M. Wirth, A. Ortiz, and A. Klein, ”Risk-Aware Bandits for Digital Twin Placement in Non-Stationary Mobile Edge Computing”, in *IEEE International Conference on Communications (ICC)*, Denver, CO, USA, June 2024.

Risk-Aware Bandits for Digital Twin Placement in Non-Stationary Mobile Edge Computing

Maximilian Wirth, Andrea Ortiz, and Anja Klein
Communications Engineering Lab, Technische Universität Darmstadt, Germany
{m.wirth, a.ortiz, a.klein}@nt.tu-darmstadt.de

***Abstract*—In the context of the metaverse, digital twins (DTs) enable the convergence of the virtual space and the physical world. To this aim, in this work, we investigate DT placement in Mobile Edge Computing (MEC) systems, i.e., the selection of an edge server (ES) to host the DT of a physical system (PS). DT placement in MEC is a challenging task. Firstly, the dynamic characteristics of MEC systems, e.g., wireless channels and the loads of the ESs, are unknown and can exhibit a statistically non-stationary behavior. Secondly, the accuracy of the DT relies on periodically synchronizing the PS with its DT. However, in order for the synchronization to be successful, the incurred latency must be below a predefined deadline. Thirdly, the synchronization should be energy efficient as many PSs are battery powered. Lastly, switching between ESs causes additional overhead as it requires the migration of the DT to a new ES. In this work, we investigate the DT placement problem in a dynamic and statistically non-stationary MEC system. Our goal is to jointly minimize the synchronization latency and energy consumed by the PS while accounting for the overhead caused by switching between ESs. Furthermore, we aim for reducing the risk of failed synchronization events. To this end, we propose a novel risk-aware piece-wise stationary Multi-Armed Bandit (MAB) algorithm. Our simulations verify that our proposed algorithm outperforms state-of-the-art schemes by 50% and 58% in terms of the percentage of failed synchronizations and the number of DT migrations, respectively.**

I. Introduction

Digital twins (DTs) are virtual software-based representations of physical systems (PSs), such as Internet of Things (IoT) devices or autonomous vehicles [1]. A DT emulates the behavior of its PS in real-time by simulating the PS's status. For this reason, DTs are regarded as a key enabling technology for the metaverse. The metaverse is a virtual three-dimensional space in which humans, represented by avatars, can interact with other humans and objects in real-time using virtual reality and augmented reality [2]. Leveraging DTs, the metaverse merges the physical and virtual space by constructing digital replicas of the physical world [3] and allowing DTs to interact on behalf of their PSs with other objects and avatars. Enabled by the future sixth-generation (6G) mobile networks, the metaverse is expected to benefit various applications, ranging from gaming over education to healthcare [4].

For a seamless integration of the physical world in the metaverse, DTs need to accurately represent their PSs. To maintain a precise replica of its PS, the DT relies on regular status updates from the PS, i.e., the PS periodically sends data about its current state to the DT over a wireless link. The DT, which is hosted on a computation server, processes this data to update its model of the PS. This process is called synchronization. However, having in mind that IoT devices are battery powered, two challenges arise when synchronizing DTs. Firstly, ensuring a frequent and timely synchronization for a reliable PS emulation. Secondly, using the available energy at the PS efficiently for the status update transmission.

In order to achieve a small synchronization latency and reduce the energy consumption of the PS, it is advantageous to deploy the computation servers that host the DTs in proximity of the PSs. Mobile Edge Computing (MEC) [5] is particularly suitable for this purpose, because in MEC systems, computation servers are deployed at the network edge. Furthermore, by flexibly changing the edge server (ES) that hosts the DT to another ES, one can account for dynamic changes of the environment or PS mobility while maintaining a low synchronization latency.

The aim of DT placement in MEC systems is to select the optimal ES for hosting the DT considering the synchronization latency and energy consumed by the PS. The DT placement is challenging, as both, synchronization latency and energy consumption, depend on the dynamic characteristics of the MEC system, i.e., the quality of the wireless channels between the PS and the ESs as well as the ESs' computational capabilities and their varying loads. Moreover, the migration of a DT from one ES to another ES causes additional overhead due to the effort required for instantiating the DT on the new ES.

To accurately represent the physical world in the metaverse, recent works solve the DT placement problem using optimization theory. [6], [7] and [8] derive algorithms to approximately solve optimization problems for the minimization of the synchronization latency. However, these works are based on the unrealistic premise of perfect knowledge of the MEC system characteristics, i.e., wireless channels, ES computation capabilities and loads. In a realistic setting, these quantities cannot be known in advance as they change over time.

