Robust Controller Placement and Assignment in Software-defined Cellular Networks

Mohammad J. Abdel-Rahman, EmadelDin A. Mazied, Kory Teague, Allen B. MacKenzie, and Scott F. Midkiff
Wireless @ Virginia Tech, Department of Electrical and Computer Engineering, Virginia Tech, USA
{mo7ammad, emazied, koryt, mackenab, midkiff}@vt.edu

***Abstract*—Software-defined cellular networks (SDCN) have been recently introduced to enable flexible cellular network design that facilitates fulfilling 5G design requirements. Placement of controllers within the SDCN plays a crucial role in optimizing its performance. In this paper, we study the controller placement problem in SDCN, considering the uncertainty in cellular user locations. Specifically, our contributions are as follows. First, we develop $\mathbf{C^3P^2}$, a *static* joint stochastic controller placement and evolved node B (eNB)-controller assignment problem. The objective of $\mathbf{C^3P^2}$ is to minimize the number of controllers needed to control all eNBs, while ensuring that the response time to each eNB will exceed δ seconds with probability less than $1-\beta$. Second, we develop CPPA, a joint stochastic controller placement and *adaptive* eNB-controller assignment problem. In contrast to $\mathbf{C^3P^2}$, in CPPA the eNB-controller assignment adapts to variations in the eNB request rates, resulting from the variations in the cellular user locations. Finally, we use sample average approximation combined with various linearization techniques to solve and evaluate $\mathbf{C^3P^2}$ and CPPA under various system parameters. Our results demonstrate the advantages of (i) joint compared to sequential optimization, (ii) stochastic compared to deterministic optimization, and (iii) adaptive compared to static optimization.**

***Index Terms*—Software-defined cellular networks, controller placement problem, chance-constrained stochastic binary programming, two-stage stochastic binary programming, sample average approximation.**

I. Introduction

The drastic growth in the demand of mobile users introduced a big challenge for the cellular networks to fulfill the users' coverage and capacity quality of service (QoS) requirements. Mobility management, security, policy management, and access control are the most critical functions in the design of cellular networks [1]. In current cellular networks, e.g., LTE 4G, most of these functions are managed through expensive centralized data entity, i.e., packet gateway (P-GW). Inspired by its successful deployment in wired network, software-defined networking (SDN) principles have been incorporated in a diverse range of wireless and cellular network architectures, e.g., [1], [2].

SDN abstracts and centralizes the network control functions in a software entity that runs on a server, which is known as an SDN controller [3]. Recently, SDN has been extended to include cellular network control and management functionalities [1], [2]. The SDN controller is the key resource element in the network control plane design. In software-defined cellular networks (SDCN), SDN controllers interact with evolved node Bs (eNBs) to provide a robust and flexible network management framework [2].

As SDN controllers play a crucial role, they need to respond to their associated eNBs in a timely manner (within a few milliseconds). Optimal placement of SDN 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.

Channel uncertainty and *user mobility* are two key characteristics of wireless networks that impose unique challenges on the CPP in software-defined wireless networks (SDWN). 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 to the CPP. Specifically, the distribution of mobile users across eNBs will be stochastic, making the request rate from eNBs to the SDN 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 SDN controllers and their controlled elements nor account for users mobility. Recently, 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 elements. Stochastic programming provides a powerful mathematical tool to handle optimization under uncertainty. It has been recently exploited to optimize resource allocation in various types of wireless networks operating under uncertainties (examples include [19]–[24]).

Main contributions. In this paper, we develop the first mathematical framework for solving the CPP in SDCN, while considering the uncertainty in the geographical distribution of cellular users (and hence, eNB request rates). Specifically:

1. We develop a chance-constrained joint controller placement

and assignment scheme (denoted by C^{3}P^2): Using chance-constrained stochastic programming, we formulate a *static* joint stochastic controller placement and eNB-controller assignment problem. The objective of C^{3}P^2 is to minimize the number of controllers needed to control all eNBs, while ensuring that the response time to each eNB will exceed δ seconds with probability less than $1-\beta$.

2. We develop a two-stage joint controller placement and adaptive assignment scheme (denoted by CPPA): Using two-stage stochastic programming, we formulate a joint stochastic controller placement and *adaptive* eNB-controller assignment problem. In contrast to C^{3}P^2, where the eNB-controller assignment is static, in CPPA the eNB-controller assignment adapts to the variations in the eNB request rates, resulting from variations in cellular user locations.

   The goal of the CPPA first-stage is to optimally place the minimum number of controllers. Our optimality criteria are: (i) minimizing the number of controllers and (ii) minimizing the response time to various eNBs. In contrast to C^{3}P^2, CPPA does not ensure that the eNB 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 eNB request rates will occur. In the CPPA second-stage, the eNB-controller assignment is optimized under each realization of eNB request rates aiming at minimizing the response time to various eNBs.
3. To solve our stochastic programs, C^{3}P^2 and CPPA:
   - First, we use the sample average approximation (SAA) to convert C^{3}P^2 and CPPA into deterministic programs.
   - Then, using various linearization techniques, we derive the equivalent mixed integer linear formulations of the sampled C^{3}P^2 and CPPA problems.

Paper organization. The rest of the paper is organized as follows. In Section II, we describe our system model and state our problem. C^{3}P^2 and CPPA are developed in Sections III and IV, respectively. Our proposed approaches for solving C^{3}P^2 and CPPA are also described in Sections III and IV, respectively. We evaluate our schemes in Section V. Finally, we conclude the paper in Section VI.

II. System Model and Problem Statement

We consider a set $\mathcal{B} = \{1, 2, \ldots, B\}$ of eNBs forming a cellular network, and a set $\mathcal{C} = \{1, 2, \ldots, C\}$ of candidate locations for deploying SDN controllers to control the eNBs. A simplified view of our topology is depicted in Figure 1 for $B = 9$ and $C = 4$. The SDN controllers can be connected to the eNBs through wired or wireless links, as explained in [25]. In this paper, we consider wired links between the eNBs and the controllers. Multiple eNBs can be controlled by the same SDN controller. We assume that each cellular user has a request rate of k requests/second. The locations of the cellular users are time-varying and can be modeled as a stochastic process. Hence, at a given time instant, we model the request rate of eNB b, defined as the number of users served by eNB b at that time instant multiplied by k, as a stochastic variable, denoted by $\tilde{r}_b$.

