Edge Intelligence-Based Ultra-Reliable and Low-Latency Communications for Digital Twin-Enabled Metaverse

Dang Van Huynh, *Student Member, IEEE*, Saeed R. Khosravirad, *Member, IEEE*, Antonino Masaracchia, *Member, IEEE*, Octavia A. Dobre, *Fellow, IEEE*, and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—In this letter, we propose a novel digital twin scheme supported metaverse by jointly considering the integrated model of communications, computing, and storage through the employment of mobile edge computing (MEC) and ultra-reliable and low latency communications (URLLC). The MEC-based URLLC digital twin architecture is proposed to provide powerful computing infrastructure by exploring task offloading, and task caching techniques in nearby edge servers to reduce the latency. In addition, the proposed digital twin scheme can guarantee stringent requirements of reliability and low latency, which are highly applicable for the future networked systems of metaverse. For this first time in the literature, our paper addresses the optimal problem of the latency/reliablity in digital twins-enabled metaverse by optimizing various communication and computation variables, namely, offloading portions, edge caching policies, bandwidth allocation, transmit power, computation resources of user devices and edge servers. The proposed scheme can improve the quality-of-experience of the digital twin in terms of latency and reliability with respect to metaverse applications.**

***Index Terms*—Digital twin, metaverse, mobile edge computing, ultra-reliable and low latency communications.**

I. Introduction

IT IS expected that the next generation of wireless communications network will revolutionise human lives in an unprecedented way. With a growing trend in the development of mobile wireless augmented reality (AR) and virtual reality (VR), the manufacturing empowered by industrial Internet-of-Things (IIoT) has been transformed from physical prototypes to virtual and immersive interactions, which eventually will profoundly enhance the efficiency of productivity. Recently, leveraged by real-time optimization theory, artificial intelligence, and digital twin (DT) has been considered as a promising technique to realise the practical implementation of metaverse. For supporting immersive and interoperable metaverse, DT, implemented through computer simulation but completely different from the computer model, will represent digital replicas of physical objects with a real-time two-way interaction. As such, DT provides real-time insight into how the metaverse can be operated and help to optimise the decision-making.

DT empowered by edge intelligence will revolutionise the networks and help attain the aim of connected intelligence for immersive metaverse. Edge intelligence is an emerging concept by optimizing the efficiency, allocation, and operation of resources and tasks that can integrate with the ultra-reliable and low-latency communications (URLLC) to support the seamless real-time immersion in the metaverse. However, research in DT is still in early stage and far from fully realizing the potential of metaverse. There is still little understanding of major issues in implementing DT from communication, networking, and computing perspectives. One of the formidable challenges in DT is that the huge amount of high-fidelity and real-time data require immensely computational capability to satisfy extreme quality-of-service (QoS) constraints. Similarly, AR/VR applications and tactile Internet resulted in DT demand strictly stringent QoS services in terms of very high reliability and very low latency transmission, which is a significant challenge for current wireless mobile networks.

Recently, edge computing assisted DT has attracted attention from the research community [1]–[3]. In particular, a DT edge network has been presented in [1] to deal with the offloading latency minimization problem. The actor-critic deep reinforcement learning (DRL) has been exploited to solve the optimization problem. In [2], the mobile edge computing (MEC) architecture with the assistance of DT for IIoT has been investigated. This letter has taken into account various variables including transmit power, user association, offloading portions, and the estimated processing of IIoT devices to minimise the end-to-end latency with an iterative optimization algorithm. Another DT-asisted MEC based on edge collaboration has been addressed in [3], which deals with the edge selection and task offloading variables by applying the decision tree algorithm and the DRL-based solution. More recently, the combination of URLLC and the edge computing in the DT paradigm has been introduced in [4], [5]. In particular, the latency minimization problem formulated in DT-enabled MEC including URLLC-based transmission latency and task processing latency was solved by the alternative optimization solution. However, these DT schemes may not be directly applicable to metaverse applications where the storage of AR/VR data has been neglected.

