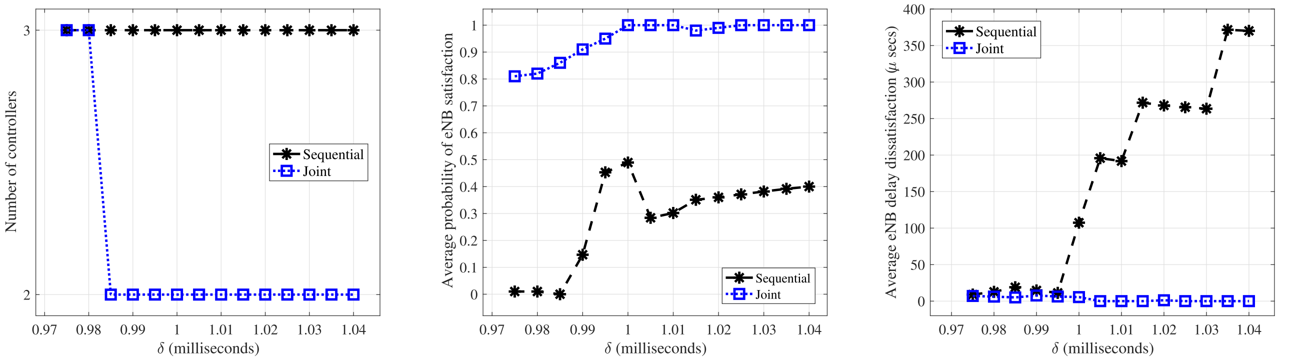
In this subsection, we demonstrate the advantages of jointly optimizing the RIC placement and the RAN node-RIC assignment problems, as compared to solving these problems sequentially. Specifically, we compare a modified version of C^{3}P^2, after replacing the stochastic per-link delay constraint (6) with the following deterministic average delay constraint:

$$\frac{\sum_{b \in \mathcal{B}} 2\, t_{bc}\, x_{bc}}{\sum_{b \in \mathcal{B}} x_{bc}} + \frac{1}{\mu - \sum_{b \in \mathcal{B}} \mathbb{E}[\tilde{r}_b]\, x_{bc}} \leq \delta, \tag{48}$$

with a sequential scheme, similar to the one proposed in [12], in which the controller placement and the switch-controller assignment problems were solved separately.

In our sequential scheme, the set of RAN nodes $\mathcal{B}$ is divided into C subsets, denoted by $\mathcal{S}_c, c \in C$, that are not necessarily mutually exclusive. The subset $\mathcal{S}_c$ includes the RAN nodes that are in the immediate neighborhood of the RIC candidate location $c \in C$. (A RAN node can be in the immediate neighborhood of multiple RIC candidate locations and hence exist in multiple subsets.) If a RIC is to be deployed at location c, then the RAN nodes that are

[9] Every time, we pick a different set of $|\Omega|$ scenarios.



(a) Number of RICs (b) Average probability of RAN node satisfaction (c) Average RAN node delay dissatisfaction

FIGURE 4. Comparison between sequential [12] and joint RIC placement and RAN node-RIC assignment.

```
Procedure 1: Performance Metrics Evaluation
  Input: RIC placement and assignment,
  $t_{bc}, \forall b \in \mathcal{B}, c \in \mathcal{C}$, $\mu$, $r_b^{(\omega)}, \forall b \in \mathcal{B}, \omega \in \Omega$, and $\delta$
  Output: Average probability of RAN node satisfaction
  and average RAN node delay dissatisfaction for each
  RAN node $b \in \mathcal{B}$ do
    for each scenario $\omega \in \Omega$ do
      Evaluate the response time to RAN node $b$
      under scenario $\omega$, following the RIC placement
      and assignment provided by each scheme.
      if response time $\leq \delta$ then
        $I_{b\omega} = 1$
        $d_{b\omega} = 0$
      else
        $I_{b\omega} = 0$
        $d_{b\omega} =$ response time $- \delta$
      end
    end
    Compute the probability of delay satisfaction for
    RAN node $b$ as $\frac{1}{|\Omega|} \sum_{\omega \in \Omega} I_{b\omega}$.
    Compute the average delay dissatisfaction for RAN
    node $b$ as $\frac{1}{|\Omega|} \sum_{\omega \in \Omega} d_{b\omega}$.
  end
  Compute the average probability of RAN node
  satisfaction as $\frac{1}{B} \sum_{b \in \mathcal{B}} \frac{1}{|\Omega|} \sum_{\omega \in \Omega} I_{b\omega}$.
  Compute the average RAN node delay dissatisfaction
  as $\frac{1}{B} \sum_{b \in \mathcal{B}} \frac{1}{|\Omega|} \sum_{\omega \in \Omega} d_{b\omega}$.
```

in $\mathcal{S}_c$ and do not exist in any other subset will be assigned to it. A RAN node that belongs to multiple subsets will be randomly assigned to one of the corresponding RIC locations (if a RIC is to be deployed at more than one of these locations).

After precomputing $\mathcal{S}_c, \forall c \in \mathcal{C}$, the RIC placement is optimized taking $\mathcal{S}_c, c \in \mathcal{C}$ as input data by finding the minimal sub-collection of $\mathcal{S}_c, c \in \mathcal{C}$ that serves all RAN nodes. For each RAN node, it is required that at least one of the subsets containing it is selected.

As shown in Figure 4, the joint scheme (i) reduces the number of RICs from three to two for most values of δ, (ii) increases the average probability of RAN node satisfaction by at least 50% (when $\delta = 1$ millisecond), and (iii) brings the level of average RAN node delay dissatisfaction close to 0.