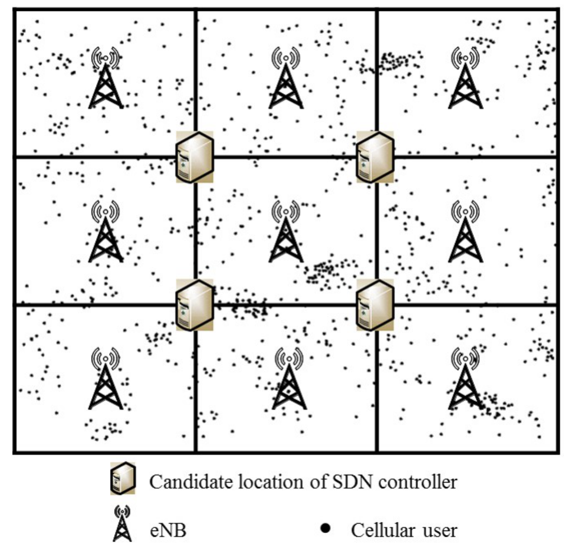


Fig. 1: Network topology ($B = 9$ and $C = 4$). The locations of the cellular users form a realization of a Poisson point process.

A. Distribution of eNB Request Rates

To know the distribution of $\tilde{r}_b, \forall b \in \mathcal{B}$, we want to know the distribution of the cellular user locations. In [26], 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 [27] 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 [27], 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 [27], each point represents a user, or a collection of users within a pixel surrounding the point, with the specified demand. It is assumed in [27] 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. 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:

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

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 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) is retained within the process only with probability $\lambda(x_i, y_i)/\lambda_{\max}$.

B. Problem Statement

Our objective in this paper is to find *the minimum number of SDN controllers, their optimal locations, and the optimal assignment of these controllers to the eNBs, where the optimality criteria are based on satisfying the eNBs 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 eNB b, and equals zero otherwise.

C. Queuing Delay at the SDN Controllers

In addition to the transmission (and propagation) delays, given by $2t_{bc}$ for the link between eNB b and controller c, an eNB will encounter a queuing delay at the controller. We model each controller as an M/M/1 queuing system [28], under which the mean service time (say, of controller c) can be expressed as [29]:

$$[D_{qc}] = \frac{1}{\mu - \sum_{b\in\mathcal{B}} \tilde{r}_b\, x_{bc}} \tag{3}$$

where μ is the controller service rate (a.k.a. controller processing capacity) and $\tilde{r}_b$ is the request rate of eNB b.

III. Static Joint Placement and Assignment

In this section, we consider the *static* joint stochastic SDN controller placement and eNB-controller assignment problem, referred to as C^{3}P^2.

A. Problem Formulation

Using chance-constrained stochastic programming [17], we formulate C^{3}P^2 under the uncertainty of $\tilde{r}_b, b \in \mathcal{B}$. The objective of C^{3}P^2 is to minimize the number of controllers needed to control all eNBs, while ensuring that the response time to each eNB will exceed δ seconds with probability less than $1-\beta$. C^{3}P^2 can be stated as follows:

Problem 1: C^{3}P^2

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

subject to:

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

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

$$x_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \tag{7}$$

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

B. Sample Average Approximation

Chance-constrained stochastic programs are largely intractable due to the difficulty in checking the feasibility of a particular solution [30]. In other words, for a given $x_{bc}, b \in \mathcal{B}, c \in \mathcal{C}$, computing $\Pr\left\{2\, t_{bc}\, x_{bc} + \frac{1}{\mu - \sum_{b\in\mathcal{B}} \tilde{r}_b\, x_{bc}} \leq \delta\right\}$ accurately is hard. One standard technique for addressing this difficulty in solving chance-constrained stochastic programs is SAA.

The basic idea of SAA is to approximate the true distribution of stochastic variables with an empirical distribution by sampling. We generate a set Ω of independent and identically distributed (i.i.d.) samples (scenarios) from the distribution of the eNB request rates, described in Section II-A. After generating the scenarios, the chance constraint can be estimated using an indicator function as: $|\Omega|^{-1} \sum_{\omega\in\Omega} {}_{\left\{2\, t_{bc}\, x_{bc} + \frac{1}{\mu - \sum_{b\in\mathcal{B}} r_b^{(\omega)}\, x_{bc}} \leq \delta\right\}} \geq \beta$, where $r_b^{(\omega)}$ is the request rate of eNB b under scenario ω. The SAA problem of C^{3}P^2 is given by:

C^{3}P^2 (SAA)

$$\underset{\{x_{bc}, b\in\mathcal{B}, c\in\mathcal{C}\}}{\text{minimize}} \sum_{c\in\mathcal{C}} {}_{\{\sum_{b\in\mathcal{B}} x_{bc}\geq 1\}} \tag{8}$$

subject to:

$$\sum_{c\in\mathcal{C}} x_{bc} = 1, \forall b \in \mathcal{B} \tag{9}$$

$$\sum_{\omega\in\Omega} {}_{\left\{2\, t_{bc}\, x_{bc} + \frac{1}{\mu - \sum_{b\in\mathcal{B}} r_b^{(\omega)}\, x_{bc}} \leq \delta\right\}} \geq \alpha\, |\Omega|, \quad \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \tag{10}$$

$$x_{bc} \in \{0, 1\}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \tag{11}$$

where $\alpha \in (0, 1]$ and it can be different from β.

The advantages of solving the SAA problem are:

- **Lower bound on the required number of controllers.** Specifically, if o^*_β and $\hat{o}_\alpha$ are the optimal objective function values (i.e., minimum number of controllers) of the C^{3}P^2 and C^{3}P^2 (SAA) problems, respectively, then, it has been shown that $\hat{o}_\alpha \leq o^*_\beta$ with probability at least $1-\eta$ if $|\Omega| \geq \frac{1}{2(\beta-\alpha)^2}\ln\left(\frac{1}{\eta}\right)$ and $\alpha < \beta$ [31].