Motivated by the aforementioned discussion, this letter proposes the edge intelligence with URLLC by taking into account the joint design of communication, computing, and storage from the perspective of DT for extreme time-sensitive

Manuscript received 24 March 2022; revised 9 May 2022; accepted 26 May 2022. Date of publication 2 June 2022; date of current version 9 August 2022. This work was supported in part by the U.K. Royal Academy of Engineering (RAEng) under the RAEng Research Chair and Senior Research Fellowship Scheme under Grant RCSRF2021\11\41. The work of Octavia A. Dobre was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), though its Discovery Program. The associate editor coordinating the review of this article and approving it for publication was A. A. Nasir. *(Corresponding author: Trung Q. Duong.)*

Dang Van Huynh, Antonino Masaracchia, and Trung Q. Duong are with Queen's University Belfast, Belfast BT7 1NN, U.K. (e-mail: dhuynh01@qub.ac.uk; a.masaracchia@qub.ac.uk; trung.q.duong@qub.ac.uk).

Saeed R. Khosravirad is with Nokia Bell Labs, Chicago, IL, USA (e-mail: saeed.khosravirad@nokia-bell-labs.com).

Octavia A. Dobre is with Memorial University, St. John's, NL A1C 5S7, Canada (e-mail: odobre@mun.ca).

Digital Object Identifier 10.1109/LWC.2022.3179207

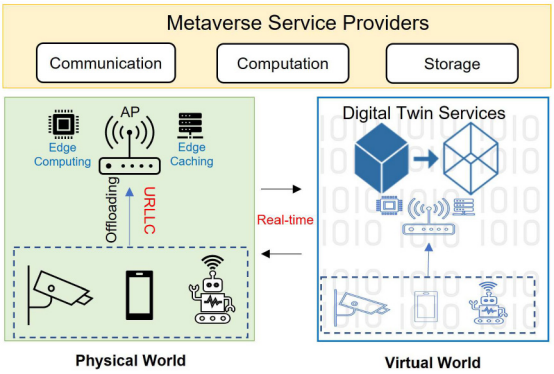

Fig. 1. Edge-based 6G URLLC-enabled Metaverse System.

applications in metaverse. More specifically, we formulate a latency minimization problem under stringent constraints of URLLC-based transmissions by optimizing edge caching strategies, task offloading policies, as well as computation and communication resources. The problem is solved by an effective iterative algorithm in the fashion of the alternating optimization approach, that demonstrates the effectiveness of the proposed DT in supporting metaverse applications.

II. System Model and Problem Formulation

We propose a DT-enabled metaverse employed URLLC and edge intelligence, which consists of the physical world and the virtual world as presented in Fig. 1. In the physical world, there is a set of M IIoT devices (UEs), $\mathcal{M} = \{1, 2, \ldots, M\}$ which are randomly distributed in an industrial area such as a smart factory. These IIoT devices are connected with an access point (AP) via URLLC links. There is an edge server (ES) associated to the AP to provide both edge computing and edge caching services in order to reduce the end-to-end (e2e) latency of computation-intensive tasks offloaded from the UEs. In the virtual world, DT services fully replicate the devices of the physical world including the device configuration, resource budget, and current working states in order to interact with the physical objects in real-time. In the control centre, metaverse service providers jointly optimise communication, computation, and storage resources and make prompt decisions to efficiently manage the entire system.