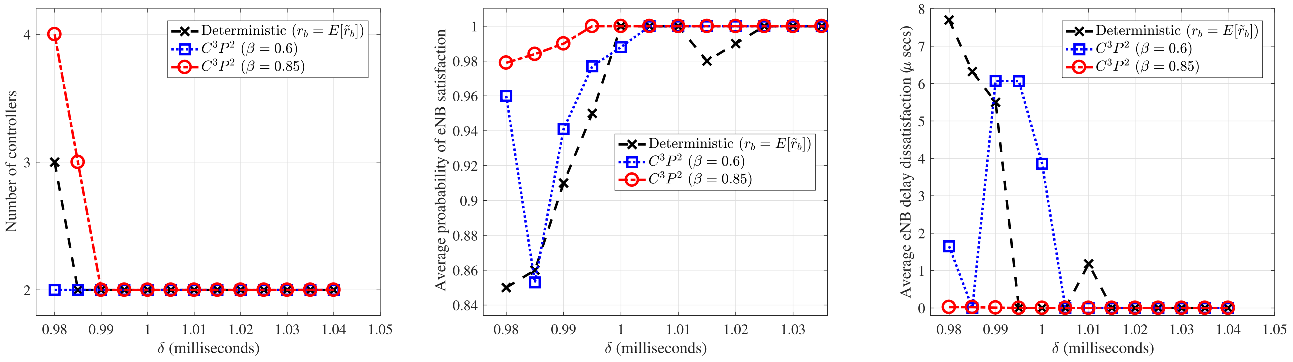
C. DETERMINISTIC VS. STOCHASTIC PLACEMENT AND ASSIGNMENT

In this subsection, we illustrate the gains of stochastic optimization as compared to deterministic optimization. Specifically, we compare C^{3}P^2 with a deterministic version of it, when we replace $\tilde{r}_b$ with $\mathbb{E}[\tilde{r}_b]$ and remove the probability term in (6). We considered 100 i.i.d. scenarios, i.e., realizations of mobile user locations (and hence, RAN node request rates), each containing 1000 users. Each scenario was generated as a non-stationary PPP from the SSLT field (as described in Section III-B) with $\omega_{\max} = \pi/30$, $\sigma = 1$, $\gamma = 0$, and $L = 25$. The field is valid over the domain $x, y \in [0, 750]$ meters. $\mathbb{E}[\tilde{r}_b] = [2629.4, 3957.8, 3360.8, 2824.8, 4544, 2591.6, 2806.8, 2635.4, 3039.8]$. $\Pr\{\tilde{r}_b < \mathbb{E}[\tilde{r}_b]\} = [0.51, 0.53, 0.48, 0.55, 0.52, 0.51, 0.55, 0.51, 0.45]$.

Since $0.45 \leq \Pr\{\tilde{r}_b < \mathbb{E}[\tilde{r}_b]\} \leq 0.55$, Figure 5 shows a comparable performance of the deterministic scheme to C^{3}P^2 with $\beta = 0.6$. However, when β increases to 0.85, C^{3}P^2 improves the average probability of RAN node satisfaction significantly and brings the average RAN node delay dissatisfaction level close to 0.

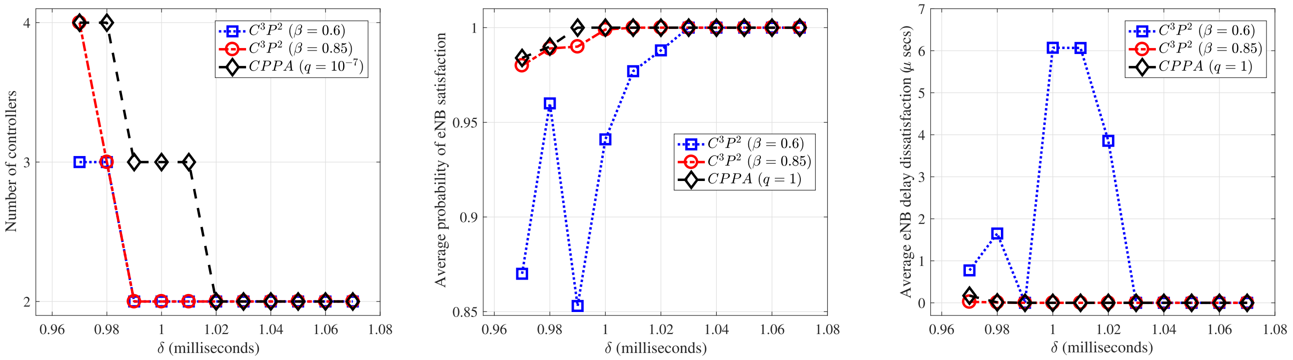
D. STOCHASTIC PLACEMENT AND ASSIGNMENT: STATIC VS. ADAPTIVE

In this subsection, we compare our static single-stage scheme (C^{3}P^2) with the adaptive two-stage scheme (CPPA) using the same 100 scenarios used in Section VI-C. As can be seen from (47), $q \overset{\text{def}}{=} q_{bc}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}$, controls the tradeoff between the number of RICs needed and the RAN node delay satisfaction. The objective function of CPPA (47) is a weighted sum of the number of RICs and the expected delay, where the weight of the number of RICs is one and the weight of the expected delay is q.



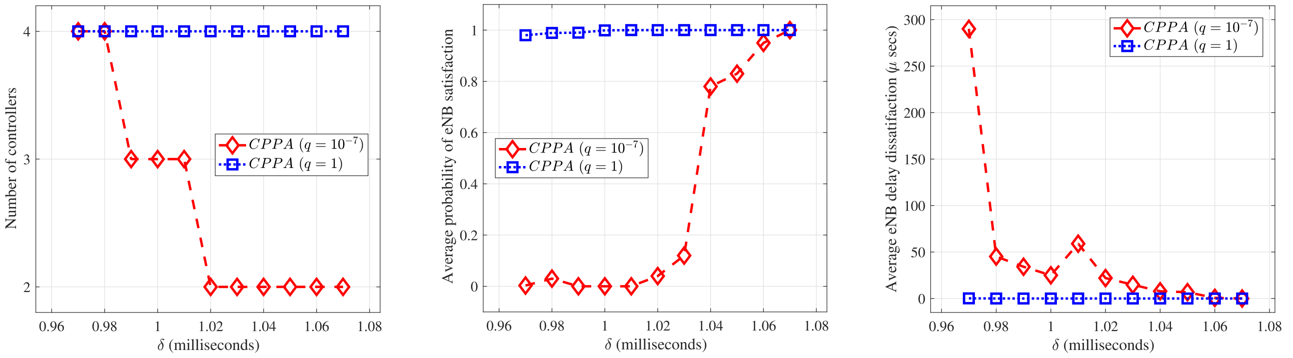
(a) Number of RICs (b) Average probability of RAN node satisfaction (c) Average RAN node delay dissatisfaction

FIGURE 5. Comparison between deterministic and static stochastic RIC placement and RAN node-RIC assignment.



(a) Number of RICs (b) Average probability of RAN node satisfaction (c) Average RAN node delay dissatisfaction

FIGURE 6. Comparison between static (C³P²) and adaptive (CPPA) joint RIC placement and RAN node-RIC assignment.



