A Stochastic Optimization Framework for Joint RAN Intelligent Controller Placement and RAN Nodes Assignment in O-RAN Networks

Mohammad Abdel-Rahman ¹, EMADELIN MAZIED ², FAHID HASSAN ², Kory Teague ², ALLEN MACKENZIE ², Scott Midkiff ², and Kleber V. Cardoso ²

¹Princess Sumaya University for Technology
²Affiliation not available

December 16, 2023
A Stochastic Optimization Framework for Joint RAN Intelligent Controller Placement and RAN Nodes Assignment in O-RAN Networks

MOHAMMAD J. ABDEL-RAHMAN, EMADELDIN A. MAZIED, FAHID HASSAN, KORY TEAGUE, ALLEN B. MACKENZIE, SCOTT F. MIDKIFF, and KLEBER V. CARDOSO

Abstract—O-RAN Architecture is consolidating the concept of software-defined cellular networks (SDCNs) for beyond 5G networks, mainly through the introduction of the Near-Real-Time RAN Intelligent Controller (Near-RT RIC) and the xApps. In this context, the deployment of the Near-RT RIC faces the traditional controller placement problem in an SDCN, which considers the number of controllers, their placement, and the assignment of RAN nodes to the controllers. In this paper, we study the controller placement problem in SDCNs, considering the uncertainty in user locations. Specifically, our contributions are as follows. First, we develop C^P^2, a robust static joint controller placement and RAN node-controller assignment scheme. The objective of C^P^2 is to minimize the number of controllers needed to control all RAN nodes while ensuring that the response time to each RAN node will not exceed δ seconds with a probability greater than β. Second, we develop CPPA, a robust joint controller placement and adaptive RAN node-controller assignment scheme. In contrast to C^P^2, CPPA enjoys a recourse capability, where the RAN node-controller 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 (MILP) formulation for C^P^2. Two-stage stochastic optimization with recourse, combined with several linearization techniques, is used to develop an MILP formulation for CPPA. The optimal performance of C^P^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 C^P^2 and CPPA. Our results demonstrate the robustness of the proposed stochastic resource allocation schemes for SDCNs 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—Software-defined cellular networks, RAN controller, controller placement problem, chance-constrained stochastic optimization, two-stage stochastic optimization with recourse, sample average approximation.

I. INTRODUCTION

THE O-RAN Alliance [1] has been changing the RAN industry by proposing open, virtualized, interoperable, and intelligent mobile cellular networks. Among the O-RAN specifications, its new RAN architecture is one of the key contributions. This architecture combines concepts from SDN and NFV, in addition to cloud-native and Artificial Intelligence (AI) / Machine Learning (ML) technologies [2]. From SDN, O-RAN architecture [3] employs concepts such as control and data planes separation and the adoption of a remote RAN controller. The O-RAN architecture splits the controller into two main parts: 1) Near-Real-Time RAN Intelligent Controller (Near-RT RIC) for time-sensitive operations and 2) Non-Real-Time RAN Intelligent Controller (Non-RT RIC) for operations that present looser time constraints.

A RAN controller runs AI/ML-based applications that establish control loops with the RAN nodes under their responsibility. Non-RT RIC runs applications (named rApps) that can deal with control loop latency above 1 s. Near-RT RIC runs applications (named xApps) that employ control loops constrained to time intervals between 10 ms and 1 s. The response time constraints of a control loop is related to the RAN function under the management of a specific xApp. In a large RAN, the Near-RT RIC and latency-sensitive xApps running over it must be replicated and a certain number of RAN nodes needs to be assigned to each controller. Optimal placement of SDN (or RAN) controllers has a prominent effect on minimizing this response time. Distributing a minimum number of controllers at optimal locations to complete control functions in a timely manner is known as the controller placement problem (CPP). The CPP is well studied in wired networks (examples include [4]–[12]) using different objectives and constraints that are related to the network latency, reliability, and load balancing. The problem becomes even more challenging considering the dynamics of a mobile wireless network.

Channel uncertainty and user mobility are two key characteristics of wireless networks that impose unique challenges on the CPP in software-defined wireless networks (SDWNs). Recently, to achieve an energy-efficient architecture for 5G networks, the authors in [13] have proposed an SDN-based power management framework for 5G heterogeneous networks, in which the SDN controller communicates with each assigned base station through a wireless link. This makes the controller response time uncertain (stochastic). User mobility is another big challenge, particularly in SDCN, which creates another
source of uncertainty for the CPP. Specifically, the distribution of mobile users across RAN nodes will be stochastic, making the request rate from RAN nodes to the RAN controllers uncertain.

Several works have been proposed to address the CPP in SDWN (examples include [14]–[16]). However, these works neither assume wireless links between the RAN controllers and their controlled elements nor account for users’ mobility. Using chance-constrained stochastic programming [17], the authors in [18] have provided the first solution to the optimal controller placement and assignment problem under uncertainty on the wireless channel between the controller and its controlled nodes.

Stochastic programming provides a powerful mathematical tool to handle optimization under uncertainty. It has been exploited to optimize resource allocation in various types of wireless networks operating under uncertainties (examples include [18]–[21])

Main Contributions:

In this paper, we develop a mathematical framework for formulating and solving the CPP in SDCN, while considering the uncertainty in the geographical distribution of mobile users (and hence, RAN node request rates). Specifically:

- We develop a robust static joint controller placement and assignment scheme (denoted by C3P2): Using chance-constrained stochastic programming, we formulate a static joint stochastic controller placement and RAN node-controller assignment problem that is robust to the variations in the mobile user locations (and hence the RAN node request rates). The objective of C3P2 is to minimize the number of RAN controllers needed to control all RAN nodes while ensuring that the response time to each RAN node will exceed δ seconds with a probability less than 1 − β.

- We develop a robust joint controller placement and adaptive assignment scheme (denoted by CPPA): Using two-stage stochastic programming, we formulate a joint stochastic controller placement and adaptive RAN node-controller assignment problem. In contrast to C3P2, where the RAN node-controller assignment is static, CPPA enjoys a recourse capability, where the RAN node-controller 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 RAN controllers. Our optimality criteria are: (i) minimizing the number of controllers and (ii) minimizing the response time to various RAN nodes. In contrast to C3P2, 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-controller assignment is optimized under each realization of RAN node request rates aiming at minimizing the response time to various RAN nodes.