A. Communication Model of DT-Enabled Metaverse

The AP is equipped with L antennas to serve M single-antenna UEs. Let $\mathbf{h}_m = \sqrt{g_m}\bar{\mathbf{h}}_m \in \mathbb{C}^{L\times 1}$ be the channel vector between the AP and the m-th UE, where g_m denotes the large-scale channel coefficient, and $\bar{\mathbf{h}}_m$ is the small-scale fading following the distribution of $\mathcal{CN}(0, \mathbf{I})$. Let $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_M] \in \mathbb{C}^{L\times M}$ be the channel matrix from M devices to the AP. The allocated bandwidth coefficient of the m-th UE is denoted by b_m. The signal-to-noise (SNR) of the m-th UE is given by $\gamma_m(b_m, p_m) = \frac{p_m\|\mathbf{h}_m\|^2}{b_m B N_0}$, where B is the system bandwidth, p_m is the transmit power of the m-th UE, and N_0 is the single-side noise spectral density. Then, the uplink URLLC transmission rate (bit/s) is expressed as follows [6], [7]

$$R_m(b_m, p_m) \approx \frac{B}{\ln 2}\Big[b_m \ln(1+\gamma_m(b_m,p_m)) - \sqrt{\frac{b_m V_m(b_m,p_m)}{\phi B}}Q^{-1}(\epsilon_m)\Big], \quad (1)$$

where ϕ is the transmission time interval, ϵ_m is decoding error probability, $\gamma_m(b_m, p_m)$ denotes the SNR of the m-th UE, $Q^{-1}(.)$ is the inverse function $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp^{(\frac{-t^2}{2})}\, dt$, and V_m is the channel dispersion given by $V_m(b_m, p_m) = 1 - [1+\gamma_m(b_m, p_m)]^{-2}$.

As a results, the uplink transmission latency is given by

$$T_m^{\mathsf{co}}(\alpha_m, p_m, b_m) = \frac{D_m}{R_m(p_m, b_m)} \quad (2)$$

where D_m is the data size (bits).

B. Computation Model of DT-Enabled Metaverse

A task that comes from the m-th UE is characterized by a tuple $J_m = (D_m, C_m, T_m^{\max})$, where C_m is the required computation resource (cycles) and $T_m^{\max}$ is the maximum latency requirement of this task. Let $\boldsymbol{\alpha} \triangleq \{\alpha_m\}_{\forall m}$ be the portion of tasks which is executed locally at the UEs. Then, the offloaded portion from the m-th UE executed by the ES is $(1-\alpha_m)$.

The DT service for local processing of the m-th UE is denoted as $\mathrm{DT}_m^{\mathsf{ue}}$, which can be modeled as $\mathrm{DT}_m^{\mathsf{ue}} = (f_m^{\mathsf{ue}}, \hat{f}_m^{\mathsf{ue}})$, where f_m^{ue} is the estimated processing rate of the m-th UE and $\hat{f}_m^{\mathsf{ue}}$ is the deviation between the estimated value and the real value of the processing rate. The deviation can be positive or negative to model the replicated processing rate in the DT [1], [2], [5]. Consequently, the local processing latency of the m-th for executing a task locally is given by

$$T_m^{\mathsf{ue}}(\alpha_m, f_m^{\mathsf{ue}}) = \frac{\alpha_m C_m}{f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}}}, \quad (3)$$

which can be derived from $T_m^{\mathsf{ue}} = \tilde{T}_m^{\mathsf{ue}} + \Delta T_m^{\mathsf{ue}}$ with the estimated processing latency $\tilde{T}_m^{\mathsf{ue}} = \alpha_m C_m / f_m^{\mathsf{ue}}$ and the deviation latency $\Delta T_m^{\mathsf{ue}} = \alpha_m C_m \hat{f}_m^{\mathsf{ue}} / [f_m^{\mathsf{ue}}(f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}})]$.

Similarly, the processing latency of the ES to execute the offloaded task from the m-th UE can be calculated as follows

$$T_m^{\mathsf{es}}(\alpha_m, f_m^{\mathsf{es}}) = \frac{(1-\alpha_m)C_m}{f_m^{\mathsf{es}} - \hat{f}_m^{\mathsf{es}}}, \quad (4)$$

where f_m^{es}, $\hat{f}_m^{\mathsf{es}}$ are the estimated processing rate and the deviation value of the ES in its DT. As we can see from (3) and (4), the deviation between the real and estimated processing rate has affected the system performance. As such, it is important for the DT to correctly estimate all the parameters of the physical world to avoid the performance loss.