Other works treat the DT placement problem from an online learning perspective. The authors of [9] and [10] propose algorithms for the synchronization latency minimization while accounting for uncertainty about the dynamic MEC system characteristics. However, these works fail to address the fact that MEC systems can evolve in a statistically non-stationary fashion. Examples for non-stationary changes are sudden obstructions of the line-of-sight (LoS) transmission path that deteriorate the channel quality or abrupt increases in the ES load. Statistical non-stationarity is particularly challenging, as an online-learning algorithm has to autonomously identify

This work has been funded by the German Federal Ministry of Education and Research (BMBF) project Open6GHub, grant number 16KISK014, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Projektnummer 210487104 - SFB 1053 MAKI, by DAAD with funds from BMBF and by the LOEWE Center emergenCity.

when the learned policy is no longer valid without knowing when or how a change in the performance occurred. Moreover, despite the fact that [9] and [10] consider a synchronization latency constraint, i.e., a maximum latency allowed for the synchronization process, they only aim at satisfying the constraint on average. This means that some synchronization events might fail as they cannot be completed before the deadline. In order for the DT to be a precise representation of its PS in the metaverse, it is crucial to finish the synchronization processes to ensure that the DT can properly track the PS's dynamics. Thus, the risk of not completing the synchronization process should be reduced.

In this paper, we consider DT placement in a statistically non-stationary MEC system. In our model, a PS selects one ES for hosting its DT, while having no prior knowledge about the system characteristics, such as expected uplink data rates as well as ES computation capabilities and loads. Our contributions are:

- We investigate ES selection for DT placement in an uncertain, dynamic and statistically non-stationary environment. We model the wireless channels and the loads of the ESs with piece-wise stationary random variables.
- We propose a novel risk-aware reinforcement learning algorithm, named *Risk-Aware Discounted Upper-Confidence-Bound* (RAD-UCB), for a joint minimization of the PS's energy consumption and the synchronization latency. RAD-UCB is based on Multi-Armed Bandits (MABs) and quickly adapts to the statistically non-stationary behavior of MEC systems. Furthermore, RAD-UCB reduces the risk of not completing the synchronization processes. Moreover, it considers the additional overhead incurred by switching the selected ESs.
- Our simulation results show that our proposed algorithm strikes an excellent balance between adaptability to statistically non-stationary environment changes, low risk of failed synchronization processes as well as low average energy consumption and synchronization latency.

In the following, Sec. II introduces the system model. The problem formulation and our proposed algorithm are explained in Sec. III and Sec. IV, respectively. In Sec. V, the simulation results are presented and Sec. VI concludes the paper.

II. System Model

A. Overview

Our scenario consists of a PS and a set $\mathcal{K} = \{1, \dots, K\}$ of $K \in \mathbb{N}$ base stations. Each base station is equipped with an adjacent ES that can host the DT of the PS. For notational simplicity, in the remainder of this work, we consider a base station with co-located ES as one unit and refer to it as an ES. In order for the DT to maintain a reliable model of the PS, it is assumed that the PS needs to synchronize with its DT periodically every τ seconds. Thus, the considered finite time horizon is divided into $T \in \mathbb{N}$ time steps of length τ.

We assume that the ESs are positioned at fixed locations. Furthermore, similarly to [11], it is assumed that the PS only

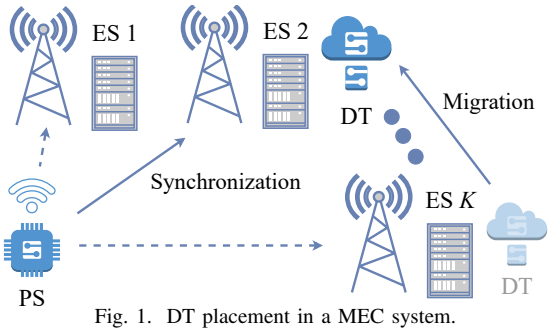

Fig. 1. DT placement in a MEC system.

makes small movements. Thus, the distances between the PS and the ESs stay approximately constant during the considered time horizon of T time steps. However, the PS's mobility can still cause significant variations in the quality of the wireless channel, as the PS can move from line-of-sight (LoS) to no-line-of-sight (NLoS).

At the beginning of every time step $t = 1, \dots, T$, the PS selects one ES $k \in \mathcal{K}$ for hosting its DT. Switching between ESs can improve the performance, as it allows for an adaptation to environment dynamics, e.g., changes of wireless channels or ES loads. However, it also causes an additional overhead, as the DT needs to be migrated to and instantiated on the new ES. Due to the complexity of real systems, the aforementioned overhead is hard to quantify [11]. In order to facilitate our considerations, we model the additional effort required for the DT migration with the constant $\beta \in \mathbb{R}^+$.