- To evaluate and study our proposed C3P2 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 C3P2 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.
  - We solve the mixed integer linear programs using CPLEX.
  - 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 C3P2 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, C3P2 is presented, mathematically formulated, and reformulated as an MILPP. An SAA-based algorithm is also developed in Section IV to efficiently solve C3P2. In Section V, CPPA is presented, mathematically formulated, and reformulated as an MILPP. Furthermore, an SAA-based algorithm is developed in Section V to solve CPPA efficiently. C3P2 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 I.

II. LITERATURE REVIEW

Several models have been proposed for the CPP in software-defined wired and wireless networks. In this section, we will review some of the important contributions in this domain. For more references about the CPP in SDN, including radio access networks, please see [22]–[24] and references therein.

Several works have been proposed to address the CPP in wired networks with different objectives and constraints (e.g., latency, reliability, and load balancing). In [4], the authors examined fundamental limits to control plane propagation latency considering different WAN topologies. They concluded that one controller location is often sufficient to meet existing reaction-time requirements (though certainly not fault tolerance requirements). Considering the queuing latency in addition to the propagation latency, the authors in [25] introduced the concept of network partitioning to decrease the end-to-end latency. They proposed a clustering-based network partitioning algorithm. To mitigate the reliability issues that may result from decoupling the control and forwarding planes, the authors in [5] considered the CPP with the objective of maximizing the reliability of the control network. To quantify the impact of the number of controllers on the reliability
Sets:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{B}$</td>
<td>Set of RAN node forming the considered mobile network.</td>
</tr>
<tr>
<td>$C$</td>
<td>Set of candidate locations for deploying RAN controllers.</td>
</tr>
</tbody>
</table>

Data:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>Request rate of each mobile user.</td>
</tr>
<tr>
<td>$r_b$</td>
<td>Request rate of RAN node $b$.</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Scaling parameter of the log-normal distribution.</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Location parameter of the log-normal distribution.</td>
</tr>
<tr>
<td>$\rho(S)(x,y)$</td>
<td>Standardized version of the Gaussian stochastic field $\rho(G)(x,y)$.</td>
</tr>
<tr>
<td>$\nu_b, \nu_l$</td>
<td>Uniform stochastic variables between 0 and $\omega_{\text{max}}$ that represent angular frequencies.</td>
</tr>
<tr>
<td>$\phi_l, \phi_l$</td>
<td>Uniform stochastic variables between 0 and $2\pi$ that represent phases.</td>
</tr>
<tr>
<td>$L_b$</td>
<td>Number of stochastic sinusoidal fields used to generate $\rho(G)(x,y)$.</td>
</tr>
<tr>
<td>$t_{bc}$</td>
<td>Sum of transmission and propagation delays over the link between RAN node $b$ and a controller at location $c$.</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Controller service rate (a.k.a. processing capacity).</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Requested minimum probability of delay satisfaction.</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Requested upper-bound on the controllers’ response time.</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>A set of i.i.d. samples (scenarios) generated from the distribution of the RAN node request rates.</td>
</tr>
<tr>
<td>$\theta_{bc}$</td>
<td>A design coefficient for the link between RAN node $b$ and a controller at location $c$, introduced to balance the tradeoff between minimizing the number of controllers and minimizing the response time to the RAN nodes.</td>
</tr>
</tbody>
</table>

Decision Variables:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{bc}$</td>
<td>A binary decision variables; it equals one if a RAN controller is placed at location $c$ to control RAN node $b$, and it equals zero otherwise.</td>
</tr>
<tr>
<td>$y_c$</td>
<td>A binary decision variables; it equals one if a RAN controller is placed at location $c$, and it equals zero otherwise.</td>
</tr>
<tr>
<td>$u_{bc}^{(\omega)}$</td>
<td>A binary decision variable; it equals zero if the response time of RAN controller $c$ to RAN node $b$ under scenario $\omega$ is less than $\delta$, and it equals one otherwise.</td>
</tr>
<tr>
<td>$x_{bc}^{(\omega)}$</td>
<td>$x_{bc} \cdot s_{bc}$.</td>
</tr>
<tr>
<td>$z_{bc}^{(\omega)}$</td>
<td>$u_{bc} \cdot s_{bc}$.</td>
</tr>
<tr>
<td>$v_{bc}$</td>
<td>A positive decision variable; it equals max (delay over the link $bc, \delta$).</td>
</tr>
<tr>
<td>$f_{bc}$</td>
<td>A binary decision variable; it equals one if the response time over the link $bc$ is greater than $\delta$, and it equals zero otherwise.</td>
</tr>
<tr>
<td>$d_{bc}$</td>
<td>$1 - f_{bc}$.</td>
</tr>
<tr>
<td>$j_{bc}$</td>
<td>$x_{bc} \cdot f_{bc}$.</td>
</tr>
<tr>
<td>$\epsilon_{bc}$</td>
<td>$x_{bc} \cdot v_{bc}$.</td>
</tr>
</tbody>
</table>

TABLE I
NOTATION.
framework that takes into account the data flow processing and control applications. In [30], the authors proposed using SDN in wireless edge networks to enhance their control capability. They modeled the CPP in wireless edge networks, presented approximate, close-to-optimal, solutions to the CPP, and analyzed the interplay between different performance and reliability objectives. The authors in [31] provided an overview of the wireless SDN control plane and reviewed its deployment in recent 5G networks. They also proposed a generic framework for a wireless SDN control plane using deep reinforcement learning principles. Furthermore, the authors in [31] presented a road map for future research. In [32], the authors considered the problem of joint placement of SDN controllers and satellite gateways in 5G-satellite integrated networks. They developed a simulated annealing and clustering hybrid algorithm to solve the joint placement problem. Based on extensive experiments on real network topologies, the authors demonstrated the ability of their algorithm to achieve approximate performance with much lower computational complexity compared to the enumeration algorithms. The authors in [33] considered the CPP for an LTE-A architecture with SDN-based device-to-device (D2D) communications management. They studied the effect of D2D communications management on the number of SDN controllers used in a cellular network. The authors in [34] jointly optimized the reactive and proactive assignment of SDN controllers to the controlled elements, aiming to balance the load in the control plane.