(a) Number of RICs (b) Average probability of RAN node satisfaction (c) Average RAN node delay dissatisfaction

FIGURE 7. Effect of $q_{bc} \overset{\text{def}}{=} q, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}$, on the joint RIC placement and adaptive RAN node-RIC assignment (CPPA).

We consider two use cases. In the first use case, q is set to a value that makes the expected delay dominated by the number of RICs needed (i.e., it is more important for the optimizer to minimize the number of RICs than minimizing the expected delay). Considering the range of values that the expected *total* delay can take in the given setup, setting q to 10^{-7} makes the second term in the objective function in (47), which is $q \times$ expected delay, less than 0.1.

In the second use case, q is set to a value that makes the number of RICs dominated by the expected delay (i.e., it is more important for the optimizer to minimize the expected delay than minimizing the number of RICs). Setting q to 1 makes the second term in the objective function in (47), which is $q \times$ expected delay, several orders of magnitude higher than the first term (i.e., the number of RICs).

As shown in Figures 6(a) and 7(a), in the first use case two RICs are enough for most values of δ, which is significantly less than the number of RICs needed in the second use case, and comparable to the number of RICs of C³P².

Recall that in contrast to C³P², CPPA does not provide guarantees on RAN node delay satisfaction. Even though,

TABLE 3. Results of C^3P^2 for different values of $|\acute{\Omega}|$ when $I = 5$, $J = 5$, and $|\Omega| = 50$.

$\lvert\acute{\Omega}\rvert$	SAA			$\lvert\Omega\rvert$	Sampled C^3P^2
	Lower bound	Upper bound	Optimality gap (%)		
150	3	3	0	1250	3
200	3	3	0		
250	3	3	0		

TABLE 4. Results of C^3P^2 for different values of $|\acute{\Omega}|$ when $I = 7$, $J = 5$, and $|\Omega| = 50$.

$\lvert\acute{\Omega}\rvert$	SAA			$\lvert\Omega\rvert$	Sampled C^3P^2
	Lower bound	Upper bound	Optimality gap (%)		
150	3	3	0	1750	3
200	3	3	0		

Figure 6 shows that when $q = 1$ CCPA has a superior performance in RAN node satisfaction compared to C^3P^2 when $\beta = 0.6$, and a comparable performance to C^3P^2 when $\beta = 0.85$. Finally, Figure 7 demonstrates the effect of q on controlling the tradeoff between the number of RICs and the RAN node delay satisfaction.

E. SAMPLE AVERAGE APPROXIMATION (SAA)

In this subsection, we evaluate the SAA algorithms of C^3P^2 and CPPA.

1) C^3P^2

Considering the setup explained in Section VI-A, in Tables 3 and 4 we compare the results obtained from running the C^3P^2 with sampled request rate distribution ((14)–(28)) with the results obtained from running the SAA algorithm of C^3P^2 (Algorithm 1). We considered different values of I, J, $|\Omega|$, and $|\acute{\Omega}|$.

Two use cases are shown in Tables 3 and 4. In the first use case, I, J, and $|\Omega|$ in Algorithm 1 were set to 5, 5, and 50, respectively. Three different values of $|\acute{\Omega}|$ were examined in this use case. The results of the SAA algorithm of C^3P^2 with these parameter values are compared with the results of the C^3P^2 (with sampled request rate distribution) with $|\Omega| = 5 \times 5 \times 50 = 1250$ scenarios and shown in Table 3. Table 3 shows that, for the three different values of $|\acute{\Omega}|$, the SAA algorithm was able to achieve the same optimal solutions as the C^3P^2 (optimality gap is 0%). This means that instead of running one MILP for 1250 scenarios, using the SAA algorithm, we can run in parallel $5 \times 5 = 25$ independent MILPs, each for 50 scenarios, and obtain the optimal solution.

In the second use case, I, J, and $|\Omega|$ in Algorithm 1 were set to 7, 5, and 50, respectively. Two different values of $|\acute{\Omega}|$ were examined in this use case. The results of the SAA algorithm of C^3P^2 with these parameter values are compared with the results of the C^3P^2 with $|\Omega| = 7 \times 5 \times 50 = 1750$ scenarios and shown in Table 4. Again, Table 4 shows that, for the two different values of $|\acute{\Omega}|$, the SAA algorithm was able to achieve the same optimal solutions as the C^3P^2 (optimality gap is 0%). This means that instead of running one MILP for 1750 scenarios, using the SAA algorithm, we can run in parallel $7 \times 5 = 35$ independent MILPs, each for 50 scenarios, and obtain the optimal solution.

TABLE 5. Results of the SAA algorithm of CPPA for different values of $|\Omega|$ when $J = 8$ and $|\acute{\Omega}| = 1100$.

Number of scenarios ($\lvert\Omega\rvert$)	Lower bound	Upper bound	Optimality gap (%)
60	2.23247	2.24096	0.380059
80	2.23834	2.24096	0.117186
100	2.2395	2.24096	0.065016

TABLE 6. Results of the SAA algorithm of CPPA for different values of $|\acute{\Omega}|$ and J when $|\Omega| = 80$.

$\left(\lvert\acute{\Omega}\rvert, J\right)$	Lower bound	Upper bound	Optimality gap (%)
(500, 6)	2.24369	2.23755	0.273639
(700, 6)	2.24369	2.24055	0.14018
(700, 8)	2.23834	2.24055	0.0987944
(1100, 8)	2.23834	2.24096	0.117186

2) CPPA

Tables 3 and 4 motivate us to use the SAA algorithm of CPPA to solve the CPPA for a much larger set of scenarios, compared to the set of scenarios that can be considered when solving the CPPA with sampled request rate distribution. Using the SAA algorithm, we obtained estimates of lower and upper bounds on the optimal objective function value, considering up to 10,000 scenarios (compared to the 100 scenarios considered when solving the CPPA with sampled request rate distribution).

Two experiments were conducted. In the first experiment, J and $|\acute{\Omega}|$ were set to 8 and 1100, respectively and three different values of $|\Omega|$ were considered. The lower and upper statistical bounds along with the optimality gaps are shown in Table 5. As shown in Table 5, increasing the number of considered scenarios ($|\Omega|$) reduces the optimality gap.