- **Feasible solution to C^{3}P^2.** Specifically, it has been shown that for $\alpha > \beta$, every feasible solution to the SAA problem will be feasible to C^{3}P^2 with probability at least $1-\eta$ if $|\Omega| \geq \frac{1}{2(\alpha-\beta)^2}\ln\left(\frac{|X\setminus X_\beta|}{\eta}\right)$, where $X \setminus X_\beta = \Big\{x_{bc}, b \in \mathcal{B}, c \in \mathcal{C} : \Pr\Big\{2\,t_{bc}\,x_{bc} + \frac{1}{\mu - \sum_{b\in\mathcal{B}}\tilde{r}_b\,x_{bc}} \leq \delta\Big\} < \beta\Big\}$ [31]. Therefore, we do not need to solve the SAA problem to optimality to obtain a solution to C^{3}P^2. Simple heuristics can be enough to obtain a solution that is feasible to C^{3}P^2 with high probability. Design of efficient heuristics for solving C^{3}P^2 (SAA) is left for future research.
- **Optimal solution to C^{3}P^2.** Let $\alpha_{\min} = \min\Big\{\Pr\Big\{2\,t_{bc}\,x_{bc} + \frac{1}{\mu-\sum_{b\in\mathcal{B}}\tilde{r}_b\,x_{bc}} \leq \delta\Big\} : x_{bc}, b \in \mathcal{B}, c \in \mathcal{C}, \text{ is an optimal solution to C}^3\text{P}^2\Big\}$, $\alpha_{\max} = \max\Big\{\Pr\Big\{2\,t_{bc}\,x_{bc} + \frac{1}{\mu-\sum_{b\in\mathcal{B}}\tilde{r}_b\,x_{bc}} \leq \delta\Big\} : x_{bc}, b \in \mathcal{B}, c \in \mathcal{C}, \text{ is not a feasible solution to C}^3\text{P}^2\Big\}$, and $\kappa = \min\{\alpha_{\min}-\beta, \beta-\alpha_{\max}\}$. Then, if $\alpha_{\min} > \beta$, $\Pr\{\hat{o}_\beta = o^*_\beta\} \geq 1 - \left(2^{BC}+1\right)e^{-2\kappa^2|\Omega|}$ [31]. Hence, solving C^{3}P^2 (SAA) with $\alpha = \beta$ yields an exact optimal solution to C^{3}P^2 with probability approaching one exponentially fast with $|\Omega|$.

C. Mixed Integer Linear Reformulation

In this subsection, we derive an equivalent mixed integer linear programing (MILP) formulation of C^{3}P^2 (SAA), in order to solve it using CPLEX [32].

First, note that the objective function of C^{3}P^2 (SAA) is non-linear. It can be represented in a linear form by introducing new binary decision variables, $y_c \stackrel{\text{def}}{=} {}_{\{\sum_{b\in\mathcal{B}} x_{bc} \geq 1\}}, \forall c \in \mathcal{C}$, and reformulating the indicator function as follows [33]:

- If $\sum_{b\in\mathcal{B}} x_{bc} \geq 1$ then $y_c = 1$ can be reformulated as:
$$\sum_{b\in\mathcal{B}} x_{bc} - (M+\epsilon)\,y_c \leq 1-\epsilon \tag{12}$$
where M is an upper bound of $\sum_{b\in\mathcal{B}} x_{bc} - 1$ and $\epsilon > 0$ 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, (12) reduces to $\sum_{b\in\mathcal{B}} x_{bc} \leq B\,y_c$.
- If $y_c = 1$ then $\sum_{b\in\mathcal{B}} x_{bc} \geq 1$ can be reformulated as[1]:
$$\sum_{b\in\mathcal{B}} x_{bc} + m\,y_c \geq m+1 \tag{13}$$
where m is a lower bound of $\sum_{b\in\mathcal{B}} x_{bc} - 1$. Selecting m to be -1, (13) reduces to $\sum_{b\in\mathcal{B}} x_{bc} \geq y_c$.

Therefore,
$$y_c = {}_{\{\sum_{b\in\mathcal{B}} x_{bc} \geq 1\}} \iff y_c \leq \sum_{b\in\mathcal{B}} x_{bc} \leq B\,y_c, \forall c \in \mathcal{C}.$$

To reformulate (10), we introduce a binary variable $u^{(\omega)}_{bc}$ for each link between eNB $b \in \mathcal{B}$ and SDN controller $c \in \mathcal{C}$, and each scenario $\omega \in \Omega$. $u^{(\omega)}_{bc} = 0$ if the response time of controller c to eNB b under scenario ω is less than δ, and $u^{(\omega)}_{bc} = 1$ otherwise. Then, (10) is equivalent to the following constraints:
$$2\,t_{bc}\,x_{bc} + \frac{1}{\mu - \sum_{b\in\mathcal{B}} r^{(\omega)}_b\,x_{bc}} - \delta + \epsilon \leq N_{bc}\,u^{(\omega)}_{bc}, \quad \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega \tag{14}$$
$$\sum_{\omega\in\Omega}\left(1 - u^{(\omega)}_{bc}\right) \geq \alpha\,|\Omega|, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \tag{15}$$
where $N_{bc} = \left(2\,t_{bc} + \frac{1}{\mu - \sum_{b\in\mathcal{B}} r^{(\omega)}_b} - \delta + \epsilon\right)$ is an upper-bound of the left-hand-side of (14) and $\epsilon > 0$ is a small tolerance beyond which we regard the constraint as having been broken.