In [35], the authors introduce an SDWN scheme focusing on energy-aware multi-controller placement and latency-aware resource management. The proposed approach uses particle swarm optimization for multi-controller placement and a deep reinforcement learning algorithm for resource allocation, reducing execution time and energy consumption for tasks in the SDWN. In [36], the authors considered the CPP in software-defined vehicular networks (SDVNs). They proposed a dynamic controller placement scheme that adaptively adjusts the number and placement of controllers according to road traffic fluctuations. The authors in [37] propose a multi-objective optimization technique employing a genetic algorithm for selecting optimal controllers also in software-defined vehicular networks. The proposed approach presents promising results through simulations by considering important metrics, including control path latency, load balancing, and clock offset for synchronization. In [38], the authors investigate the CPP in the context of O-RAN, formalizing the problem of minimizing the overall cost of the placement of Near-RT RIC while ensuring the latency-sensitive control loop. The authors explore the cloud-native properties of the O-RAN architecture to replicate only the time-sensitive components. Figure 1 provides a comprehensive overview of the existing studies in controller placement strategies for SDNs and SDWNS.

Up to our knowledge, all existing works on the CPP in SDCNs (except [18]) either considered wired links between the SDN controllers and the controlled elements or assumed wireless links but did not account for the uncertainty in the availability of these links and the delay over them. The authors in [18] conducted the first study of the ‘wireless CPP’ when the links between the controllers and the controlled nodes are wireless. They used chance-constrained stochastic optimization [17] to optimize the placement of the RAN controllers and their assignment to the controlled elements under the uncertainty in the availability of the wireless links between the controllers and the controlled nodes. In this paper, we extend our proposed framework in [20] for the CPP in SDCNs, in which the RAN nodes are controlled by RAN controllers, e.g., Near-RT RICs. In fact, we also update the terminology to facilitate the understanding of our work in the context of O-RAN networks.

III. MODELS, ASSUMPTIONS, AND PROBLEM STATEMENT

A. System Model

We consider a set \( B = \{1, 2, \ldots, B\} \) 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 [39], a RAN node can be an O-RAN-compliant LTE evolved Node Base (eNB) or a component of a Next Generation Node Base (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 [39]. 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 [40]. 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 \( r_b \).

B. Distribution of RAN node Request Rates

To know the distribution of \( r_b, \forall b \in B \), we want to know the distribution of the mobile user locations. In [41], 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 [42] 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 [42], 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 [42], each point represents a user, or a collection of users within a pixel surrounding the point, with the specified demand. It is assumed in [42] 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
Our Work

| Control plane placement (CPP) in software-defined networks (SDNs) |
|---------------------|---------------------|
| CPP in wired networks (SDNs) | CPP in wireless networks (SDWNs) |
| | Flow processing aware CP [14] |
| Latency [4], [25], [26], [27], [20] | SDN in wireless edge networks [30] |
| Reliability [5], [10], [30] | Overview of wireless SDN control plane [31] |
| Load balancing [6], [7], [26], [37] | Joint placement of SDN controllers and satellite gateways [32] |
| Density-based controller placement algorithm [29] | CPP in LTE-A architecture with D2D communications [33] |
| | Load balancing [34] |
| | Energy-aware multi-controller placement [35] |
| | Disaggregated CPP in O-RAN [38] |
| | Our Work |
| | Wireless CPP with uncertainty handling |
| | Extended framework in [20] |
| | Chance constrained stochastic optimization |
| | Applicable to O-RAN networks |

Fig. 1. Taxonomy of controller placement strategies in SDNs and SDWNs.

demand and is placed according to the log-normal density function. The model is extended, removing the grid of points and maintaining the model as a continuous density function, which acts as the parameter \( \lambda(x, y) \) for a two-dimensional, non-stationary (inhomogeneous) Poisson point process (PPP). From the model, \( \lambda(x, y) \) is defined as:

\[
\lambda(x, y) = e^{(\sigma \rho^{(S)}(x, y) + \gamma)}, \quad (1)
\]

where \( \sigma \) and \( \gamma \) 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:

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

where angular frequencies \( i_l \) and \( j_l \) are uniform stochastic variables between 0 and \( \omega_{\text{max}} \), and phases \( \phi_l \) and \( \psi_l \) are uniform stochastic variables between 0 and \( 2\pi \). \( L \) 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 \( \lambda_{\text{max}} \), the maximum value of \( \lambda(x, y) \) over the considered domain, is first generated. Then, each point \((x_i, y_i)\) is retained within the process only with probability \( \lambda(x_i, y_i) / \lambda_{\text{max}} \).

### C. Queuing Delay at the RAN Controllers

In addition to the transmission (and propagation) delays, given by \( 2t_{bc} \) for the link between RAN node \( b \) and controller \( c \), a RAN node will encounter a queuing delay at the controller. We model each controller as an M/M/1 queuing system [43], under which the mean service time (say, of controller \( c \)) can be expressed as [44]:

\[
E[D_c] = \frac{1}{\mu - \sum_{b \in \beta} \tilde{r}_b x_{bc}}, \quad (3)
\]

where \( \mu \) is the controller service rate (a.k.a. controller processing capacity), \( \tilde{r}_b \) is the request rate of RAN node \( b \).\(^1\)

\(^1\)Note that \( \mu \) of each controller needs to be greater than the total request rate from all RAN nodes assigned to that controller, i.e., the denominator of (3) needs to be positive for all \( c \in C \).
D. Problem Statement

Our objective in this paper is to find the minimum number of RAN controllers, their optimal locations, and the optimal assignment of these controllers 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 controller is placed at location \( c \) to control RAN node \( b \), and equals zero otherwise.

IV. ROBUST STATIC JOINT CONTROLLER PLACEMENT AND RAN NODE-CONTROLLER ASSIGNMENT (\( C^3P^2 \))

In this section, we present our static joint RAN controller placement and RAN node-controller assignment scheme, referred to as \( C^3P^2 \).