For synchronization, in every time step t, the PS wirelessly transmits a status update with the size of D bits to the selected ES k. Upon receiving the PS's status update, ES k processes it to update the corresponding DT. The computational complexity associated with processing the PS's status update is denoted by Γ and measured in CPU-cycles. After ES k completed the status update processing of the DT, it instantly informs the PS about the completion of the synchronization process via a perfect feedback channel with negligible latency.

If the PS cannot complete the status update transmission or does not receive the ES's feedback before the end of time step t, i.e., within τ seconds, the synchronization process is considered unsuccessful and terminated, i.e., either the PS stops the transmission or informs the ES to cancel the status update processing. At the beginning of the next time step, the PS selects again an ES and sends a new status update. Since the wireless channels as well as the ESs' computational capabilities and loads are dynamic and unknown to the PS, it cannot be guaranteed that the synchronization process will always be completed within the required time of τ seconds.

B. Communication Model

The channel coefficient $h_{k,t}$ of the wireless channel between the PS and ES k in time step t is sampled from a Rician distribution. Therefore, both, LoS and NLoS transmission conditions, are accounted for. $h_{k,t}$ is assumed to be constant for the duration of one time step. As in [12], $h_{k,t}$ is given by

$$h_{k,t} = \sqrt{\frac{\kappa_k}{\kappa_k + 1}} e^{j\theta_{k,t}} + \sqrt{\frac{1}{\kappa_k + 1}} \tilde{h}_{k,t}, \tag{1}$$

where the phases $\theta_{k,t}$ are uniformly distributed on the interval $[0, 2\pi]$ and statistically independent. Additionally, $\tilde{h}_{k,t}$ is drawn from a zero-mean and unit-variance complex Gaussian distribution. Moreover, $\kappa_k \geq 0$ denotes the energy ratio between LoS and NLoS for the wireless channel between PS and ES k, i.e., $\kappa_k = 0$ corresponds to NLoS and the larger κ_k, the stronger the LoS component. κ_k is assumed to be piece-wise constant, i.e., κ_k can change at certain time steps and stays constant for the time steps in between. These changes originate from the PS's mobility and mean that $h_{k,t}$ is drawn from a piece-wise stationary probability distribution. The channel gain of the wireless channel between PS and ES k in time step t is denoted by $H_{k,t}$ and can be expressed as

$$H_{k,t} = d_{k,t}^{-\epsilon/2}\, h_{k,t}, \tag{2}$$

where $d_{k,t}$ is the distance between PS and ES k in time step t. $d_{k,t}^{-\epsilon/2}$ is the path loss and $\epsilon \geq 2$ is the path loss coefficient.

It is assumed that the PS shares the wireless link to the ES with other users. To avoid interference between the PS and other connected users, each ES uses a separate frequency band and an orthogonal frequency-division multiple-access scheme. In order to account for the influence of other connected users on the load of the wireless network, $N_{k,t}^{\text{tx}}$ denotes the number of additional users that connect to ES k in time step t. $N_{k,t}^{\text{tx}}$ is sampled from a Poisson distribution $\text{Pois}(\eta_k)$ with parameter $\eta_k > 0$. According to Shannon's channel capacity formula, the highest possible data rate when transmitting from the PS to ES k in time step t is given by

$$R_{k,t} = \frac{B}{1 + N_{k,t}^{\text{tx}}} \log_2 \left(1 + \frac{|H_{k,t}|^2 P}{\sigma_n^2}\right) \quad \text{in } \frac{\text{bit}}{\text{s}}, \tag{3}$$

where P is the PS's constant transmit power, σ_n^2 denotes the thermal noise power and $B > 0$ is the maximum system bandwidth, which the PS shares with the other $N_{k,t}^{\text{tx}}$ connected users. The parameter η_k is assumed to be piece-wise constant. Thus, the bandwidth allocated to the PS is a piece-wise stationary random variable, which captures varying loads of the wireless network. Finally, the transmission latency when selecting ES k in time step t can be expressed as

$$T_{k,t}^{\text{tx}} = \frac{D}{R_{k,t}}. \tag{4}$$

C. Computation Model

The computation resources available at ES $k \in \mathcal{K}$ are denoted by $F_k^{\max}$ and measured in CPU-cycles per second. Moreover, let $N_{k,t}^{\text{ES}}$ be the number of users being served by ES k in addition to the PS in time step t. $N_{k,t}^{\text{ES}}$ is drawn from a Poisson distribution $\text{Pois}(\nu_k)$ with $\nu_k > 0$. It is assumed that $F_k^{\max}$ is split equally among all users simultaneously served by ES k. Consequently, the computation resources allocated to the PS, if it selects ES k in time step t, are given by