C. Latency and Energy Model With Edge Caching

We model task caching strategies by using integer decision variables, $\mathbf{s} \triangleq \{s_m\} | s_m \in \{0, 1\}, \forall m$ which indicates whether the task J_m is cached at the ES ($s_m = 1$) or not ($s_m = 0$). When the task is cached at the ES, only the edge processing latency is calculated. On the other hand, when the tasks is not cached, it is normally processed with the task offloading computing model. We note that the results returned from the AP to UEs are typical small (e.g., controlled messages) and the AP transmits the messages with more power than the UEs so that we only consider the uplink transmission latency in this letter [2], [3]. As a result, the latency model with edge caching is expressed as

$$T_m^{\mathsf{e2e}}(\alpha_m, s_m, b_m, p_m, f_m^{\mathsf{ue}}, f_m^{\mathsf{es}}) = \frac{s_m C_m}{f_m^{\mathsf{es}} - \hat{f}_m^{\mathsf{es}}} + (1-s_m) \times \left[T_m^{\mathsf{ue}}(\alpha_m, f_m^{\mathsf{ue}}) + T_m^{\mathsf{co}}(\alpha_m, p_m, b_m) + T_m^{\mathsf{es}}(\alpha_m, f_m^{\mathsf{es}})\right]. \quad (5)$$

The total energy consumption of the m-th UE, consisting of the energy for computation (E_m^{cp}) and communication (E_m^{cm}), is given by $E_m^{\mathsf{tot}}(s_m, \alpha_m, f_m^{\mathsf{ue}}, b_m, p_m) = (1-s_m)(E_m^{\mathsf{cp}} + E_m^{\mathsf{cm}}) = (1-s_m)[\alpha_m \frac{\theta}{2} C_m (f_m^{\mathsf{ue}} - \hat{f}_m^{\mathsf{ue}})^2 + \frac{(1-\alpha_m)p_m D_m}{R_m(b_m,p_m)}]$, where the constant θ is the computation power parameter for energy consumption of the UEs [2], [8].

D. Optimization Problem Formulation

In this letter, we aim to minimise the total e2e latency among M UEs by optimizing offloading portions, caching policies, bandwidth allocation, transmit power, estimated processing rate of UEs and ES subject to URLLC QoS, the energy budget of UEs, and computing and caching capacity of the ES. The addressed problem is formulated as follows:

$$\min_{\substack{\alpha_m, s_m, b_m, \\ p_m, f_m^{\mathsf{ue}}, f_m^{\mathsf{es}}}} \sum_{m=1}^{M} T_m^{\mathsf{e2e}}(\alpha_m, s_m, b_m, p_m, f_m^{\mathsf{ue}}, f_m^{\mathsf{es}}), \tag{6a}$$

$$\text{s.t. } T_m^{\mathsf{e2e}}(\alpha_m, s_m, b_m, p_m, f_m^{\mathsf{ue}}, f_m^{\mathsf{es}}) \le T_m^{\max}, \forall m, \tag{6b}$$

$$\sum_{m=1}^{M} b_m \le 1, \forall m, \tag{6c}$$

$$R_m(b_m, p_m) \ge R_{\min}, \forall m, \tag{6d}$$

$$E_m^{\mathsf{tot}}(s_m, \alpha_m, f_m^{\mathsf{ue}}, b_m, p_m) \le E_m^{\max}, \forall m, \tag{6e}$$

$$\sum_{m=1}^{M} [s_m f_m^{\mathsf{es}} + (1-s_m)(1-\alpha_m) f_m^{\mathsf{es}}] \le F_{\max}^{\mathsf{es}}, \tag{6f}$$