A. Problem Formulation

\( C^3P^2 \) aims to find the optimal placement of the minimum number of controllers and the optimal assignment of these controllers to the RAN nodes, to ensure that the response time to each RAN node will exceed \( \delta \) seconds with probability less than \( 1 - \beta \). We mathematically formulate \( C^3P^2 \) using chance-constrained stochastic optimization [17]. We formulate \( C^3P^2 \) while considering the uncertainty in \( \hat{r}_b, b \in \mathcal{B} \). We say that \( C^3P^2 \) is a robust scheme; because it accounts for the variations in \( \hat{r}_b, b \in \mathcal{B} \), resulting from variations in the mobile user locations. Mathematically, \( C^3P^2 \) can be stated as:

\[
\begin{align*}
\text{minimize} & \quad \sum_{c \in \mathcal{C}} \mathbb{1}\{\sum_{b \in \mathcal{B}} x_{bc} \geq 1\} \\
\text{subject to:} & \quad \sum_{c \in \mathcal{C}} x_{bc} = 1, \forall b \in \mathcal{B}, \quad \forall x_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C},
\end{align*}
\]

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

B. \( C^3P^2 \) with Sampled Request Rate Distribution

Chance-constrained stochastic programs are largely intractable due to the difficulty in checking the feasibility of a particular solution [45]. In other words, for a given \( x_{bc}, b \in \mathcal{B}, c \in \mathcal{C} \), computing \( \Pr \left\{ \frac{1}{2} \frac{t_{bc} x_{bc} + \frac{1}{\mu - \delta \hat{r}_b}}{\hat{r}_b} \leq \delta \right\} \geq \beta \) 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 \( \Omega \) 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 \( \sum_{\omega \in \Omega} \mathbb{1}\{\sum_{b \in \mathcal{B}} x_{bc} \geq 1\} \geq \beta \), where \( \hat{r}_b^{(\omega)} \) is the request rate of RAN node \( b \) under scenario \( \omega \). \( C^3P^2 \) with sampled request rate distribution is given by:

\[
\begin{align*}
\text{minimize} & \quad \sum_{c \in \mathcal{C}} \mathbb{1}\{\sum_{b \in \mathcal{B}} x_{bc} \geq 1\} \\
\text{subject to:} & \quad \sum_{c \in \mathcal{C}} x_{bc} = 1, \forall b \in \mathcal{B}, \quad \forall x_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C},
\end{align*}
\]

where \( \alpha \in (0, 1) \) and it can be different from \( \beta \). There are several advantages for solving the sampled version of \( C^3P^2 \), as summarized in [20]. One of these advantages is to get a lower bound on the required number of controllers. Specifically, if \( \hat{o}^{(\omega)}_b \) and \( \hat{\delta}^{(\omega)}_c \) are the optimal objective function values (i.e., the
minimum number of controllers) of C³P² and its sampled version, respectively. Then, it has been shown that \( \tilde{\alpha} \leq \alpha^* \) with probability at least \( 1 - \eta \) if \( |\Omega| \geq \frac{1}{2(\beta - \alpha^2)} \ln \left( \frac{1}{\eta} \right) \) and \( \alpha < \beta \) [46]².

In the next subsection, we use several linearization techniques to convert the sampled version of C³P² into a mixed integer linear program (MILPP), in order to solve it using CPLEX.

C. Mixed Integer Linear Reformulation

First, note that the objective function (8) is non-linear. It can be represented in a linear form by introducing new binary decision variables, \( y_c \) def \( = 1 \{ \sum_{b \in B} x_{bc} \geq 1 \} \), \( \forall c \in C \), and reformulating the indicator function as follows [47]:

- If \( \sum_{b \in B} x_{bc} \geq 1 \) then \( y_c = 1 \) can be reformulated as:
  \[
  \sum_{b \in B} x_{bc} - (M + \epsilon) y_c \leq 1 - \epsilon, \quad (12)
  \]

where \( M \) is an upper bound for \( \sum_{b \in B} x_{bc} - 1 \) and \( \epsilon > 0 \) is a small tolerance beyond which we regard the constraint as having been broken. Selecting \( M \) and \( \epsilon \) to be \( B - 1 \) and 1, respectively, (12) reduces to \( \sum_{b \in B} x_{bc} \leq B y_c \).

- If \( y_c = 1 \) then \( \sum_{b \in B} x_{bc} \geq 1 \) can be reformulated as:
  \[
  \sum_{b \in B} x_{bc} + m y_c \geq m + 1, \quad (13)
  \]

where \( m \) is a lower bound for \( \sum_{b \in B} x_{bc} - 1 \). Selecting \( m \) to be \(-1\), (13) reduces to \( \sum_{b \in B} x_{bc} \geq y_c \).

Therefore,

\[
y_c = 1 \{ \sum_{b \in B} x_{bc} \geq 1 \} \iff y_c \leq \sum_{b \in B} x_{bc} \leq B y_c, \forall c \in C.
\]

To reformulate (10), we introduce a binary variable \( u_{bc}^{(\omega)} \) for each link between RAN node \( b \in B \) and RAN controller \( c \in C \), and each scenario \( \omega \in \Omega \). If the response time of controller \( c \) to RAN node \( b \) under scenario \( \omega \) is less than \( \delta \), and \( t_{bc}^{(\omega)} = 1 \) otherwise. Then, (10) is equivalent to the following constraints:

\[
2 t_{bc} x_{bc} - \frac{1}{\mu - \sum_r r_b^{(\omega)} x_{bc}} - \delta + \epsilon \leq N_{bc}^{(\omega)} u_{bc}^{(\omega)}, \quad \forall b \in B, \forall c \in C, \forall \omega \in \Omega, \quad (14)
\]

\[
\sum_{\omega \in \Omega} \left( 1 - u_{bc}^{(\omega)} \right) \geq \alpha \left| \Omega \right|, \forall b \in B, \forall c \in C, \quad (15)
\]

where \( N_{bc}^{(\omega)} = \left( 2 t_{bc} + \frac{1}{\mu - \sum_r r_b^{(\omega)} x_{bc}} - \delta + \epsilon \right) \) is an upper-bound for the left-hand-side of (14) and \( \epsilon > 0 \) is a small tolerance beyond which we regard the constraint as having been broken.

²For example, if we select \( \alpha \) to be smaller than \( \beta \) by 0.15, then solving the sampled version of C³P² with \( |\Omega| \geq \frac{1}{2(0.85 - 0.2^2)} \ln \left( \frac{1}{0.05} \right) \approx 51 \) scenarios will provide a lower bound on the required number of controllers for C³P² with probability at least 0.9. In Section VI, we solve the sampled version of C³P² with \( |\Omega| = 100 \) scenarios.