Constraint (14) is non-linear. It can be equivalently written as:
$$\begin{aligned}2\,\mu\,t_{bc}\,x_{bc} \;-\; 2\,t_{bc}\sum_{b_1\in\mathcal{B}} r^{(\omega)}_{b_1}\,x_{b_1c}\,x_{bc} - \mu\,N_{bc}\,u^{(\omega)}_{bc} \\ + N_{bc}\sum_{b_1\in\mathcal{B}} r^{(\omega)}_{b_1}\,x_{b_1c}\,u^{(\omega)}_{bc} + (\delta-\epsilon)\sum_{b\in\mathcal{B}} r^{(\omega)}_b\,x_{bc} \\ \leq \mu\,(\delta-\epsilon) - 1, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega.\end{aligned} \tag{16}$$

Equation (16) includes the non-linear terms $x_{b_1c}\,x_{bc}$ and $x_{b_1c}\,u^{(\omega)}_{bc}$. It can be equivalently expressed in a linear form as follows:
$$\begin{aligned}2\,\mu\,t_{bc}\,x_{bc} - 2\,t_{bc}\sum_{b_1\in\mathcal{B}} r^{(\omega)}_{b_1}\,x_{bb_1c} - \mu\,N_{bc}\,u^{(\omega)}_{bc} \\ + N_{bc}\sum_{b_1\in\mathcal{B}} r^{(\omega)}_{b_1}\,z^{(\omega)}_{bb_1c} + (\delta-\epsilon)\sum_{b\in\mathcal{B}} r^{(\omega)}_b\,x_{bc} \\ \leq \mu\,(\delta-\epsilon) - 1, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega.\end{aligned} \tag{17}$$
after introducing the new decision variables x_{bb_1c} and $z^{(\omega)}_{bb_1c}$, $\forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega$, and adding the following constraints:
$$\begin{aligned}&x_{bb_1c} \leq x_{bc}, \forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}\\ &x_{bb_1c} \leq x_{b_1c}, \forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}\\ &x_{bb_1c} \geq x_{bc} + x_{b_1c} - 1, \forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}\\ &x_{bb_1c} \geq 0, \forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}.\end{aligned} \tag{18}$$
$$\begin{aligned}&z^{(\omega)}_{bb_1c} \leq u^{(\omega)}_{bc}, \forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega\\ &z^{(\omega)}_{bb_1c} \leq x_{b_1c}, \forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega\\ &z^{(\omega)}_{bb_1c} \geq u^{(\omega)}_{bc} + x_{b_1c} - 1, \forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega\\ &z^{(\omega)}_{bb_1c} \geq 0, \forall b, b_1 \in \mathcal{B}, \forall c \in \mathcal{C}, \forall \omega \in \Omega.\end{aligned} \tag{19}$$

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

Therefore, C^3P^2 (SAA) can be equivalently written as an MILP as follows:

Mixed Integer Linear Reformulation of C^3P^2 (SAA)

$$\underset{\substack{\left\{\substack{x_{bc},y_c,x_{bb_1c},\\ z_{bb_1c}^{(\omega)},u_{bc}^{(\omega)},\\ b,b_1\in\mathcal{B},c\in\mathcal{C},\omega\in\Omega}\right\}}}{\text{minimize}} \sum_{c\in\mathcal{C}} y_c \tag{20}$$

subject to:

$$\sum_{c\in\mathcal{C}} x_{bc} = 1, \forall b\in\mathcal{B} \tag{21}$$

$$\sum_{b\in\mathcal{B}} x_{bc} \le B\, y_c, \forall c\in\mathcal{C} \tag{22}$$

$$\begin{aligned}&2\,\mu\, t_{bc}\, x_{bc} - 2\, t_{bc} \sum_{b_1\in\mathcal{B}} r_{b_1}^{(\omega)}\, x_{bb_1c} - \mu\, N_{bc}\, u_{bc}^{(\omega)}\\ &\quad + N_{bc} \sum_{b_1\in\mathcal{B}} r_{b_1}^{(\omega)}\, z_{bb_1c}^{(\omega)} + (\delta-\epsilon)\sum_{b\in\mathcal{B}} r_b^{(\omega)}\, x_{bc}\\ &\qquad \le \mu\,(\delta-\epsilon) - 1, \forall b\in\mathcal{B}, \forall c\in\mathcal{C}, \forall\omega\in\Omega\end{aligned} \tag{23}$$

$$\sum_{\omega\in\Omega}\left(1-u_{bc}^{(\omega)}\right) \ge \alpha\,|\Omega|, \forall b\in\mathcal{B}, \forall c\in\mathcal{C} \tag{24}$$

$$x_{bb_1c} \le x_{bc}, \forall b,b_1\in\mathcal{B}, \forall c\in\mathcal{C} \tag{25}$$

$$x_{bb_1c} \le x_{b_1c}, \forall b,b_1\in\mathcal{B}, \forall c\in\mathcal{C} \tag{26}$$

$$x_{bb_1c} \ge x_{bc} + x_{b_1c} - 1, \forall b,b_1\in\mathcal{B}, \forall c\in\mathcal{C} \tag{27}$$

$$z_{bb_1c}^{(\omega)} \le u_{bc}^{(\omega)}, \forall b,b_1\in\mathcal{B}, \forall c\in\mathcal{C}, \forall\omega\in\Omega \tag{28}$$

$$z_{bb_1c}^{(\omega)} \le x_{b_1c}, \forall b,b_1\in\mathcal{B}, \forall c\in\mathcal{C}, \forall\omega\in\Omega \tag{29}$$

$$z_{bb_1c}^{(\omega)} \ge u_{bc}^{(\omega)} + x_{b_1c} - 1, \forall b,b_1\in\mathcal{B}, \forall c\in\mathcal{C}, \forall\omega\in\Omega \tag{30}$$

$$x_{bc}, y_c \in\{0,1\}, \forall b\in\mathcal{B}, \forall c\in\mathcal{C} \tag{31}$$

$$u_{bc}^{(\omega)} \in\{0,1\}, \forall b\in\mathcal{B}, \forall c\in\mathcal{C}, \forall\omega\in\Omega \tag{32}$$

$$x_{bb_1c}, z_{bb_1c}^{(\omega)} \ge 0, \forall b,b_1\in\mathcal{B}, \forall c\in\mathcal{C}, \forall\omega\in\Omega. \tag{33}$$

IV. Joint Placement and Adaptive Assignment

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

A. Problem Formulation

Using two-stage stochastic programming [17], we formulate CPPA under the uncertainty of $\tilde{r}_b, \forall b\in\mathcal{B}$. In contrast to C^3P^2, in CPPA the eNB-controller assignment adapts to the variations in the eNB request rates.