$$\sum_{m=1}^{M} s_m D_m \le S_{\max}^{\mathsf{es}}, \tag{6g}$$

$$\boldsymbol{\alpha} \in \mathscr{A}, \mathbf{p} \in \mathscr{P}, \mathbf{f} \in \mathscr{F}, \tag{6h}$$

where $\mathscr{A} \triangleq \{\alpha_m, \forall m | 0 \le \alpha_m \le 1, \forall m\}$, $\mathscr{P} \triangleq \{p_m, \forall m | 0 \le p_m \le P_m^{\max}, \forall m\}$, $\mathscr{F} \triangleq \{\mathbf{f} = \{f_m^{\mathsf{ue}}, f_m^{\mathsf{es}}\}, \forall m | 0 \le f_m^{\mathsf{ue}} \le F_{\max}^{\mathsf{ue}}, \forall m; 0 \le f_m^{\mathsf{es}} \le F_{\max}^{\mathsf{es}}\}$ are the sets of constraints of the offloading decisions, the uplink transmission power, and the processing rates, respectively. Constraint (6b) indicates maximum latency requirements. Constraints (6c), (6d) represent the bandwidth allocation requirement and the QoS of the uplink rate, respectively. The maximum energy consumption requirement of the UE is described in constraint (6e). Finally, the maximum computing and caching capacity of the ES are presented in (6f) and (6g), respectively.

III. Proposed Solution

The problem (6) is highly computationally complex due to the non-convex objective function (6a), strong coupled integer and continuous variables in (6a), (6e), (6f) and non-convex constraints (6d), (6e), (6f). Therefore, we propose an alternating optimization (AO)-based solution by alternatively solving the problem in one set of variables while keeping other variables fixed. In the following subsections, we develop the final solution with three subproblems, namely, caching policy optimization, offloading policy optimization, and joint communication and computation resources optimization.

A. Caching Policy Optimization

We are in the position to find the next iterative point $\mathbf{s}^{(i+1)}$ for $\mathbf{s}$ with fixed values of $(\boldsymbol{\alpha}^{(i)}, \mathbf{b}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f}^{(i)})$, which leads to solve the following subproblem

$$\mathsf{SP1}: \min_{\substack{s_m \in \{0,1\} | \\ \boldsymbol{\alpha}^{(i)}, \mathbf{b}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f}^{(i)}}} \sum_{m=1}^{M} T_m^{\mathsf{e2e}}(s_m), \tag{7a}$$

$$\text{s.t.} \quad \text{(6b), (6e), (6f), (6g)}. \tag{7b}$$

This problem is non-convex due to the integer variable s_m. To solve (7), we define $t_m^s = T_m^{\mathsf{ue}}(\alpha_m^{(i)}, f_m^{\mathsf{ue}(i)}) + T_m^{\mathsf{co}}(\alpha_m^{(i)}, p_m^{(i)}, b_m^{(i)}) + T_m^{\mathsf{es}}(\alpha_m^{(i)}, f_m^{\mathsf{es}(i)}), \forall m$. Next, we sort t_m^s among M UEs in a descending order and decide to cache the task with a higher t^s (higher latency) until the constraints (6g) is violated with respect to other constraints in order to find the optimal values of $\mathbf{s}$ at the i-th iteration. This procedure simply requires a few constraint checks ($< M$) [8].

B. Offloading Policy Optimization

This subsection finds the next iterative point $\boldsymbol{\alpha}^{(i+1)}$ for $\boldsymbol{\alpha}$ with fixed $(\mathbf{s}^{(i+1)}, \mathbf{b}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f}^{(i)})$, given by the following subproblem:

$$\mathsf{SP2}: \min_{\substack{\alpha_m \in [0,1] \mid \mathbf{s}^{(i+1)}, \\ \mathbf{b}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f}^{(i)}}} \sum_{m=1}^{M} T_m^{\mathsf{e2e}}(s_m), \tag{8a}$$

$$\text{s.t.} \quad \text{(6b), (6e), (6f), (6h)}, \tag{8b}$$

which is obviously a convex problem with all linear constraints. This problem can be solved efficiently with CVX [9]. The problem (8) includes M scalar variables and $5M + 1$ linear constraints; therefore, the per-iteration computational complexity of solving (8) is $\mathcal{O}(M^2\sqrt{5M+1})$ [10, Sec. 6].