³Note that this condition is equivalent to \( \sum_{b \in B} x_{bc} = 0 \iff y_c = 0 \), which is already enforced by the objective function since it aims at minimizing the number of controllers. Hence, (13) is redundant.

Constraint (14) is non-linear. It can be equivalently written as:

\[
2 \mu t_{bc} x_{bc} - 2 t_{bc} x_{bc} \sum_{b \in B} r_b^{(\omega)} x_{bc} - \mu N_{bc}^{(\omega)} u_{bc}^{(\omega)}
+ N_{bc}^{(\omega)} u_{bc}^{(\omega)} \sum_{b \in B} r_b^{(\omega)} x_{bc} + (\delta - \epsilon) \sum_{b \in B} r_b^{(\omega)} x_{bc}
\leq \mu (\delta - \epsilon) - 1, \forall b \in B, \forall c \in C, \forall \omega \in \Omega. \quad (16)
\]

Equation (16) includes the non-linear terms \( x_{bc} x_{bc} \) and \( u_{bc}^{(\omega)} x_{bc} \). It can be equivalently expressed in a linear form as follows:

\[
2 \mu t_{bc} x_{bc} - 2 t_{bc} \sum_{b \in B} r_b^{(\omega)} x_{bc} - \mu x_{bc} u_{bc}^{(\omega)}
+ N_{bc}^{(\omega)} x_{bc} + (\delta - \epsilon) \sum_{b \in B} r_b^{(\omega)} x_{bc}
\leq \mu (\delta - \epsilon) - 1, \forall b \in B, \forall c \in C, \forall \omega \in \Omega. \quad (17)
\]

After introducing the new decision variables \( x_{bbc}^{(\omega)} \) and \( z_{bbc}^{(\omega)} \), (17) reduces to:

\[
\begin{align*}
x_{bbc}^{(\omega)} &\leq x_{bc}, \forall b \in B, \forall c \in C, \forall \omega \in \Omega, \\
x_{bbc}^{(\omega)} &\leq x_{bc}^{(\omega)}, \forall b \in B, \forall c \in C, \forall \omega \in \Omega, \\
x_{bbc}^{(\omega)} &\geq x_{bc}^{(\omega)} + x_{bc} - 1, \forall b \in B, \forall c \in C, \forall \omega \in \Omega, \\
x_{bbc}^{(\omega)} &\geq 0, \forall b \in B, \forall c \in C, \forall \omega \in \Omega.
\end{align*}
\]

(18)

Therefore, the sampled version of C³P² can be equivalently written as an MILPP as follows:

MILPP Reformulation of the Sampled Version of C³P²

\[
\begin{align*}
\text{minimize} \quad & y_c \quad \sum_{c \in C} y_c \quad (20) \\
\text{subject to:} \quad & \sum_{c \in C} x_{bc} = 1, \forall b \in B, \quad (21) \\
& \sum_{b \in B} x_{bc} \leq B y_c, \forall c \in C, \quad (22) \\
& 2 \mu t_{bc} x_{bc} - 2 t_{bc} \sum_{b \in B} r_b^{(\omega)} x_{bc} - \mu x_{bc} u_{bc}^{(\omega)}
+ N_{bc}^{(\omega)} x_{bc} + (\delta - \epsilon) \sum_{b \in B} r_b^{(\omega)} x_{bc}
\leq \mu (\delta - \epsilon) - 1, \forall b \in B, \forall c \in C, \forall \omega \in \Omega, \quad (23) \\
& \sum_{\omega \in \Omega} \left( 1 - u_{bc}^{(\omega)} \right) \geq \alpha \left| \Omega \right|, \forall b \in B, \forall c \in C, \quad (24)
\end{align*}
\]
\[
\begin{align*}
\text{(25)} & \quad x_{bbc} \leq x_{bc}, \forall b, h \in B, \forall c \in C, \\
\text{(26)} & \quad x_{bbc} \leq x_{bc}, \forall b, h \in B, \forall c \in C, \\
\text{(27)} & \quad x_{bbc} \geq x_{bc} + x_{bc} - 1, \forall b, h \in B, \forall c \in C, \\
\text{(28)} & \quad z_{\text{bbc}}^{(a)} \leq u_{bc^{(a)}}, \forall b, h \in B, \forall c \in C, \forall \omega \in \Omega, \\
\text{(29)} & \quad z_{\text{bbc}}^{(a)} \leq x_{bc}, \forall b, h \in B, \forall c \in C, \forall \omega \in \Omega, \\
\text{(30)} & \quad \Omega_{x} \in \{0, 1\}, \forall b \in B, \forall c \in C, \\
\text{(31)} & \quad \bar{v}_{bc}^{(a)} \in \{0, 1\}, \forall b \in B, \forall c \in C, \forall \omega \in \Omega, \\
\text{(32)} & \quad x_{bbc}, z_{bbc}^{(a)} \geq 0, \forall b, h \in B, \forall c \in C, \forall \omega \in \Omega. \\
\end{align*}
\]

This MILPP formulation can be solved optimally using CPLEX.

\section*{D. Sample Average Approximation (SAA) Algorithm for \(C^3P^2\)}

In the previous subsection, we converted the sampled version of \(C^3P^2\) 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 \(C^3P^2\) (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 \(C^3P^2\) 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 \(\Omega\) be the set of generated scenarios.

- **Solution of \(C^3P^2\) with sampled request rate distribution.** The second process in the SAA algorithm is to solve the sampled version of \(C^3P^2\), considering the scenarios that have been generated in the first process.