$$F_{k,t} = \frac{F_k^{\max}}{1 + N_{k,t}^{\text{ES}}}. \tag{5}$$

In order to capture the varying loads of the ESs, ν_k is assumed to be piece-wise constant. As a result, $F_{k,t}$ is a sample from a piece-wise stationary random distribution. The latency term $T_{k,t}^{\text{ES}}$ caused by the processing of the PS's status update is given by

$$T_{k,t}^{\text{ES}} = \frac{\Gamma}{F_{k,t}}. \tag{6}$$

D. Synchronization Latency and PS's Energy Consumption

Taking into account that the synchronization process is terminated if it is not completed within τ seconds, the PS cannot perceive synchronization latencies larger than τ. Therefore, we define the synchronization latency $T_{k,t}$ when ES k hosts the DT of the PS in time step t as

$$T_{k,t} = \min\{T_{k,t}^{\text{tx}} + T_{k,t}^{\text{ES}},\ \tau\} \tag{7}$$

and the PS's energy consumption associated with the DT synchronization as

$$E_{k,t} = \min\{T_{k,t}^{\text{tx}} P,\ \tau P\}. \tag{8}$$

Note that both, $T_{k,t}$ and $E_{k,t}$, are realizations of piece-wise stationary random variables, as the corresponding probability distributions depend on the piece-wise constant parameters κ_k, η_k and ν_k. We assume that all the underlying probability distributions for $T_{k,t}$ and $E_{k,t}$ can change in up to $\Phi \in \mathbb{N}$ time steps for all $k \in \mathcal{K}$ until $t = T$ is reached, with $\Phi << T$.

III. Problem Formulation

Hosting the PS's DT on an ES incurs a cost in terms of the synchronization latency $T_{k,t}$ and the PS's energy consumption $E_{k,t}$. For this purpose, we define the cost $C_{k,t}$ associated with hosting the DT on ES k in time step t as

$$C_{k,t} = \alpha \tilde{E}_{k,t} + (1 - \alpha)\tilde{T}_{k,t} \quad \in [0, 1], \tag{9}$$

where $\tilde{T}_{k,t} = T_{k,t}/\tau$ and $\tilde{E}_{k,t} = E_{k,t}/(\tau P)$ are the normalized synchronization latency and PS's energy consumption, respectively, with $\tilde{T}_{k,t}, \tilde{E}_{k,t} \in [0, 1]$. The parameter $\alpha \in [0, 1]$ is a weighting factor for the latency and energy consumption.

Let $y_{k,t}$ be a decision variable, i.e., $y_{k,t} = 1$ if the PS selects ES k in time step t and $y_{k,t} = 0$ if not, and $\mathbb{1}_{\{\cdot\}}$ denote the indicator function. Accounting for the fact that besides the cost $C_{k,t}$ of hosting the DT on an ES, switching ESs causes the additional overhead β, we formulate our optimization problem

$$\underset{\{y_{k,t}\}_{k\in\mathcal{K},\ t\in\{1,\ldots,T\}}}{\text{minimize}} \quad \sum_{t=1}^{T}\sum_{k=1}^{K} y_{k,t} C_{k,t} + \beta \mathbb{1}_{\{y_{k,t}=1 \wedge y_{k,t-1}\neq 1\}} \tag{10a}$$

$$\text{subject to} \quad \sum_{k=1}^{K} y_{k,t} = 1, \qquad \forall t, \tag{10b}$$

$$y_{k,t} \in \{0, 1\}, \quad \forall k,\ \forall t, \tag{10c}$$

where constraint (10b) enforces that only one ES is selected per time step and constraint (10c) ensures that the decision variable is binary.

IV. RAD-UCB Algorithm

A. Overview

Solving the optimization problem in (10) requires non-causal knowledge of the synchronization cost $C_{k,t}$ of every ES $k \in \mathcal{K}$ and for every time step $t = 1, \ldots, T$. However, in real systems, the PS has neither prior knowledge about the statistically non-stationary behavior of the wireless channels nor about the ESs' computation capabilities and loads. Thus, the PS has to autonomously learn the optimal ES in an unknown and dynamically changing environment. To this end, we propose *Risk-Aware Discounted Upper-Confidence-Bound* (RAD-UCB). RAD-UCB is a novel risk-aware reinforcement learning approach that reduces the risk of selecting ESs that lead to failed synchronization events, is capable of adapting to statistically piece-wise stationary changes in the behavior of the cost, and reduces the number of DT migrations.