C. Joint Communication and Computation Optimization

We find the next iterative point $(\mathbf{b}^{(i+1)}, \mathbf{p}^{(i+1)}, \mathbf{f}^{(i+1)})$ with fixed values of $(\mathbf{s}^{(i+1)}, \boldsymbol{\alpha}^{(i+1)})$. Following [7], [11], when the received SNR is sufficiently high, the approximation $V_m \approx 1$ is applied, then the transmission rate can be rewritten as

$$\begin{aligned} R_m &\approx \frac{B}{\ln 2}\left[b_m \ln(1+\gamma_m(b_m,p_m)) - \sqrt{\frac{b_m}{\phi B}} Q^{-1}(\epsilon_m)\right] \\ &\triangleq \frac{B}{\ln 2}[G_m(b_m,p_m) - W_m(b_m)], \end{aligned}$$

where $G_m(b_m, p_m) = b_m \ln(1 + \gamma_m(b_m, p_m))$ and $W_m(b_m) = \sqrt{b_m}\frac{Q^{-1}(\epsilon_m)}{\sqrt{\phi B}}$.

By following the approximations in the Appendix, the transmission rate can be expressed as follows

$$\begin{aligned} R_m(b_m, p_m) &\ge R_m^{(i)}(b_m, p_m) \\ &\triangleq \frac{B}{\ln 2}\Big[\mathcal{G}_m^{(i)}(b_m, p_m) - \mathcal{W}_m^{(i)}(b_m)\Big], \end{aligned} \tag{9}$$

where $\mathcal{G}_m^{(i)}(b_m, p_m)$ and $\mathcal{W}_m^{(i)}(b_m)$ are defined as (17) and (19) in the Appendix, respectively.

As a result, constraint (6g) is alternatively approximated as follows

$$R_m^{(i)}(b_m, p_m) \ge R_{\min}, \forall m, k. \tag{10}$$

To handle constraint (6e), we introduce the variables $\boldsymbol{\tau}_m \triangleq \{\tau_m\}_{\forall m}$ that satisfy $1/R_m \le \tau_m, \forall m$. Then, the constraint (6e) is equivalently given by

$$\begin{cases} (1-s_m^{(i+1)})\Big[\dfrac{\theta}{2}\alpha_m^{(i+1)} C_m\Big(f_m^{\mathsf{ue}}-\hat{f}_m^{\mathsf{ue}}\Big)^2 \\ +\,(1-\alpha_m^{(i+1)})p_m\tau_m,\Big] \le E_m^{\max}, \forall m, \qquad (11a) \\ \dfrac{1}{R_m^{(i)}} \le \tau_m. \qquad (11b) \end{cases}$$

The constraint (11b) is now convex, while (11a) is still non-convex. Therefore, we apply the following inequality with $x = p_m$, $\bar{x} = p_m^{(i)}$, $y = \tau_m$, $\bar{y} = \tau_m^{(i)}$

$$xy \le \frac{1}{2}\left(\frac{\bar{y}}{\bar{x}}x^2 + \frac{\bar{x}}{\bar{y}}y^2\right). \qquad (12)$$

Then, the constraint (11a) can be inner approximated as follows

$$(1-s_m^{(i+1)})\Big[\frac{\theta}{2}\alpha_m^{(i+1)} C_m\Big(f_m^{\mathsf{ue}}-\hat{f}_m^{\mathsf{ue}}\Big)^2 + \frac{(1-\alpha^{(i+1)})}{2}\left(\frac{\tau_m^{(i)}}{p_m^{(i)}}p_m^2 + \frac{p_m^{(i)}}{\tau_m^{(i)}}\tau_m^2\right)\Big] \le E_m^{\max}, \forall m. \quad (13)$$