The goal of the first-stage problem is to optimally place the minimum number of controllers, knowing the distribution of $\tilde{r}_b, b\in\mathcal{B}$. Our optimality criteria are: (i) minimizing the number of controllers and (ii) minimizing the response time to various eNBs, without decreasing it below δ. In contrast to C^3P^2, CPPA does not ensure that the eNB 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 $\tilde{r}_b, b\in\mathcal{B}$, will occur. In the second-stage problem, the eNB-controller assignment is optimized under each realization of $\tilde{r}_b, b\in\mathcal{B}$, aiming at minimizing the response time to various eNBs, without decreasing it below δ. Our two-stage stochastic optimization problem can be formulated as follows:

Problem 2: CPPA

$$\underset{\{y_c,c\in\mathcal{C}\}}{\text{minimize}} \left\{\sum_{c\in\mathcal{C}} y_c + \quad [h(\boldsymbol{x},\tilde{\boldsymbol{r}})]\right\} \tag{34}$$

subject to:

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

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

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

subject to:

$$y_c = \quad_{\left\{\sum_{b\in\mathcal{B}} x_{bc}\ge 1\right\}}, \forall c\in\mathcal{C} \tag{37}$$

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

$$x_{bc}\in\{0,1\}, \forall b\in\mathcal{B}, \forall c\in\mathcal{C} \tag{39}$$

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

B. Sample Average Approximation

A key source of difficulty in solving two-stage stochastic programs is in evaluating $\quad [h(\boldsymbol{x},\tilde{\boldsymbol{r}})]$. One standard technique for addressing this difficulty is to replace $\quad [h(\boldsymbol{x},\tilde{\boldsymbol{r}})]$ by an SAA and solve the corresponding optimization problem (denote this problem by CPPA (SAA)).

It has been shown that a solution to CPPA (SAA) is an optimal solution to CPPA with probability approaching one exponentially fast as $|\Omega|$ increases [34], [35]. Specifically, let X_ϵ and $\hat{X}_\epsilon$ denote the sets of ϵ-optimal solutions of CPPA and CPPA (SAA), respectively. Then, for any $\epsilon>0$ and $\delta\in[0,\epsilon]$, there exists a constant $\zeta(\delta,\epsilon)\ge 0$ such that $\Pr\left\{\hat{X}_\delta\subset X_\epsilon\right\}\ge 1-2^C e^{-\zeta(\delta,\epsilon)|\Omega|}$.

C. Mixed Integer Linear Reformulation

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, $v_{bc} \overset{\text{def}}{=} \max\left(2\, t_{bc}\, x_{bc} + \frac{1}{\mu-\sum_{b\in\mathcal{B}}\tilde{r}_b\, x_{bc}}, \delta\right), \forall b\in\mathcal{B}, \forall c\in\mathcal{C}$, (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:

$$v_{bc} \geq \delta, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \tag{40}$$

$$\begin{aligned} v_{bc} \geq \frac{1}{\mu} \sum_{b_1 \in \mathcal{B}} \tilde{r}_{b_1}\; x_{b_1 c}\; \tilde{v}_{bc} + 2\, t_{bc}\; x_{bc} + \frac{1}{\mu} \\ - \frac{2}{\mu}\, t_{bc} \sum_{b_1 \in \mathcal{B}} \tilde{r}_{b_1}\; x_{b_1 c}\; x_{bc}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \end{aligned} \tag{41}$$

$$\begin{aligned} v_{bc} \leq \frac{1}{\mu} \sum_{b_1 \in \mathcal{B}} \tilde{r}_{b_1}\; x_{b_1 c}\; \tilde{v}_{bc} + 2\, t_{bc}\; x_{bc} + M\,(1 - f_{bc}) \\ - \frac{M}{\mu} \left(\sum_{b \in \mathcal{B}} \tilde{r}_b\; x_{bc} - \sum_{b_1 \in \mathcal{B}} \tilde{r}_{b_1}\; x_{b_1 c}\; f_{bc} \right) \\ - \frac{2}{\mu}\, t_{bc} \sum_{b_1 \in \mathcal{B}} \tilde{r}_{b_1}\; x_{b_1 c}\; x_{bc} + \frac{1}{\mu}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \end{aligned} \tag{42}$$

$$v_{bc} \leq \delta + M\,(1 - d_{bc})\,, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \tag{43}$$

$$f_{bc} + d_{bc} = 1, \forall b \in \mathcal{B}, \forall c \in \mathcal{C} \tag{44}$$

where M is a sufficiently large number. The terms $x_{b_1 c}\; x_{bc} \stackrel{\text{def}}{=} x_{bb_1 c}$ and $x_{b_1 c}\; f_{bc} \stackrel{\text{def}}{=} f_{bb_1 c}$ can be linearized similar to (18) and (19). The term $x_{b_1 c}\; \tilde{v}_{bc}$, which represents a product of a binary variable with a positive continuous variable, can be reformulated by introducing the new positive decision variables $e_{bb_1 c}, \forall b, b_1 \in \mathcal{B}, c \in \mathcal{C}$, and adding the following constraints:

$$\begin{aligned} & e_{bb_1 c} \leq \max\left(2\, t_{bc} + \frac{1}{\mu - \sum_{b \in \mathcal{B}} \tilde{r}_b}, \delta \right) x_{b_1 c} \\ & e_{bb_1 c} \geq \delta\; x_{b_1 c} \\ & e_{bb_1 c} \geq v_{bc} - \max\left(2\, t_{bc} + \frac{1}{\mu - \sum_{b \in \mathcal{B}} \tilde{r}_b}, \delta \right) (1 - x_{b_1 c}) \\ & e_{bb_1 c} \leq v_{bc} - \delta\,(1 - x_{bc})\,. \end{aligned} \tag{45}$$

Therefore, CPPA (SAA) can be equivalently written as an MILP as follows:

Mixed Integer Linear Reformulation of CPPA (SAA)

$$\underset{\left\{\substack{y_c, x_{bc}^{(\omega)}, \\ v_{bc}^{(\omega)}, f_{bc}^{(\omega)}, \\ d_{bc}^{(\omega)}, x_{bbc}^{(\omega)}, \\ f_{bbc}^{(\omega)}, e_{bbc}^{(\omega)}, \\ b \in \mathcal{B}, c \in \mathcal{C}, \\ \omega \in \Omega}\right\}}{\text{minimize}} \left\{ \sum_{c \in \mathcal{C}} y_c + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left(\sum_{b \in \mathcal{B}} \sum_{c \in \mathcal{C}} q_{bc}\; v_{bc}^{(\omega)} \right) \right\} \tag{46}$$

subject to:

(21), (22), (40) – (45), $\forall \omega \in \Omega$

linearization constraints of $x_{bb_1 c}^{(\omega)}$ and $f_{bb_1 c}^{(\omega)}$, similar to (18) and (19).