RAD-UCB is based on a MAB approach and aims to learn the expected cost of every ES. In the MAB context, the PS is the learning agent, which has no prior knowledge about the expected cost or the probability of successfully completing the synchronization associated to every ES. The arms of the bandit correspond to the K ESs in the set $\mathcal{K}$ which the PS can choose from. Moreover, we define the reward as the cost $C_{k,t}$ associated with hosting the DT on an ES. Note that this definition slightly abuses the terminology as the reward is usually maximized and our goal is minimizing the reward.

In order to reduce the risk of selecting ESs that lead to an incomplete synchronization process, RAD-UCB penalizes failed synchronization events by increasing the associated cost by a finite factor $\rho \geq 1$. Furthermore, to adapt to piece-wise stationary probability distributions of the cost $C_{k,t}$, i.e., changes in the expected cost over time, RAD-UCB considers discounting of the past rewards as done in [13]. In particular, RAD-UCB uses discounted estimates of the expected synchronization cost. To this aim, the discount factor $\gamma \in (0, 1]$ is considered. Using γ, RAD-UCB computes a weighted average of the obtained cost samples in which recently obtained cost samples are weighted higher than older ones. Note that in this way, we account for the fact that older samples might have been obtained even before the occurrence of a non-stationary change in the probability distribution of the cost. Additionally, RAD-UCB takes into account the overhead for switching between ESs by limiting the exploration of ESs if the synchronization events are expected to be successful when choosing the ES with lowest expected cost.

B. ES Selection Process

The main objective of RAD-UCB is to select the ES with the lowest estimated expected cost while reducing the risk of a failed synchronization and accounting for the migration overhead. As the cost $C_{k,t}$ is not known in advance, in every time step t, RAD-UCB updates the estimates $\hat{\mu}^{\text{cost}}_{k,t}$ for the expected synchronization cost of every ES $k \in \mathcal{K}$.

The pseudocode of RAD-UCB is found in Algorithm 1. Let $N_{k,t}$ be a variable to track the number of times each ES has been selected. $N_{k,t}$ is initialized with zeros for all $k \in \mathcal{K}$ (line 2) before the iteration over the time steps $t = 1, \ldots, T$ starts (lines 3-20). To guarantee that we have a sample for the estimated cost $\hat{\mu}^{\text{cost}}_{k,t}$ of each ES $k \in \mathcal{K}$, in the first K time steps, every ES is selected once (lines 6-7). In every succeeding time step $t > K$, the PS selects an ES k_t based on the estimated cost $\hat{\mu}^{\text{cost}}_{k,t}$ and the exploration term $b_{k,t}$. The exploration term $b_{k,t}$ encourages the selection of ESs that have not been selected frequently until time step t. Specifically, the PS selects an ES k_t according to the following rule (lines 4-5)

$$k_t = \arg\min_{k \in \mathcal{K}} \left(\hat{\mu}^{\text{cost}}_{k,t-1} - b_{k,t-1} \right). \tag{11}$$

Algorithm 1 RAD-UCB

1: Input Parameters: γ, ξ, ρ, λ and T
2: Set $N_{k,0} = 0, \quad \forall k$.
3: **for** each $t = 1, \ldots, T$ **do**
4: **if** $N_{k,t-1} > 0 \; \forall k \in \mathcal{K}$ **then**
5: Select best ES k_t. ▷ Eq. (11)
6: **else**
7: Select ES k_t randomly from $\{k \in \mathcal{K} | N_{k,t-1} = 0\}$ with equal probability.
8: **end if**
9: Observe synchronization cost $C_{k_t,t}$ and latency $T_{k_t,t}$ for selected ES k_t.
10: **if** Synchronization is unsuccessful **then**
11: Set $C_{k_t,t} \leftarrow \rho \, C_{k_t,t}$.
12: Set $T_{k_t,t} \leftarrow \rho \, T_{k_t,t}$.
13: **end if**
14: Update $\hat{\mu}^{\text{cost}}_{k,t}$ and $N_{k,t}$ for all $k \in \mathcal{K}$. ▷ Eqs. (12) and (13)
15: Update $b_{k,t}$ and $\hat{\mu}^{\text{lat}}_{k,t}$ for all $k \in \mathcal{K}$. ▷ Eqs. (14) and (15)
16: Identify ES $k^*_t = \arg\min_{k \in \mathcal{K}} \hat{\mu}^{\text{cost}}_{k,t}$ with lowest expected cost.
17: **if** $\hat{\mu}^{\text{lat}}_{k^*_t,t} \leq \tau$ **then**
18: Set $b_{k,t} \leftarrow \lambda \, b_{k,t}$ for all $k \in \mathcal{K}$.
19: **end if**
20: **end for**