- **Verification of the solution feasibility.** Assume that \(\hat{x} \equiv [\hat{x}_{bc}, \forall b \in B, \forall c \in C]\) is an optimal solution for the sampled version of \(C^3P^2\). In the third process of the SAA algorithm, we verify the feasibility of \(\hat{x}\) to \(C^3P^2\). To do this, we first estimate the true probability function \(\Pr(\hat{x})\) (defined in (34)) by \(\hat{q}(\hat{x})\) (defined in (35)) using the set of scenarios \(\Omega\) generated in the first process.

\[
\text{(34)} \quad q(\hat{x}) \equiv \Pr\left\{ 2 t_{bc} x_{bc} + \frac{1}{\mu - \sum_{b \in B} \bar{r}_{h} x_{bc}} > \delta \right\},
\]

\[
\text{(35)} \quad \hat{q}(\hat{x}) \equiv \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \frac{1}{n} \left\{ 2 t_{bc} x_{bc} + \frac{1}{\mu - \sum_{b \in B} \bar{r}_{h} x_{bc}} \right\} > \delta.
\]

Next, following the method described in [45] and [48], we construct a \((1-\xi)\)-confidence upper bound on \(q(\hat{x})\), given by:

\[
\text{(36)} \quad U(\hat{x}) \equiv \hat{q} + z_{\xi} \sqrt{\frac{\hat{q} \left[ 1 - \hat{q} \right]}{\hat{\Omega}}},
\]

where \(\hat{\Omega}\) is a set of new scenarios generated for the verification of the feasibility of \(\hat{x}\), \(\hat{q}(\hat{x})\) is the estimated value of \(q(\hat{x})\) for the set of scenarios \(\hat{\Omega}\), \(z_{\xi} \equiv \Phi^{-1}(1-\xi)\), and \(\Phi(\cdot)\) is the cumulative distribution function (CDF) of the standard normal distribution. If \(U(\hat{x})\) is less than the risk level \((1-\beta)\), then \(\hat{x}\) is feasible with confidence level \((1-\xi)\).

- **Computation of the statistical lower and upper bounds.** If \(\hat{x}\) is a feasible solution to \(C^3P^2\) (with confidence level \((1-\xi)\)), then the corresponding objective function value constitutes a statistical upper bound on the optimal objective function value of \(C^3P^2\). To get a lower bound for the optimal objective function value, we take \(I\) iterations. For each iteration, we run the sampled version of \(C^3P^2\) with \(|\Omega|\) scenarios \(J\) times. For these \(J\) runs, we follow the same scheme as the one described in [45] to pick the \(Ith\) smallest optimal value. Specifically, \(l\) is the largest integer such that:

\[
B(l - 1; \theta_{\Omega}, \xi) \leq \xi, \tag{37}
\]

where \(B(l - 1; \theta_{\Omega}, \xi)\) is the CDF of the binomial distribution, given by (38), and \(\theta_{\Omega}\) is given by (39).

\[
\text{(38)} \quad B(l - 1; \theta_{\Omega}, J) \equiv \sum_{i=0}^{l-1} \binom{J}{i} (1 - \theta_{\Omega})^{J-i},
\]

\[
\theta_{\Omega} \equiv B\left(\left\lfloor (1 - \alpha) |\Omega| \right\rfloor ; 1 - \beta, |\Omega|\right). \tag{39}
\]

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

\section*{V. Robust Joint Controller Placement and Adaptive and RAN Node-Controller Assignment (CPPA)}

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

\subsection*{A. Problem Formulation}

Using two-stage stochastic programming [17], we formulate CPPA under the uncertainty of \(\tilde{r}_{bc}, \forall b \in B\). In contrast to \(C^3P^2\), in CPPA the RAN node-controller 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 controllers, knowing the distribution of \(\tilde{r}\). Our optimality criteria are: (i) minimizing the number of controllers and (ii) minimizing the response time to various RAN nodes, without decreasing it below \(\delta\). In contrast to \(C^3P^2\), CPPA does not ensure that the RAN node response time constraints are satisfied with a minimum probability of \(\beta\). The first-stage problem decision is static and is taken before knowing which realization of \(\tilde{r}\) will occur. In the second-stage problem, the RAN node-controller assignment is optimized under each realization of \(\tilde{r}\) aiming at minimizing the response

\footnote{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 C³P²

Result: Lower and upper statistical bounds on the optimal objective function value.

\[ i \leftarrow 1 \quad j \leftarrow 1 \]

while \( i \leq I \) do

while \( j \leq J \) do

a. Generate a new set \( \Omega \) of scenarios.
b. Solve the sampled version of C³P² with the new set \( \Omega \) of scenarios. Denote the solution by \( \hat{x}_j \) and the optimal value by \( \hat{o}_j \).
c. Generate a new set \( \hat{\Omega} \) of scenarios.
d. Estimate \( q(\hat{x}_j) \) in (34) using the set \( \hat{\Omega} \) of scenarios, as in (35). Denote the estimated value of \( q(\hat{x}_j) \) by \( \hat{q}(\hat{x}_j) \). Use (36) to compute \( U(\hat{x}_j) \).
e. If \( U(\hat{x}_j) \leq 1 - \beta \) then
   | Go to step e.
else
   | Continue to iteration \( j + 1 \).
end

\[ \hat{o} \overset{\text{def}}{=} \frac{1}{T} \sum_{i=1}^{T} \hat{o}_i \]
is a lower statistical bound for the optimal objective function value.

\[ a \overset{\text{def}}{=} \min_{1 \leq i \leq I} a_i \]
is an upper statistical bound for the optimal objective function value.

The optimality gap is estimated to be \( \frac{(a - \hat{o})}{\hat{o}} \times 100\% \).

time to various RAN nodes, without decreasing it below \( \delta \). Our two-stage stochastic optimization problem can be formulated as follows:

**Problem 2: CPPA**

\[
\begin{align*}
\text{minimize} & \quad \sum_{c \in C} y_c + \mathbb{E}[h(\mathbf{x}, \mathbf{r})] \\
\text{subject to:} & \quad y_c \in \{0, 1\}, \forall c \in C, \\
\end{align*}
\]

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

\[
\begin{align*}
\text{minimize} & \quad \sum_{b \in \mathcal{B}} \sum_{c \in C} q_{bc} x_{bc} \\
\text{max} & \quad 2 t_{bc} x_{bc} + \frac{1}{\mu - \sum_{b \in \mathcal{B}} r_b x_{bc}}, \delta)
\end{align*}
\]

subject to:

\[ y_c = \mathbb{I}\{\sum_{b \in \mathcal{B}} x_{bc} \geq 1\}, \forall c \in C, \]  \hspace{1cm} (43)

\[ \sum_{c \in C} x_{bc} = 1, \forall b \in \mathcal{B}, \] \hspace{1cm} (44)

\[ x_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in C, \] \hspace{1cm} (45)

where \( q_{bc}, b \in \mathcal{B}, c \in C \), are design coefficients introduced to balance the tradeoff between minimizing the number of controllers 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 \( \mathbb{E}[h(\mathbf{x}, \mathbf{r})] \). One standard technique for addressing this difficulty is sampling. The basic idea, as described in Section IV-B, is to approximate the true distribution of stochastic variables with an empirical distribution by sampling. We generate a set \( \Omega \) of i.i.d. samples (scenarios) from the distribution of the RAN node request rates using Monte Carlo simulation. After generating the scenarios, \( \mathbb{E}[h(\mathbf{x}, \mathbf{r})] \) can be estimated as \( \frac{1}{|\Omega|} \sum_{\omega \in \Omega} h(\mathbf{x}, \mathbf{r}^{\omega}) \),

where \( \mathbf{r}^{\omega} = [\mathbf{r}^{\omega}_b], \forall b \in \mathcal{B} \) is the vector of the RAN node request rates under scenario \( \omega \). CPPA with sampled request rate distribution is given by:

**CPPA with Sampled Request Rate Distribution**

\[
\begin{align*}
\text{minimize} & \quad \sum_{c \in C} y_c + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left( \sum_{b \in \mathcal{B}} \sum_{c \in C} q_{bc} x_{bc}^{(\omega)} \right) \\
\text{subject to:} & \quad y_c \in \{0, 1\}, \forall c \in C, \\
\end{align*}
\]

\[ x_{bc}^{(\omega)} = \mathbb{I}\{\sum_{b \in \mathcal{B}} x_{bc}^{(\omega)} \geq 1\}, \forall c \in C, \] \hspace{1cm} (47)

\[ \sum_{c \in C} x_{bc}^{(\omega)} = 1, \forall b \in \mathcal{B}, \forall \omega \in \Omega, \] \hspace{1cm} (48)

\[ x_{bc}^{(\omega)} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in C, \forall \omega \in \Omega, \] \hspace{1cm} (49)

\[ y_c \in \{0, 1\}, \forall c \in C. \] \hspace{1cm} (50)

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 [49], [50]. Specifically, let \( \mathbf{X}_c \) and \( \hat{\mathbf{X}}_c \) denote the sets of \( \epsilon \)-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 \( \Pr\{\hat{\mathbf{X}}_c \subset \mathbf{X}_c\} \geq 1 - 2\epsilon e^{-\zeta(\delta, \epsilon)|\Omega|} \).

In the next subsection, we use several linearization techniques to convert the sampled version of CPPA into a mixed integer linear program (MILPP), in order to solve it using CPLEX.
C. Mixed Integer Linear Reformulation

The max (·, ·) term in the second-stage problem objective function can be represented in a linear form by (i) introducing new positive decision variables, \( v_{bc} \), \( b, c \in C \), (ii) introducing new binary decision variables, \( f_{bc} \) (equals one if the response time over link \( bc \) is greater than \( \delta \), and equals zero otherwise) and \( d_{bc} \) (equals one if the response time over link \( bc \) is less than \( \delta \), and equals zero otherwise), \( \forall b \in B, \forall c \in C \), and (iii) adding the following constraints:

\[
\begin{align*}
v_{bc} & \geq \delta, \forall b \in B, \forall c \in C, \quad (51) \\
v_{bc} & \geq \frac{1}{\mu} v_{bc} \sum_{b \in B} \bar{r}_b \cdot x_{bc} + 2 t_{bc} \cdot x_{bc} + \frac{1}{\mu} \quad (52) \\
v_{bc} & \leq \frac{1}{\mu} v_{bc} \sum_{b \in B} \bar{r}_b \cdot x_{bc} + 2 t_{bc} \cdot x_{bc} + M (1 - f_{bc}) - \frac{M}{\mu} \left( \sum_{b \in B} \bar{r}_b \cdot x_{bc} - f_{bc} \sum_{b \in B} \bar{r}_b \cdot x_{bc} \right) \quad (53) \\
v_{bc} & \leq \delta + M (1 - f_{bc}), \forall b \in B, \forall c \in C, \quad (54) \\
f_{bc} + d_{bc} & = 1, \forall b \in B, \forall c \in C, \quad (55) \\
f_{bc}, d_{bc} & \in \{0, 1\}, \forall b \in B, \forall c \in C \quad (56)
\end{align*}
\]

where \( M \) is a sufficiently large number. The terms \( x_{bc} \) and \( f_{bc} \) can be linearized similar to (18) and (19). The term \( x_{bc} \), which represents a product of a binary variable with a positive continuous variable, can be reformulated by introducing the new decision variables \( e_{bc}, \forall b \in B, \forall c \in C \), and adding the following constraints:

\[
\begin{align*}
e_{bc} & \leq \max \left( 2 t_{bc} + \frac{1}{\mu - \sum_{b \in B} \bar{r}_b}, \delta \right) x_{bc} \quad (57) \\
e_{bc} & \geq \delta \cdot x_{bc} \\
e_{bc} & \geq v_{bc} - \max \left( 2 t_{bc} + \frac{1}{\mu - \sum_{b \in B} \bar{r}_b}, \delta \right) (1 - x_{bc}) \\
e_{bc} & \leq v_{bc} - \delta \cdot (1 - x_{bc}) \quad (58)
\end{align*}
\]

Therefore, the sampled version of CPPA can be equivalently written as an MILPP as follows:

\[
\text{MILPP Reformulation of the Sampled Version of CPPA}
\]

\[
\begin{align*}
\text{minimize} & \quad \left\{ \sum_{c \in C} y_c + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left( \sum_{b \in B} \sum_{c \in C} q_{bc} y_{bc}^{(\omega)} \right) \right\} \\
\text{subject to:} & \quad \begin{align*}
y_c & \geq 0, \forall c \in C, \\
n_{bc} & \geq 0, \forall b \in B, \forall c \in C, \\
e_{bc} & \geq 0, \forall b \in B, \forall c \in C, \\
e_{bc} & \geq x_{bc} \cdot \delta, \forall b \in B, \forall c \in C \\
e_{bc} & \geq (1 - x_{bc}) \cdot (1 - \delta), \forall b \in B, \forall c \in C \\
\end{align*}
\end{align*}
\]

Algorithm 2 SAA Algorithm for CPPA

\textbf{Result:} Lower and upper statistical bounds on the optimal objective function value.

\begin{algorithm}
\begin{algorithmic}
\State j \leftarrow 1
\While {j \leq J}
\State a \leftarrow \frac{1}{J} \sum_{j=1}^{J} \bar{o}_j \text{ is a lower statistical bound for the optimal objective function value.}
\State \bar{a} \leftarrow \min_{1 \leq j \leq J} \bar{o}_j \text{ is an upper statistical bound for the optimal objective function value.}
\State \text{The optimality gap is estimated to be } \frac{(a - \bar{a})}{\bar{a}} \times 100\%.
\EndWhile
\end{algorithmic}
\end{algorithm}

This MILPP formulation can be solved optimally using CPLEX.