V. Performance Evaluation

In this section, we evaluate our stochastic joint controller placement and eNB-controller assignment schemes, and compare with [12].

A. Evaluation Setup

We used the grid topology shown in Figure 1 with $B = 9$ and $C = 4$. We considered 100 i.i.d. scenarios, i.e., realizations of cellular user locations (and hence, eNB request rates), each contains 1000 users. Each scenario was generated as a non-stationary PPP from the SSLT field (as described in Section II-A) with $\omega_{\max} = \pi/30$, $\sigma = 1$, $\gamma = 0$, and $L = 25$. The field is valid over the domain $x, y \in [0, 500]$ meters. $[\tilde{r}_b] = [2629.4,\ 3957.8,\ 3360.8,\ 2824.8,\ 4544,\ 2591.6,\ 2806.8,\ 2635.4,\ 3039.8]$. $\Pr\{\tilde{r}_b < \ [\tilde{r}_b]\} = [0.51, 0.53, 0.48, 0.55, 0.52, 0.51, 0.55, 0.51, 0.45]$. The controller processing capacity (μ) was set to 20000 requests/second. k in Section II was set to 0.2. The packet size of the flow setup request was set to 1500 Bytes [36] and the channel data rate was set to 25 Mbps (hence, the transmission time equals $\frac{1500 \times 8}{25 \times 10^6} = 0.48$ milliseconds). α in (10) equals β in (6). We ran our experiments on an Intel core i5 3.3 GHz core duo with 16 GB RAM. We used CPLEX [32] to solve our optimization problems.

Our performance evaluation metrics are: (i) the number of controllers, (ii) average probability of eNB satisfaction (averaged over all eNBs), and (iii) average eNB delay dissatisfaction (averaged over all scenarios and all eNBs). We studied the effects of δ, β, and $q_{bc} \stackrel{\text{def}}{=} q, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}$, on these metrics.

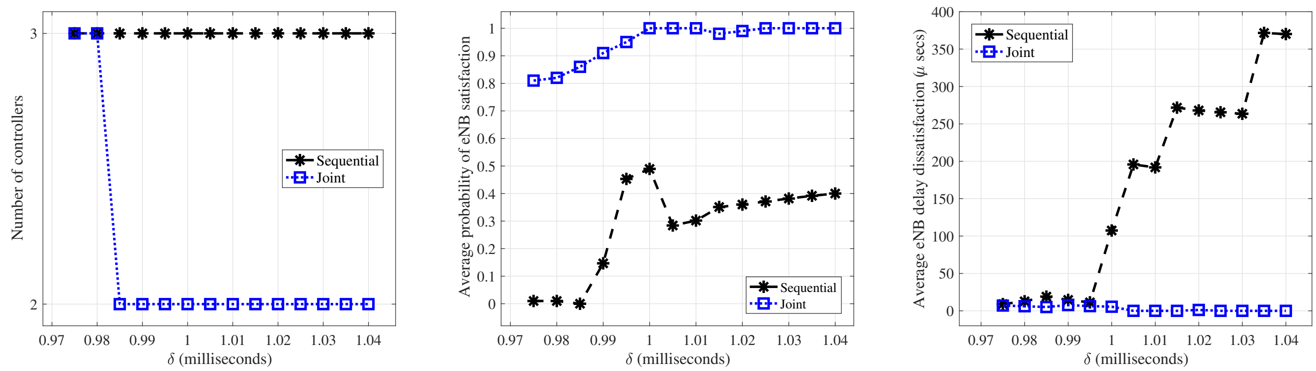
B. Sequential vs. Joint

In this subsection, we demonstrate the advantages of jointly optimizing the controller placement and the eNB-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:

$$\frac{\sum_{b \in \mathcal{B}} 2\, t_{bc}\; x_{bc}}{\sum_{b \in \mathcal{B}} x_{bc}} + \frac{1}{\mu - \sum_{b \in \mathcal{B}} \ [\tilde{r}_b]\; x_{bc}} \leq \delta, \tag{47}$$

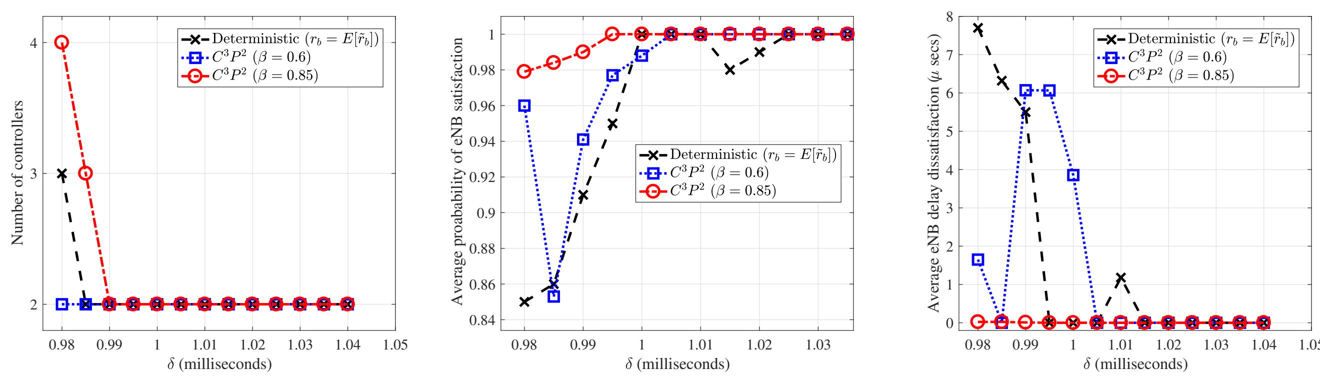
with a sequential scheme in [12], in which the controller placement and the switch-controller assignment problems are solved separately. Specifically, in [12], a subset of switches is assigned first to every potential controller location and then the controller placement problem is solved based on these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

As shown in Figure 2, the joint scheme (i) reduces the number of controllers from three to two for most values of δ, (ii) increases the average probability of eNB satisfaction by at least 50% (when $\delta = 1$ millisecond), and (iii) brings the level of average eNB delay dissatisfaction close to 0.