In the second experiment, $|\Omega|$ was set to 80 and four different combinations of $(|\acute{\Omega}|, J)$ were considered. The lower and upper statistical bounds along with the optimality gaps are shown in Table 6. As shown in Table 6, increasing the number of scenarios used in the verification process ($|\acute{\Omega}|$) or J reduces the optimality gap. Tables 5 and 6 show the ability of the SAA algorithm to estimate the optimal objective function value with a very low optimality gap. This is further illustrated using the circular dendrograms shown in Figures 8 and 9.

VII. CONCLUSION

Using stochastic programming, in this paper we studied the controller placement problem in software-defined cellular

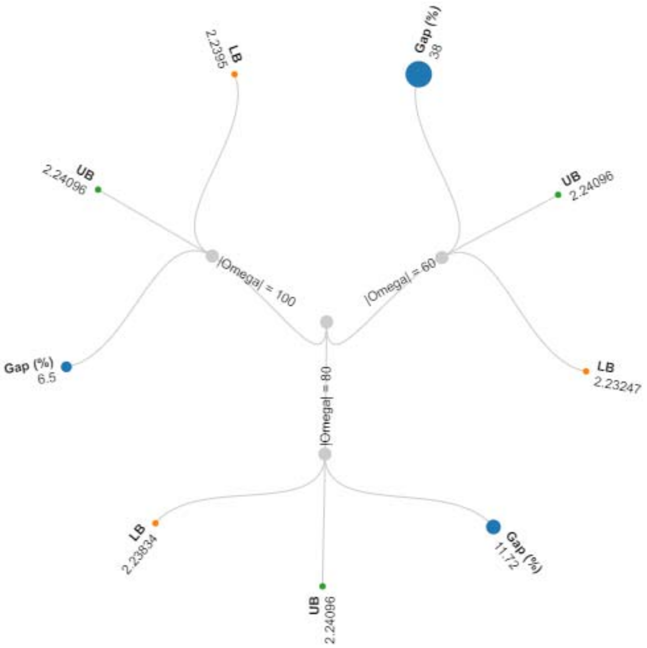


FIGURE 8. Lower bound (LB), upper bound (UB), and optimality gap (Gap) of SAA-based CPPA for different values of $|\Omega|$ when $J = 8$ and $|\hat{\Omega}| = 1100$.

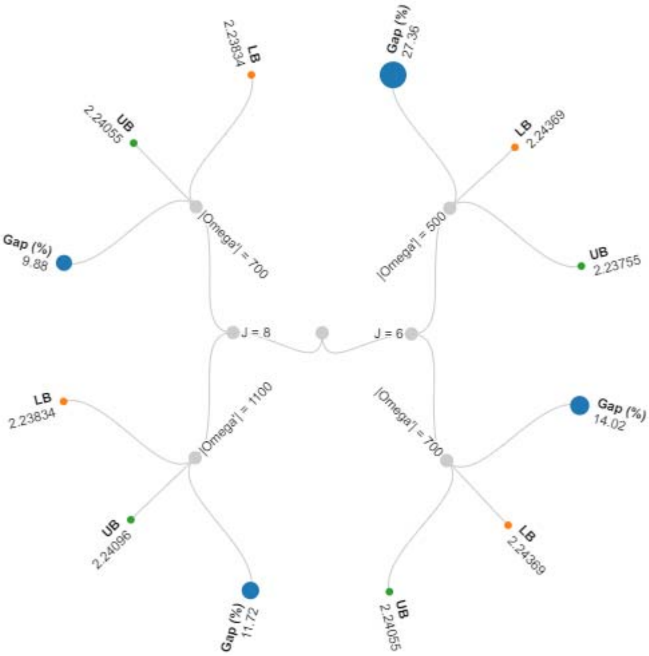


FIGURE 9. Lower bound (LB), upper bound (UB), and optimality gap (Gap) of SAA-based CPPA for different values of $|\hat{\Omega}|$ and J when $|\Omega| = 80$.

networks, considering the uncertainty in the mobile user locations. We employed a generic theoretical approach that can be applied to different SDCN technologies, but we also showed the applicability of our proposal to the under-development O-RANs. We developed a static (C^3P^2) and an adaptive (CPPA) joint stochastic controller placement and RAN node-controller assignment problems. Our optimization criteria are: (i) minimizing the number of controllers and (ii) minimizing the response time to various RAN nodes. In contrast to C^3P^2, in CPPA the RAN node-controller assignment adapts to the variations in the mobile user locations. However, CPPA does not ensure that the RAN node response time constraints are satisfied with a minimum probability of β, whereas C^3P^2 ensures that. Using stochastic optimization and sample average approximation (SAA), combined with various linearization techniques, we extensively evaluated C^3P^2 and CPPA. Our results demonstrated the advantages of (i) joint compared to sequential optimization, (ii) stochastic compared to deterministic optimization, and (iii) adaptive compared to static optimization. They also illustrated the ability of the proposed SAA framework in solving C^3P^2 and CPPA efficiently (with much lower time complexity compared to solving the full deterministic equivalent mixed-integer linear programs) and estimating their optimal objective function values with very low optimality gaps.

TABLE 7. List of abbreviations.

Abbreviation	Description
LTE	Long-term evolution.
eNB	Evolved node B.
gNB	Next-generation node B.
WAN	Wide area network.
RAN	Radio access network.
O-RAN	Open radio access network.
RIC	RAN intelligent controller.
Near-RT RIC	Near-real-time RIC.
Non-RT RIC	Non-real-time RIC.
CU	Central unit.
DU	Distributed unit.
xApps	Near-RT RIC applications.
rApps	Non-RT RIC applications.
NFV	Network function virtualization.
SDN	Software-defined networking.
SDCN	Software-defined cellular network.
SDVN	Software-defined vehicular network.
CPP	Controller placement problem.
C^3P^2	Chance-constrained scheme for CPP.
CPPA	Adaptive scheme for CPP.
MILP	Mixed-integer linear program.
SAA	Sample average approximation.
SSLT	Scalable, spatially-correlated, and log-normally distributed traffic.
PPP	Poisson point process.
i.i.d	Independent and identically distributed.
CDF	Cumulative distribution function.

APPENDIX A ABBREVIATIONS

The list of abbreviations used in the paper are listed in Table 7.

APPENDIX B
MILP REFORMULATION OF C^{3}P^2

The objective function of C^{3}P^2 (9) and the response time constraint (11) have indicator functions, which are non-linear. In the following, we explain the linearization methodology for each of these indicator functions [51].