D. 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 sample 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-D. If \( \bar{y} \) is a feasible solution to CPPA, then \( \sum_{c \in C} y_c + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} h(\bar{x}, r^{(\omega)}) \) constitutes a statistical upper bound on the optimal objective function value of CPPA, where \( \Omega \) 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\(^5\). Taking the average of the optimal values across the \( J \) iterations provides a lower bound for the optimal objective function value.

\(^5\)Every time, we pick a different set of \( |\Omega| \) scenarios.
time equals channel data rate was set to 100 Mbps (hence, the transmission of the flow setup request was set to 1500 k requests/second.

locations of the mobile users form a realization of a Poisson point process.

placement and RAN node-controller assignments schemes and core duo with RAN node delay dissatisfaction (averaged over all scenarios faction (averaged over all RAN nodes), and (iii) the average controllers, (ii) the average probability of RAN node satis-

optimizing the controller placement and the RAN node assignment.

In this section, we evaluate our stochastic joint controller

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

$$\sum_{b \in B} \frac{2 f_{bc}}{x_{bc}} + \frac{1}{\mu - \sum_{b \in B} \mathbb{E}[f_b]} x_{bc} \leq \delta, \quad (59)$$

with a sequential scheme in [12], in which the controller placement and the switch-controller assignment problems are solved separately. Specifically, in [12], a subset of switches is assigned first to every potential controller location, and then the controller placement problem is solved based on these precomputed sets of subsets. The QoS metric used in [12] is the average response time.

As shown in Figure 4, the joint scheme (i) reduces the number of controllers from three to two for most values of $\delta$, (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^3P^2 with a deterministic version of it, when we replace $\bar{r}_b$ with $\mathbb{E}[f_b]$ 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 contains 1000 users. Each scenario was generated as a non-stationary PPP from the SSLT field (as described in Section III-B) with $\omega_{\text{max}} = \pi/30$, $\sigma = 1$, $\gamma = 0$, and $L = 25$. The field is valid over the domain $x, y \in [0, 750] \text{ meters.}$. $\mathbb{E}[f_b] = [2629.4, 3957.8, 3360.8, 2824.8, 4544, 2591.6, 2806.8, 2635.4, 3039.8]$. $\Pr \{f_b < \mathbb{E}[f_b]\} = [0.5, 0.53, 0.5, 0.5, 0.48, 0.5, 0.45, 0.5, 0.5, 0.45]$. Since $0.45 \leq \Pr \{f_b < \mathbb{E}[f_b]\} \leq 0.55$, Figure 5 shows a comparable performance of the deterministic scheme to C^3P^2 with $\beta = 0.6$. However, when $\beta$ increases to 0.85, C^3P^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^3P^2) with the adaptive two-stage scheme (C^3P^2) using the same 100 scenarios used in Section VI-C. As can be seen from (58), $q = q_{\text{bc}}, \forall b \in B, \forall c \in C$, controls the tradeoff between the number of controllers and the RAN node delay satisfaction.

When $q$ is set to a sufficiently small value ($10^{-7}$, which makes $\mathbb{E}[h(x, \bar{r})] < 0.1$), Figures 6(a) and 7(a) show that two controllers are enough for most values of $\delta$, which is significantly less than that when $q = 1$, and comparable to the number of controllers of C^3P^2. As illustrated in Figure 6, when $q = 1$ CCPA shows a superior performance in RAN node satisfaction compared to C^3P^2 when $\beta = 0.6$, however when $\beta$ is sufficiently high ($\beta = 0.85$), C^3P2 shows a comparable RAN node satisfaction to CCPA. Finally, Figure 7 demonstrates the effect of $q$ on controlling the tradeoff between the number of controllers 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. Considering the setup explained in Section VI-A,

Fig. 3. Considered network topology in our performance evaluation. The locations of the mobile users form a realization of a Poisson point process.
Fig. 4. Comparison between sequential [12] and joint controller placement and RAN node-controller assignment.

Fig. 5. Comparison between deterministic and static stochastic controller placement and RAN node-controller assignment.

Fig. 6. Comparison between static (\(C^3P^2\)) and adaptive (CPPA) joint controller placement and RAN node-controller assignment.
in Table II we compare the results obtained from running the C^3P^2 with sampled request rate distribution ((20)–(33)) with the results obtained from running the SAA algorithm of C^3P^2. We considered different values of \( I, J, \vert \Omega \vert, \) and \( \vert \Omega \prime \vert \) in Table II. Table II shows that the SAA algorithm with \((I, J, \vert \Omega \vert) = (5, 5, 50)\) and \((7, 5, 50)\), as two examples, can achieve the same optimal solutions as the sampled version of C^3P^2 with \( \vert \Omega \vert = 5 \times 5 \times 50 = 1250 \) and \( 7 \times 5 \times 50 = 1750 \), respectively. Instead of running one MILPP for 1750 scenarios, using SAA we can run \( 7 \times 5 = 35 \) independent MILPPs\(^5\), each for 50 scenarios.

Table II motivates 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). The lower and upper statistical bounds along with the optimality gap are shown in Tables III and IV for different values of \( J \), \( \vert \Omega \vert \), and \( \vert \Omega \prime \vert \). They are also further illustrated as circular dendrograms in Figures 8 and 9, respectively.

As shown in Tables III and IV, increasing the number of considered scenarios (\( \vert \Omega \prime \vert \)), the number of scenarios used in the verification process (\( \vert \Omega \prime \vert \)), or \( J \) reduces the optimality gap. These tables show the ability of the SAA algorithm to estimate the optimal objective function value with a very low optimality gap.

### VII. Conclusions

Using stochastic programming, in this paper we studied the controller placement problem in software-defined cellular 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-RAN networks. We developed a static (C3P2) 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 C3P2, 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 $\beta$, whereas C3P2 ensures that. Using stochastic optimization and sample average approximation (SAA), combined with various linearization techniques, we extensively evaluated C3P2 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 C3P2 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.

REFERENCES