Next, the PS observes the synchronization cost $C_{k_t,t}$ and latency $T_{k_t,t}$ (line 9). If the synchronization process was incomplete, RAD-UCB increases $C_{k_t,t}$ and $T_{k_t,t}$ by multiplying both with ρ in order to account for the risk of a failed synchronization (lines 10-13). Note that ρ tunes the risk-awareness, i.e., $\rho = 1$ corresponds to the risk neutral case and the larger ρ, the greater the focus on minimizing the risk of incomplete synchronization. Afterwards, the estimated expected cost is updated for all ESs $k \in \mathcal{K}$ as

$$\hat{\mu}^{\text{cost}}_{k,t} = \frac{1}{N_{k,t}} \sum_{n=1}^{t} \gamma^{t-n} \, C_{k,n} \, \mathbb{1}_{\{k_n = k\}} \tag{12}$$

(line 14) and the counter $N_{k,t}$ is updated as

$$N_{k,t} = \sum_{n=1}^{t} \gamma^{t-n} \, \mathbb{1}_{\{k_n = k\}}, \tag{13}$$

where the potentially penalized cost samples $C_{k,n}$ obtained in time step n are discounted with the discount factor γ^{t-n}. As mentioned before, this discounting allows RAD-UCB to adapt to the changes in the piece-wise stationary cost distributions.

The exploration term $b_{k,t}$ is updated for all ESs $k \in \mathcal{K}$ as

$$b_{k,t} = \xi \sqrt{\frac{\log\left(\sum_{k'=1}^{K} N_{k',t}\right)}{N_{k,t}}} \tag{14}$$

(line 15), where $\xi > 0$ is a factor for adjusting the exploration sensitivity. The term $b_{k,t}$ encourages the exploration of ES k if $b_{k,t}$ is large, i.e., if the counter $N_{k,t}$ is small. This happens either if ES k has not been explored in many time steps, or if the last time step ES k has been selected lies many time steps in the past. While exploration helps to identify the optimal ES and changes of the expected cost, a frequent exploration can also be disadvantageous, as this increases the impact of the migration overhead. Furthermore, from a risk-aware point of view, the risk of incomplete synchronization processes might be increased due to recurrent exploration. Therefore, we aim for a risk-aware reduction of the exploration frequency. Specifically, RAD-UCB reduces the exploration terms $b_{k,t}$ for all ESs if it is expected that the synchronization processes can be completed when picking the ES that minimizes the cost. To this aim, RAD-UCB first updates estimates of the expected synchronization latency $\hat{\mu}^{\text{lat}}_{k,t}$ for all ESs (line 15) as

$$\hat{\mu}^{\text{lat}}_{k,t} = \frac{1}{N_{k,t}} \sum_{n=1}^{t} \gamma^{t-n} \, T_{k,n} \, \mathbb{1}_{\{k_n = k\}}. \tag{15}$$

Next, RAD-UCB determines the ES k^*_t with the lowest expected cost (line 16). If the expected synchronization latency $\hat{\mu}^{\text{lat}}_{k^*_t,t}$ when picking ES k^*_t is below the deadline τ, RAD-UCB reduces the exploration term $b_{k,t}$ with a scaling factor $\lambda \in (0, 1]$ for all ESs (lines 17-19). If even the best ES, in terms of cost, is not expected to complete the synchronizations, RAD-UCB explores more opportunistically as the PS might identify an ES that improved due to a non-stationary change.

V. Numerical Results

We consider an area of 500 m $\times$ 500 m for our simulations. We assume that this area is partitioned into 25 squares of equal size. If not stated otherwise, we place $K = 16$ ESs uniformly inside of this area at the intersections of the squares. For every simulation instance, we place the PS at a random location within the considered area and draw the available computation resources for every ES k from uniform distributions with intervals stated in Table I. For capturing the non-stationary variations of the wireless channels between the PS and the ESs as well as the loads of the ESs, we assume that the underlying probability distributions describing the aforementioned quantities change at Φ time steps within the considered time horizon, i.e., there are $\Phi + 1$ stationary phases of equal duration. Specifically, for every stationary phase, we assume that for a random number of ESs, which is drawn with equal probability from $\{1, \ldots, K\}$, the distribution parameters κ_k, ν_k and η_k are randomly redrawn from uniform distributions with intervals given in Table I. The remaining simulation parameters, which are based on [7] and [9], are summarized in Table I. To benchmark RAD-UCB, the following state-of-the-art MAB approaches are considered:

- *UCB* [14]: The risk-neutral Upper-Confidence-Bound MAB algorithm for stationary reward distributions.
- *D-UCB* [13]: Risk-neutral Discounted Upper-Confidence-Bound MAB algorithm, that adapts to piece-wise stationary reward distributions by discounting reward samples.