(a) Number of controllers (b) Average probability of eNB satisfaction (c) Average eNB delay dissatisfaction

Fig. 2: Comparison between sequential [12] and joint controller placement and eNB-controller assignment.



(a) Number of controllers (b) Average probability of eNB satisfaction (c) Average eNB delay dissatisfaction

Fig. 3: Comparison between deterministic and static stochastic controller placement and eNB-controller assignment.

C. *Deterministic vs. Stochastic*

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 $[\tilde{r}_b]$ and remove the probability term in (6).

Since $0.45 \leq \Pr\{\tilde{r}_b < [\tilde{r}_b]\} \leq 0.55$, Figure 3 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 eNB satisfaction significantly and brings the average eNB delay dissatisfaction level close to 0.

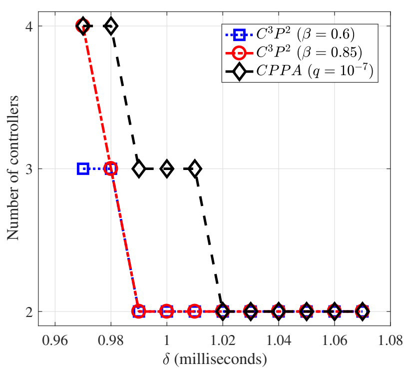
D. *Static vs. Adaptive*

In this subsection, we compare our static single-stage scheme (C^{3}P^2) with the adaptive two-stage scheme (CPPA). As can be seen from (46), $q \overset{\text{def}}{=} q_{bc}, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}$, controls the tradeoff between the number of controllers and the eNBs delay satisfaction.

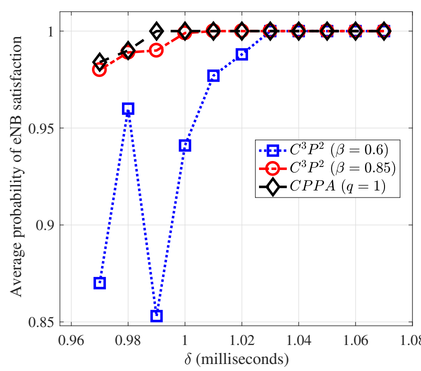
When q is set to a sufficiently small value (10^{-7}, which makes $[h(\boldsymbol{x}, \tilde{\boldsymbol{r}})] < 0.1$), Figures 4(a) and 5(a) show that two controllers are enough for most values of δ, which is significantly less than that when $q = 1$, and comparable to the number of controllers of C^{3}P^2. As illustrated in Figure 4, when $q = 1$ CCPA shows a superior performance in eNB satisfaction compared to C^{3}P^2 when $\beta = 0.6$, however when β is sufficiently high ($\beta = 0.85$), C^{3}P^2 shows a comparable eNB satisfaction to CPPA. Finally, Figure 5 demonstrates the effect of q on controlling the tradeoff between the number of controllers and the eNB delay satisfaction.

VI. Conclusions

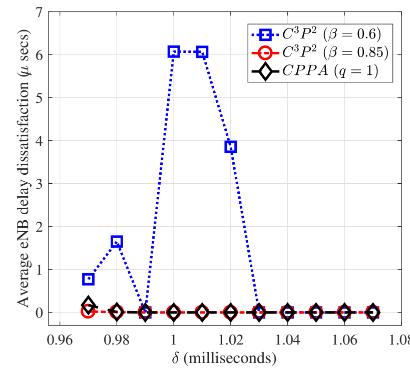
Using stochastic programming, in this paper we studied the controller placement problem in software-defined cellular networks, considering the uncertainty in the cellular user locations. We developed a static (C^{3}P^2) and an adaptive (CPPA) joint stochastic controller placement and eNB-controller assignment problems. Our optimization criteria are: (i) minimizing the number of controllers and (ii) minimizing the response time to various eNBs. In contrast to C^{3}P^2, in CPPA the eNB-controller assignment adapts to the variations in the cellular user locations. However, CPPA does not ensure that the eNB response time constraints are satisfied with a minimum probability of β, whereas C^{3}P^2 ensures that. We extensively evaluated C^{3}P^2 and CPPA. Our results demonstrated



(a) Number of controllers

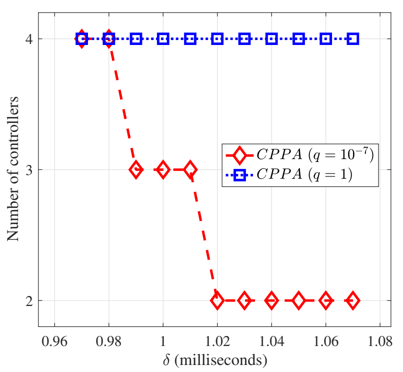


(b) Average probability of eNB satisfaction

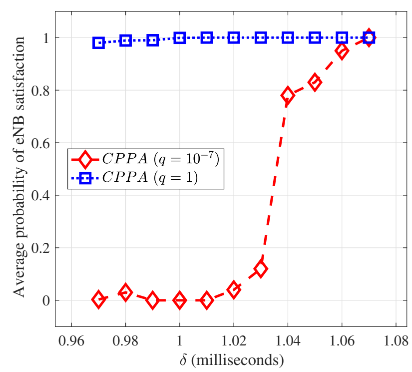


(c) Average eNB delay dissatisfaction

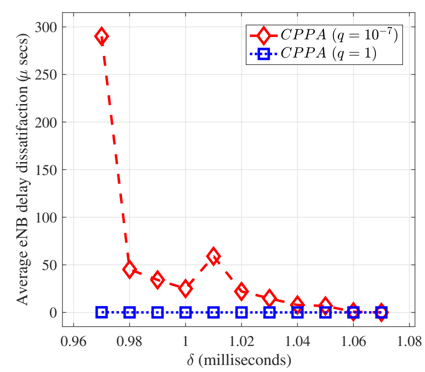
Fig. 4: Comparison between static (C^3P^2) and adaptive (CPPA) joint controller placement and eNB-controller assignment.