A.LINEARIZING THE OBJECTIVE FUNCTION

To linearize the indicator function in (9), we introduce new *auxiliary* binary decision variables, $y_c, \forall c \in \mathcal{C}$, which are defined in terms of the *true* decision variables $x_{bc}, b \in \mathcal{B}$, $c \in \mathcal{C}$, as follows:

$$y_c \overset{\text{def}}{=} \mathbb{1}_{\{\sum_{b \in \mathcal{B}} x_{bc} \geq 1\}}, \forall c \in \mathcal{C}. \tag{49}$$

The objective function (9) is then rewritten as:

$$\underset{\{x_{bc}, b \in \mathcal{B}, c \in \mathcal{C}\}}{\text{minimize}} \sum_{c \in \mathcal{C}} y_c \tag{50}$$

subject to:

$$y_c \overset{\text{def}}{=} \mathbb{1}_{\{\sum_{b \in \mathcal{B}} x_{bc} \geq 1\}}, \forall c \in \mathcal{C}. \tag{51}$$

The relationship between y_c and x_{bc}, given by (51), needs then to be linearly reformulated so that the indicator function is omitted. Equation (51) says that $y_c = 1$ if and only if $\sum_{b \in \mathcal{B}} x_{bc} \geq 1$, which means the following:

- If $\sum_{b \in \mathcal{B}} x_{bc} \geq 1$ then $y_c = 1$.
- If $y_c = 1$ then $\sum_{b \in \mathcal{B}} x_{bc} \geq 1$.

Each of the above if-then statements needs to be linearly expressed.

Let M be an upper bound of $\sum_{b \in \mathcal{B}} x_{bc} - 1$. Then, the first if-then statement above can be reformulated as follows:

$$\sum_{b \in \mathcal{B}} x_{bc} - (M + \epsilon) y_c \leq 1 - \epsilon, \tag{52}$$

where $\epsilon > 0$ is a small tolerance beyond which we regard the constraint as having been broken. Selecting M and ϵ to be $B - 1$ and 1, respectively, (1) reduces to:

$$\sum_{b \in \mathcal{B}} x_{bc} \leq B y_c. \tag{53}$$

Equation (53) expresses the if-then statement "if $\sum_{b \in \mathcal{B}} x_{bc} \geq 1$ then $y_c = 1$" linearly. If $\sum_{b \in \mathcal{B}} x_{bc} \geq 1$, then, according to (53), y_c cannot be assigned to 0 and it has to be 1.

Next, we reformulate the second if-then statement above. Let m be a lower bound of $\sum_{b \in \mathcal{B}} x_{bc} - 1$. Then, the second if-then statement can be reformulated as follows:

$$\sum_{b \in \mathcal{B}} x_{bc} + m\, y_c \geq m + 1, \tag{54}$$

Selecting m to be -1, (2) reduces to:

$$\sum_{b \in \mathcal{B}} x_{bc} \geq y_c. \tag{55}$$

Equation (55) expresses the if-then statement "if $y_c = 1$ then $\sum_{b \in \mathcal{B}} x_{bc} \geq 1$" linearly. If $y_c = 1$, then, according to (55), $\sum_{b \in \mathcal{B}} x_{bc} \geq 1$. Note that the second if-then statement above is equivalent to $\sum_{b \in \mathcal{B}} x_{bc} = 0 \Longrightarrow y_c = 0$, which is already enforced by the objective function since it aims at minimizing the number of RICs. Hence, (55) is redundant.

Considering (53) and (55), it follows that (51) can be linearly reformulated as follows:

$$y_c \leq \sum_{b \in \mathcal{B}} x_{bc} \leq B y_c, \forall c \in \mathcal{C}. \tag{56}$$

B.LINEARIZING THE RESPONSE TIME CONSTRAINT

To reformulate (11), we introduce a binary variable $u_{bc}^{(\omega)}$ for each link between RAN node $b \in \mathcal{B}$ and RIC $c \in \mathcal{C}$, and each scenario $\omega \in \Omega$. $u_{bc}^{(\omega)} = 0$ if the response time of RIC c to RAN node b under scenario ω is less than δ, and $u_{bc}^{(\omega)} = 1$ otherwise. Then, (11) is equivalent to the following constraints:

$$\begin{aligned} &2\, t_{bc}\, x_{bc} + \frac{1}{\mu - \sum_{\acute{b} \in \mathcal{B}} r_{\acute{b}}^{(\omega)}\, x_{\acute{b}c}} - \delta + \epsilon \leq N_{bc}^{(\omega)}\, u_{bc}^{(\omega)}, \\ &\qquad \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega, \end{aligned} \tag{57}$$

$$\sum_{\omega \in \Omega} \left(1 - u_{bc}^{(\omega)}\right) \geq \alpha\, |\Omega|, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \tag{58}$$

where $N_{bc}^{(\omega)} = (2\ t_{bc} + \frac{1}{\mu - \sum_{\acute{b} \in \mathcal{B}} r_{\acute{b}}^{(\omega)}} - \delta + \epsilon)$ is an upper-bound for the left-hand-side of (57) and $\epsilon > 0$ is a small tolerance beyond which we regard the constraint as having been broken.

Constraint (57) is non-linear. It can be equivalently written as:

$$\begin{aligned} &2\, \mu\, t_{bc}\, x_{bc} \ -\ 2\, t_{bc}\, x_{bc} \sum_{\acute{b} \in \mathcal{B}} r_{\acute{b}}^{(\omega)}\, x_{\acute{b}c} - \mu\, N_{bc}^{(\omega)}\, u_{bc}^{(\omega)} \\ &\quad + N_{bc}^{(\omega)}\, u_{bc}^{(\omega)} \sum_{\acute{b} \in \mathcal{B}} r_{\acute{b}}^{(\omega)}\, x_{\acute{b}c} + (\delta - \epsilon) \sum_{\acute{b} \in \mathcal{B}} r_{\acute{b}}^{(\omega)}\, x_{\acute{b}c} \\ &\quad \leq \mu(\delta - \epsilon) - 1, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega. \end{aligned} \tag{59}$$