TABLE I
Simulation Parameters

Parameter	Value	Parameter	Value
B	20 MHz	D	1 Mb
σ^2_n	10^{-13} W	Γ	10^8 CPU-cycles
P	100 mW	τ	50 ms
ϵ	3	α	0.75
$F^{\max}_k$	$[8, 24]$ GHz	ν_k	$[1, 10]$
κ_k	$[0, 5]$	η_k	$[1, 10]$
γ	0.98	ρ	2
ξ	0.005	λ	0.001

- *Risk-Aware Oracle*: Assumes perfect knowledge of the expected synchronization cost and penalizes failed synchronization events.

The hyper-parameters for all baseline schemes are tuned such that the benchmark algorithms perform best in the considered simulation scenario. We use MATLAB for our simulations and consider the average of 400 Monte Carlo simulation runs with a time horizon of $T = 2000$ time steps.

To achieve a fair comparison, Figure 2 shows the synchronization cost per time step, normalized with respect to the fraction of synchronizations that failed in the respective time step. Specifically, for every time step, we weight the cost with the corresponding fraction of failed synchronization events. Moreover, we consider $\Phi = 1$ non-stationarity, which is located at time step $t = 1000$, i.e., the expected cost changes at $t = 1000$. During the first stationary phase, RAD-UCB reaches a stable normalized cost 25% above the oracle within 100 time steps. Note that the oracle represents a lower bound for the performance of RAD-UCB by assuming perfect knowledge of the expected cost associated to each ES. In the succeeding time steps, the normalized cost achieved by RAD-UCB slowly converges closer to the oracle achieving a 5% higher cost than the oracle at $t = 750$. UCB and D-UCB also reach a stable cost after 100 time steps. However, the cost achieved by UCB and D-UCB is approximately twice as large as for RAD-UCB. This can be explained by the fact that, unlike RAD-UCB, UCB and D-UCB are risk-neutral in the sense that they do not account for the risk of failed synchronization events. Thus, the cost normalized in terms of the frequency of uncompleted synchronization processes is larger. Furthermore, it can be observed that the normalized cost obtained from RAD-UCB behaves more stable over time compared to D-UCB. This is caused by D-UCB frequently re-exploring suboptimal ESs. Our proposed algorithm does not suffer from this, because we safely limit the exploration as long as we can expect the synchronization events to be successful. After the non-stationarity at $t = 1000$, both, RAD-UCB and D-UCB, exhibit a similar reaction time. Within 500 time steps, they both converge to a stable normalized cost comparable to before the occurrence of the non-stationarity. It is worth noting that within 100 time steps, RAD-UCB already reaches a normalized cost smaller than the normalized cost achieved by D-UCB and UCB before $t = 1000$. Within the considered time horizon, UCB is not capable of converging to the cost achieved

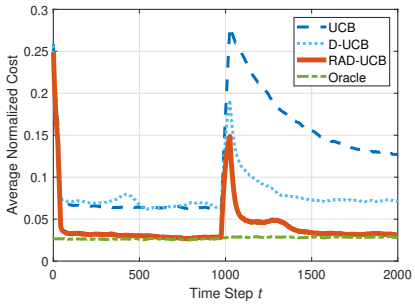

Fig. 2. Normalized synchronization cost per time step for $\Phi = 1$ and $K = 16$ ESs.

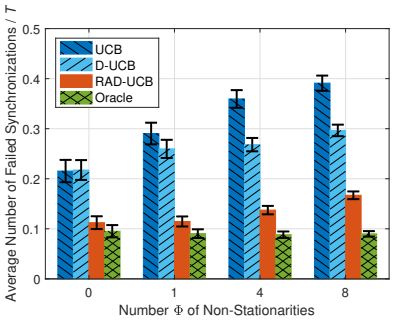

Fig. 3. Frequency of failed synchronization processes versus Φ for $K = 16$ ESs.

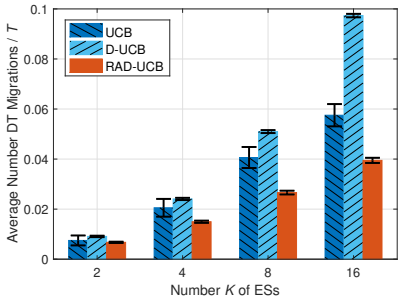

Fig. 4. Frequency of DT migrations versus number of ESs K for $\Phi = 5$.

before $t = 1000$, as it is basing its decisions on outdated cost estimates from before the non-stationarity.