(a) Number of controllers



(b) Average probability of eNB satisfaction



(c) Average eNB delay dissatisfaction

Fig. 5: Effect of $q_{bc} \stackrel{\text{def}}{=} q, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}$, on the joint controller placement and adaptive eNB-controller assignment (CPPA).

the advantages of (i) joint compared to sequential optimization, (ii) stochastic compared to deterministic optimization, and (iii) adaptive compared to static optimization.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1443978.

References

[1] I. F. Akyildiz, S.-C. Lin, and P. Wang, "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation," *Computer Networks Journal*, vol. 93, Part 1, pp. 66–79, December 2015.

[2] L. E. Li, Z. M. Mao, and J. Rexford, "Toward software-defined cellular networks," in *Proceedings of the European Workshop on Software Defined Networking*, October 2012, pp. 7–12.

[3] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *SIGCOMM Computer Communications Review*, vol. 44, no. 2, pp. 87–98, April 2014.

[4] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," *SIGCOMM Computer Communications Review*, vol. 42, no. 4, September 2012.

[5] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, May 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 Communications Letters*, vol. 18, no. 8, pp. 1339–1342, August 2014.

[7] S. Liu, H. Wang, S. Yi, and F. Zhu, "NCPSO: A solution of the controller placement problem in software defined networks," in *Proceedings of the ICA3PP Conference Part III*, November 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 *Proceedings of the IEEE ITC Conference*, September 2013, pp. 1–9.

[9] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 4–17, March 2015.

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

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

[12] T. Y. Cheng, M. Wang, and X. Jia, "QoS-Guaranteed controller placement in SDN," in *Proceedings of the IEEE GLOBECOM Conference*, December 2015, pp. 1–6.

[13] T. Han, Y. Han, X. Ge, Q. Li, J. Zhang, Z. Bai, and L. Wang, "Small cell offloading through cooperative communication in software-defined heterogeneous networks," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7381–7392, October 2016.

[14] S. Auroux and H. Karl, "Flow processing-aware controller placement in wireless DenseNets," in *Proceedings of the IEEE PIMRC Conference*, September 2014, pp. 1294–1299.

[15] I. F. Akyildiz, P. Wang, and S.-C. Lin, "SoftAir: A software defined networking architecture for 5G wireless systems," *Computer Networks Journal*, vol. 85, pp. 1–18, July 2015.

[16] A. Ksentini, M. Bagaa, and T. Taleb, "On using SDN in 5G: the controller placement problem," in *Proceedings of the IEEE GLOBECOM Conference*, December 2016, pp. 1–6.

[17] P. Kall and S. W. Wallace, *Stochastic Programming*. John Wiley and Sons, 1994.

[18] M. J. Abdel-Rahman, E. A. Mazied, A. B. MacKenzie, S. F. Midkiff, M. R. M. Rizk, and M. Y. ElNainay, "On stochastic controller placement in software-defined wireless networks," in *to appear in the Proceedings of the IEEE WCNC Conference*, March 2017.

[19] M. J. Abdel-Rahman and M. Krunz, "Stochastic guard-band-aware channel assignment with bonding and aggregation for DSA networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 7, pp. 3888–3898, July 2015.

[20] N. Y. Soltani, S. J. Kim, and G. B. Giannakis, "Chance-constrained optimization of OFDMA cognitive radio uplinks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1098–1107, March 2013.

[21] M. J. Abdel-Rahman, K. Cardoso, A. B. MacKenzie, and L. A. DaSilva, "Dimensioning virtualized wireless access networks from a common pool of resources," in *Proceedings of the IEEE CCNC Conference*, January 2016, pp. 1049–1054.

[22] M. J. Abdel-Rahman, M. AbdelRaheem, A. B. MacKenzie, K. Cardoso, and M. Krunz, "On the orchestration of robust virtual LTE-U networks from hybrid half/full-duplex Wi-Fi APs," in *Proceedings of the IEEE WCNC Conference*, April 2016.

[23] M. J. Abdel-Rahman, M. AbdelRaheem, and A. B. MacKenzie, "Stochastic resource allocation in opportunistic LTE-A networks with heterogeneous self-interference cancellation capabilities," in *Proceedings of the IEEE DySPAN Conference*, September/October 2015, pp. 200–208.

[24] R. Atawia, H. Abou-zeid, H. S. Hassanein, and A. Noureldin, "Joint chance-constrained predictive resource allocation for energy-efficient video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1389–1404, May 2016.

[25] 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 Communications Magazine*, vol. 54, no. 2, pp. 124–130, February 2016.

[26] D. Lee, S. Zhou, and Z. Niu, "Spatial modeling of scalable spatially-correlated log-normal distributed traffic inhomogeneity and energy-efficient network planning," in *Proceedings of the IEEE WCNC Conference*, April 2013, pp. 1285–1290.

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

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

[29] D. Gross, *Fundamentals of queueing theory*. John Wiley & Sons, 2008.

[30] S. Ahmed and A. Shapiro, "Solving chance-constrained stochastic programs via sampling and integer programming," *INFORMS Tutorials in Operations Research*, pp. 261–269, 2008.

[31] J. Luedtke and S. Ahmed, "A sample approximation approach for optimization with probabilistic constraints," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 674–699, 2008.

[32] IBM, "Optimization model development toolkit for mathematical and constraint programming (CPLEX)," http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud, 2012.

[33] J. Linderoth, "Lecture notes on integer programming," January 2005. [Online]. Available: http://homepages.cae.wisc.edu/~linderot/classes/ie418/lecture2.pdf

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

[35] S. Ahmed and A. Shapiro, "The sample average approximation method for stochastic programs with integer recourse," ISyE Georgia Institute of Technology, Tech. Rep., 2002.

[36] A. Bianco, R. Birke, L. Giraudo, and M. Palacin, "OpenFlow switching: Data plane performance," in *Proceedings of the IEEE ICC Conference*, May 2010, pp. 1–5.