Equation (59) includes the non-linear terms $x_{bc}\, x_{\acute{b}c}$ and $u_{bc}^{(\omega)}\, x_{\acute{b}c}$. It can be equivalently expressed in a linear form as follows:

$$\begin{aligned} &2\, \mu\, t_{bc}\, x_{bc} - 2\, t_{bc} \sum_{\acute{b} \in \mathcal{B}} r_{\acute{b}}^{(\omega)}\, x_{b\acute{b}c} - \mu\, N_{bc}^{(\omega)}\, u_{bc}^{(\omega)} \\ &\quad + N_{bc}^{(\omega)} \sum_{\acute{b} \in \mathcal{B}} r_{\acute{b}}^{(\omega)}\, z_{b\acute{b}c}^{(\omega)} + (\delta - \epsilon) \sum_{\acute{b} \in \mathcal{B}} r_{\acute{b}}^{(\omega)}\, x_{\acute{b}c} \\ &\quad \leq \mu(\delta - \epsilon) - 1, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega. \end{aligned} \tag{60}$$

After introducing the new decision variables $x_{b\acute{b}c}$ and $z_{b\acute{b}c}^{(\omega)}$, $\forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega$, and adding the following constraints:

$$\begin{aligned} &x_{b\acute{b}c} \leq x_{bc}, \forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}, \\ &x_{b\acute{b}c} \leq x_{\acute{b}c}, \forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}, \\ &x_{b\acute{b}c} \geq x_{bc} + x_{\acute{b}c} - 1, \forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}, \\ &x_{b\acute{b}c} \geq 0, \forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}. \end{aligned} \tag{61}$$

$$
\begin{aligned}
&z_{b\acute{b}c}^{(\omega)} \leq u_{bc}^{(\omega)}, \forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega, \\
&z_{b\acute{b}c}^{(\omega)} \leq x_{\acute{b}c}, \forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega, \\
&z_{b\acute{b}c}^{(\omega)} \geq u_{bc}^{(\omega)} + x_{\acute{b}c} - 1, \forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega, \\
&z_{b\acute{b}c}^{(\omega)} \geq 0, \forall b, \acute{b} \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega. \qquad (62)
\end{aligned}
$$

Therefore, the sampled version of C^3P^2 can be equivalently written as an MILP as (14)-(28).

APPENDIX C
MILP REFORMULATION OF CPPA

The $\max(\cdot,\cdot)$ term in the second-stage problem objective function can be represented in a linear form by (i) introducing new positive decision variables, $v_{bc} \overset{\text{def}}{=} \max(2\ t_{bc}\ x_{bc} + \frac{1}{\mu - \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c}}, \delta), \forall b \in \mathcal{B}, \forall c \in \mathcal{C}$, (ii) introducing new binary decision variables, f_{bc} (equals one if the response time over link bc is greater than δ, and equals zero otherwise) and d_{bc} (equals one if the response time over link bc is less than δ, and equals zero otherwise), $\forall b \in \mathcal{B}, \forall c \in \mathcal{C}$, and (iii) adding the following constraints:

$$
\begin{aligned}
&v_{bc} \geq \delta, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \qquad (63) \\
&v_{bc} \geq \frac{1}{\mu}\ v_{bc} \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c} + 2\ t_{bc}\ x_{bc} + \frac{1}{\mu} \\
&\qquad - \frac{2}{\mu}\ t_{bc}\ x_{bc} \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \qquad (64) \\
&v_{bc} \leq \frac{1}{\mu}\ v_{bc} \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c} + 2\ t_{bc}\ x_{bc} + M(1 - f_{bc}) \\
&\qquad - \frac{M}{\mu}\left(\sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c} - f_{bc} \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c}\right) \\
&\qquad - \frac{2}{\mu}\ t_{bc}\ x_{bc} \sum_{\acute{b}\in\mathcal{B}} \tilde{r}_{\acute{b}}\ x_{\acute{b}c} + \frac{1}{\mu}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \qquad (65) \\
&v_{bc} \leq \delta + M(1 - d_{bc}), \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \qquad (66) \\
&f_{bc} + d_{bc} = 1, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \qquad (67) \\
&f_{bc}, d_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \qquad (68)
\end{aligned}
$$

where M is a sufficiently large number. The terms $x_{\acute{b}c}\ x_{bc} \overset{\text{def}}{=} x_{b\acute{b}c}$ and $x_{\acute{b}c}\ f_{bc} \overset{\text{def}}{=} f_{b\acute{b}c}$ can be linearized similar to (61) and (62). The term $x_{\acute{b}c}\ v_{bc}$, which represents a product of a binary variable with a positive continuous variable, can be reformulated by introducing the new decision variables $e_{b\acute{b}c}, \forall b, \acute{b} \in \mathcal{B}, c \in \mathcal{C}$, and adding the following constraints:

$$
\begin{aligned}
&e_{b\acute{b}c} \leq \max\left(2\ t_{bc} + \frac{1}{\mu - \sum_{b\in\mathcal{B}} \tilde{r}_b}, \delta\right) x_{\acute{b}c}, \\
&e_{b\acute{b}c} \geq \delta\ x_{\acute{b}c}, \\
&e_{b\acute{b}c} \geq v_{bc} - \max\left(2\ t_{bc} + \frac{1}{\mu - \sum_{b\in\mathcal{B}} \tilde{r}_b}, \delta\right)\left(1 - x_{\acute{b}c}\right), \\
&e_{b\acute{b}c} \leq v_{bc} - \delta\left(1 - x_{\acute{b}c}\right). \qquad (69)
\end{aligned}
$$

Therefore, the sampled version of CPPA can be equivalently written as an MILP as summarized in Section V-C.

REFERENCES

[1] "O-RAN alliance." 2022. [Online]. Available: https://www.o-ran.org/

[2] B. Balasubramanian et al., "RIC: A RAN intelligent controller platform for AI-enabled cellular networks," *IEEE Internet Comput.*, vol. 25, no. 2, pp. 7–17, Mar./Apr. 2021.

[3] "O-RAN architecture description: O-RAN.WG1.O-RAN-architecture-description-v07.00." O-RAN alliance. 2022. [Online]. Available: https://www.o-ran.org/blog/o-ran-alliance-introduces-53-new-specifications-released-since-july-2022

[4] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 473–478, Sep. 2012.

[5] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, 2013, pp. 672–675.

[6] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1339–1342, Aug. 2014.

[7] S. Liu, H. Wang, S. Yi, and F. Zhu, "NCPSO: A solution of the controller placement problem in software defined networks," in *Proc. 15th Int. Conf. Algorithms Archit. Parallel Process. (ICAPP)*, 2015, pp. 213–225.

[8] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. IEEE 25th Int. Teletraffic Congr. (ITC)*, 2013, pp. 1–9.

[9] S. Lange et al., "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 1, pp. 4–17, Mar. 2015.