In Figure 3, the number of failed synchronizations during the time horizon divided by T, i.e., the frequency of failed synchronizations, is displayed over the number Φ of non-stationarities. The error-bars indicate the 95% confidence intervals. For the stationary case, i.e., $\Phi = 0$, RAD-UCB exhibits a 50% lower frequency of failed synchronization events when compared to the risk-neutral algorithms UCB and D-UCB. Moreover, RAD-UCB performs close to the oracle, as the frequency of uncompleted synchronization processes of RAD-UCB is only 18% higher. When increasing the number of non-stationarities Φ, the frequency of failed synchronization events increases for UCB, D-UCB and RAD-UCB, as the algorithms are unaware of when the non-stationarities appear and have to relearn their estimates after every occurrence. For $\Phi = 8$, compared to the risk-aware RAD-UCB, the frequency of failed synchronizations achieved by UCB and D-UCB is 140% and 85% higher, respectively. Additionally, the confidence intervals for RAD-UCB are significantly narrower compared to UCB and D-UCB, which indicates that RAD-UCB consistently achieves a lower frequency of failed synchronization events.

In Figure 4, the number of DT migrations during the time horizon normalized with T, i.e., the frequency of DT migrations, is shown for different numbers K of ESs. As explained in Sec. II, switching ESs comes at an additional overhead. With increasing K, the frequency of migrations tends to grow, as there are more ESs to choose from during exploration phases. However, since our proposed algorithm RAD-UCB safely restricts the exploration if the synchronization events can be successfully completed, the frequency of DT migrations is only 5.7 times larger when increasing the number of ESs by factor of 8. For D-UCB and UCB, the DT migration frequency grows by a factor of 7.6 and 10.8, respectively.

VI. Conclusion

In this work, we considered DT placement in a statistically non-stationary MEC system. Specifically, a PS selects an ES for hosting its DT. The dynamic characteristics of the considered MEC system, i.e., wireless channels and the ESs' loads, were modeled using piece-wise stationary random variables. The objective of the DT placement problem was to jointly minimize the synchronization latency and the energy consumed by the PS while reducing the additional overhead incurred by switching the selected ES. Moreover, we considered the risk of not completing the DT synchronization process within a given deadline. We proposed a novel risk-aware MAB algorithm that reduces the risk of failed synchronization events, quickly adapts to piece-wise stationary changes in the probability distributions of synchronization latency and energy consumption, and reduces the number of DT migrations. Numerical results revealed the effectiveness of our proposed algorithm when benchmarking it against state-of-the-art schemes.

References

[1] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019.

[2] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Communications Surveys & Tutorials*, 2022.

[3] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[4] M. Xu, W. C. Ng, W. Y. B. Lim, J. Kang, Z. Xiong, D. Niyato, Q. Yang, X. S. Shen, and C. Miao, "A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges," *IEEE Communications Surveys & Tutorials*, 2022.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[6] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, and G. Karakostas, "Digital twin placement for minimum application request delay with data age targets," *IEEE Internet of Things J.*, 2023.

[7] O. Hashash, C. Chaccour, W. Saad, K. Sakaguchi, and T. Yu, "Towards a decentralized metaverse: Synchronized orchestration of digital twins and sub-metaverses," *arXiv preprint arXiv:2211.14686*, 2022.

[8] O. Chukhno, N. Chukhno, G. Araniti, C. Campolo, A. Iera, and A. Molinaro, "Placement of social digital twins at the edge for beyond 5G IoT networks," *IEEE Internet of Things J.*, vol. 9, no. 23, pp. 23 927–23 940, 2022.

[9] J. Li, J. Wang, Q. Chen, Y. Li, and A. Y. Zomaya, "Digital twin-enabled service satisfaction enhancement in edge computing," in *IEEE Conf. Computer Commun.* IEEE, 2023, pp. 1–10.

[10] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6g," *IEEE Internet of Things J.*, vol. 8, no. 22, pp. 16 219–16 230, 2021.

[11] Y. Zhou, C. Shen, and M. van der Schaar, "A non-stationary online learning approach to mobility management," *IEEE Trans. on Wireless Commun.*, vol. 18, no. 2, pp. 1434–1446, 2019.

[12] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[13] A. Garivier and E. Moulines, "On upper-confidence bound policies for non-stationary bandit problems," *arXiv preprint arXiv:0805.3415*, 2008.

[14] R. Agrawal, "Sample mean based index policies by o (log n) regret for the multi-armed bandit problem," *Advances in Applied Probability*, vol. 27, no. 4, pp. 1054–1078, 1995.