[10] F. J. Ros and P. M. Ruiz, "On reliable controller placements in software-defined networks," *Comput. Commun. J.*, vol. 77, pp. 41–51, Mar. 2016.

[11] H. Li, P. Li, S. Guo, and A. Nayak, "Byzantine-resilient secure software-defined networks with multiple controllers in cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 4, pp. 436–447, Oct.—Dec. 2014.

[12] T. Y. Cheng, M. Wang, and X. Jia, "QoS-guaranteed controller placement in SDN," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2015, pp. 1–6.

[13] S. Auroux and H. Karl, "Flow processing-aware controller placement in wireless DenseNets," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. Conf. (PIMRC)*, 2014, pp. 1294–1299.

[14] Y. P. Llerena and P. R. L. Gondim, "SDN-controller placement for D2D communications," *IEEE Access*, vol. 7, pp. 169745–169761, 2019.

[15] F. Li, X. Xu, H. Yao, J. Wang, C. Jiang, and S. Guo, "Multi-controller resource management for software-defined wireless networks," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 506–509, Mar. 2019.

[16] S. Toufga, S. AbdelLatif, H. T. Assouane, P. Owezarski, and T. Villemur, "Towards dynamic controller placement in software defined vehicular networks," *Sensors*, vol. 20, no. 6, pp. 1701–1720, Mar. 2020.

[17] L. Alouache, S. Yassa, and A. Ahfir, "A multi-objective optimization approach for SDVN controllers placement problem," in *Proc. IEEE 13th Int. Conf. Netw. Future (NoF)*, 2022, pp. 1–9.

[18] M. J. Abdel-Rahman, E. A. Mazied, A. MacKenzie, S. Midkiff, M. R. Rizk, and M. El-Nainay, "On stochastic controller placement in software-defined wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2017, pp. 1–6.

[19] S. Chatterjee, M. J. Abdel-Rahman, and A. B. MacKenzie, "A joint optimization framework for network deployment and adaptive user assignment in indoor Millimeter wave networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7538–7554, Nov. 2021.

[20] M. J. Abdel-Rahman, E. A. Mazied, K. Teague, A. B. MacKenzie, and S. F. Midkiff, "Robust controller placement and assignment in software-defined cellular networks," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2017, pp. 1–9.

[21] S. Chatterjee, M. J. Abdel-Rahman, and A. B. MacKenzie, "On optimal orchestration of virtualized cellular networks with statistical multiplexing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 1, pp. 310–325, Jan. 2022.

[22] P. Kall and S. W. Wallace, *Stochastic Programming*, 1st ed. Hoboken, NJ, USA: Wiley, 1994.

[23] (IBM Technol. Co., Armonk, NY, USA). *IBM ILOG CPLEX Optimization Studio*. Accessed: Sep. 2023. [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio

[24] G. Wang, Y. Zhao, J. Huang, and Y. Wu, "An effective approach to controller placement in software defined wide area networks," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 344–355, Mar. 2018.

[25] G. Schütz and J. Martins, "A comprehensive approach for optimizing controller placement in software-defined networks," *Comput. Commun. J.*, vol. 159, pp. 198–205, Jun. 2020.

[26] R. Gholamrezaei, G. Mirjalily, and S. Emadi, "Learning-based multi-constraint resilient controller placement and assignment in software-defined networks using covering graph," *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 4, Apr. 2023, Art. no. e4742.

[27] D. He, J. Chen, and X. Qiu, "A density algorithm for controller placement problem in software defined wide area networks," *J. Supercomput.*, vol. 79, no. 5, pp. 5374–5402, 2023.

[28] G. M. Almeida et al., "RIC-O: Efficient placement of a disaggregated and distributed RAN intelligent controller with dynamic clustering of radio nodes," 2023, *arXiv:2301.02760.*

[29] Q. Qin, K. Poularakis, G. Iosifidis, S. Kompella, and L. Tassiulas, "SDN controller placement with delay-overhead balancing in wireless edge networks," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1446–1459, Dec. 2018.

[30] E. A. Mazied et al., "The wireless control plane: An overview and directions for future research," *J. Netw. Comput. Appl.*, vol. 126, pp. 104–122, Jan. 2019.

[31] J. Liu, Y. Shi, L. Zhao, Y. Cao, W. Sun, and N. Kato, "Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 221–232, Feb. 2018.

[32] R. Gouareb, V. Friderikos, A. H. Aghvami, and M. Tatipamula, "Joint reactive and proactive SDN controller assignment for load balancing," in *Proc. IEEE Glob. Commun. (GLOBECOM) Workshops*, 2019, pp. 1–6.

[33] T. Das, V. Sridharan, and M. Gurusamy, "A survey on controller placement in SDN," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 472–503, 1st Quart., 2020.

[34] A. Kumari and A. S. Sairam, "Controller placement problem in software-defined networking: A survey," *Networks*, vol. 78, no. 2, pp. 195–223, 2021.

[35] M. Dhar, A. Debnath, B. K. Bhattacharyya, M. K. Debbarma, and S. Debbarma, "A comprehensive study of different objectives and solutions of controller placement problem in software-defined networks," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 5, May 2022, Art. no. e4440.

[36] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1376–1411, 2nd Quart., 2023.

[37] Y. Niu, Y. Li, M. Chen, D. Jin, and S. Chen, "A cross-layer design for a software-defined millimeter-wave mobile broadband system," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 124–130, Feb. 2016.

[38] D. Lee, S. Zhou, and Z. Niu, "Spatial modeling of scalable spatially-correlated log-normal distributed traffic inhomogeneity and energy-efficient network planning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2013, pp. 1285–1290.

[39] D. Lee, S. Zhou, X. Zhong, Z. Niu, X. Zhou, and H. Zhang, "Spatial modeling of the traffic density in cellular networks," *IEEE Wireless Commun.*, vol. 21, no. 1, pp. 80–88, Feb. 2014.

[40] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, Apr. 1975.

[41] D. Gross, *Fundamentals of Queueing Theory*, 4th ed. Hoboken, NJ, USA: Wiley, 2008.

[42] S. Ahmed and A. Shapiro, "Solving chance-constrained stochastic programs via sampling and integer programming," *INFORMS Tuts. Oper. Res.*, pp. 261–269, Sep. 2008.

[43] J. Luedtke and S. Ahmed, "A sample approximation approach for optimization with probabilistic constraints," *SIAM Optim. J.*, vol. 19, no. 2, pp. 674–699, May 2008, doi: https://doi.org/10.1137/070702928.

[44] M. J. Abdel-Rahman, A. M. AlWaqfi, J. K. Atoum, M. A. Yaseen, and A. B. MacKenzie, "A novel multi-objective sequential resource allocation optimization for UAV-assisted VLC," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6896–6901, May 2023.

[45] B. Pagnoncelli, S. Ahmed, and A. Shapiro, "Sample average approximation method for chance constrained programming: Theory and applications," *J. Optim. Theory Appl.*, vol. 142, no. 2, pp. 399–416, Aug. 2009.

[46] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Optim. J.*, vol. 17, no. 4, pp. 969–996, 2007.

[47] Q. Wang, Y. Guan, and J. Wang, "A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 206–215, Feb. 2012.

[48] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM Optim. J.*, vol. 12, no. 2, pp. 479–502, Feb. 2002.

[49] S. Ahmed and A. Shapiro, "The sample average approximation method for stochastic programs with integer recourse," School Ind. Syst. Eng., Georgia Inst. Technol., Atlanta, GA, USA, 2002. [Online]. Available: http://www.optimizationonline.org/

[50] A. Bianco, R. Birke, L. Giraudo, and M. Palacin, "OpenFlow switching: Data plane performance," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2010, pp. 1–5.

[51] J. Linderoth. "Lecture notes on integer programming." 2005. [Online]. Available: https://jlinderoth.github.io/classes/ie418/lecture2.pdf

MOHAMMAD J. ABDEL-RAHMAN (Senior Member, IEEE) has been an Associate Professor with the Data Science Department, Princess Sumaya University for Technology since October 2022 and an Adjunct Assistant Professor with the Electrical and Computer Engineering (ECE) Department, Virginia Tech since January 2018. Prior to joining his current position, he worked as a Research Faculty Member with the ECE Department, Virginia Tech from January 2015 to September 2017, and later on, as an Assistant and then an Associate Professor with the Electrical Engineering and Computer Science Departments, Al Hussein Technical University (HTU) from October 2017 to October 2022. During his time at HTU, he held various leadership positions, including the Inaugural Chair of the Electrical Engineering Department from October 2018 to March 2022 and the Director of the Wireless Networks and Security Research Laboratory from April 2019 to October 2022. His current research interests primarily revolve around the integration of artificial intelligence/machine learning and operations research in the fields of communications, networking, energy, healthcare, and transportation.

Dr. Abdel-Rahman actively participates in the academic community as an Associate Editor of IEEE ACCESS, an Editor of *Wireless Personal Communications* (Springer Nature), and a Technical Program Committee member of several international conferences. He is a member of the Global Young Academy.

EMADELDIN A. MAZIED (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the Computer Science Department, Virginia Tech, USA. He is also an Assistant Lecturer (on leave) with Sohag University, Egypt. His research interests are resource allocation, scheduling, reinforcement learning, and their applications in the next-generation wireless networks.

FAHID HASSAN received the bachelor's degree in electrical engineering from the Jordan University of Science and Technology, Ar-Ramtha, Jordan, in 2020. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, Rice University. From 2019 to 2020, he was a Research Assistant with the Electrical Engineering Department, Al Hussein Technical University, Amman, Jordan. His research interests include wireless communications, metasurfaces, security, and optimization.

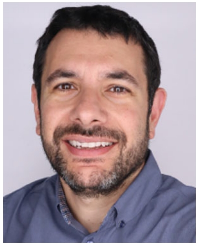
KORY TEAGUE (Member, IEEE) received the Master of Science degree in electrical engineering from Virginia Tech in 2018. He is an Electrical Engineer with the U.S. Naval Research Laboratory.

ALLEN B. MACKENZIE (Senior Member, IEEE) joined Tennessee Tech as the Chair in August 2019, and a Professor with the Department of Electrical and Computer Engineering. Prior to joining Tennessee Tech, he was a Professor with the Department of Electrical and Computer Engineering, Virginia Tech, where he was a Faculty Member from 2003 to 2019, where he was the Associate Director of Wireless @ Virginia Tech. From 2012 to 2013, he was an E. T. S. Walton Visiting Professor with Trinity College Dublin. He is the author of more than 90 refereed conference and journal articles and a coauthor of the book *Game Theory for Wireless Engineers*. His research interests include wireless communications systems and networks. His current research interests include integration of millimeter-wave technology into networks, cognitive radio and cognitive network architectures, and the analysis of wireless systems and networks using game theory and stochastic optimization.

Dr. Mackenzie was on the Editorial Board of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON COMMUNICATIONS, and IEEE TRANSACTIONS ON MOBILE COMPUTING. He was a member of the U.S. Department of Commerce's Spectrum Management Advisory Committee from 2016 to 2018. He is a member of ASEE and ACM.

SCOTT F. MIDKIFF (Life Senior Member, IEEE) is a Professor with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. From 2009 to 2012, he was the Department Head of the Bradley Department of Electrical and Computer Engineering, Virginia Tech. From 2006 to 2009, he was a Program Director of the National Science Foundation. His research interests include wireless and ad hoc networks, network services for pervasive computing, and cyber–physical systems.

KLEBER V. CARDOSO received the degree in computer science from the Universidade Federal de Goiás (UFG) in 1997, and the M.Sc. and Ph.D. degrees in electrical engineering from COPPE, Universidade Federal do Rio de Janeiro in 2002 and 2009, respectively. He is an Associate Professor with the Institute of Informatics, UFG, where he has been serving since 2009. He spent his sabbaticals at Virginia Tech, USA, in 2015 and Inria Saclay Research Center, France, in 2020. He has participated in some international research projects (including two from joint calls BR-EU) and coordinated several national-sponsored research and development projects. His research spans wireless networks, virtualization, resource allocation, and performance evaluation.



DIMITRIOS S. NIKOLOPOULOS (Fellow, IEEE) is the John W. Hancock Professor of Engineering with Virginia Tech. He conducts research in high-performance computing and computer systems. His work has impacted the OpenMP parallel programming standard, RedHat Linux, and various commercial system software products. He has received numerous prestigious research awards, including the Royal Society Wolfson Research Merit Award, the NSF CAREER Award, the DOE Early Career Principal Investigator Award, and the Faculty Awards from IBM, Cisco, and Sony. He is a Fellow of AAIA, BCS, and IET and a Distinguished Member of ACM.