Finally, the non-convex objective function (6a) with $T_m^{\mathsf{e2e}}(\alpha_m^{(i+1)}, s_m^{(i+1)}, b_m, p_m, f_m^{\mathsf{ue}}, f_m^{\mathsf{es}})$ can be innerly approximated as

$$T_m^{\mathsf{e2e}} \le \Big(1-s_m^{(i+1)}\Big)\Big[\frac{\alpha_m^{(i+1)} C_m}{f_m^{\mathsf{ue}}-\hat{f}_m^{\mathsf{ue}}} + D_m\tau_m(b_m,p_m) + \frac{(1-\alpha_m^{(i+1)})C_m}{f_m^{\mathsf{es}}-\hat{f}_m^{\mathsf{es}}}\Big] + \frac{s_m^{(i+1)}C_m}{f_m^{\mathsf{es}}-\hat{f}_m^{\mathsf{es}}} \triangleq \mathcal{T}_m^{(i)}. \quad (14)$$

Consequently, we solve the following convex problem for the resource allocation

$$\textsf{SP3-Convex:} \quad \min_{\mathbf{b},\mathbf{p},\mathbf{f}|\ \mathbf{s}^{(i+1)},\boldsymbol{\alpha}^{(i+1)}} \sum_{m=1}^{M} \mathcal{T}_m^{(i)}, \qquad (15a)$$

$$\text{s.t. } \mathcal{T}_m^{(i)}(\alpha_m^{(i+1)}, s_m^{(i+1)}, b_m, p_m, f_m^{\mathsf{ue}}, f_m^{\mathsf{es}}) \le T_m^{\max}, \quad (15b)$$

$$\text{(6c), (6f), (6h), (10), (11b), (13).} \qquad (15c)$$

For complexity analysis, this problem consists of $4M$ scalar decision variables and $7M+2$ linear or quadratic constraints, which results in the per-iteration computational complexity of $\mathcal{O}(16M^2\sqrt{7M+2})$ [10, Sec. 6].

D. Proposed Algorithm

For the i-th iteration, let us denote $\mathcal{S}_1(\mathbf{s}^{(i)})$, $\mathcal{S}_2(\boldsymbol{\alpha}^{(i)})$, $\mathcal{S}_3(\mathbf{b}^{(i)}, \mathbf{p}^{(i)}, \mathbf{f}^{(i)})$ as the feasible sets of the subproblems (7), (8) and (15), respectively. The proposed algorithm to solve the problem (6) is given as

Algorithm 1 AO-Based Algorithm for Solving (6)

1: **Input**: Set $i = 0$ and randomly choose initial feasible points $\mathcal{S}_1^{(0)}$, $\mathcal{S}_2^{(0)}$ and $\mathcal{S}_3^{(0)}$ to constraints in (7), (8) and (15); set the tolerance $\varepsilon = 10^{-3}$ and the maximum number of iterations $I^{\max} = 20$.
2: **Repeat**
3: Solve problem (7) for given $\mathcal{S}_2^{(i)}, \mathcal{S}_3^{(i)}$ with the procedure described in subsection III-A to obtain the optimal solution of $(\mathbf{s}^\star)$ and update $\mathcal{S}_1^{(i+1)} := (\mathbf{s}^\star)$;
4: Solve problem (8) with given $\mathcal{S}_1^{(i+1)}, \mathcal{S}_3^{(i)}$ to obtain the optimal solution of $(\boldsymbol{\alpha}^\star)$ and update $\mathcal{S}_2^{(i+1)} := (\boldsymbol{\alpha}^\star)$;
5: Solve problem (15) with given $\mathcal{S}_1^{(i+1)}, \mathcal{S}_2^{(i+1)}$ to obtain the optimal solution of $(\mathbf{b}^\star, \mathbf{p}^\star, \mathbf{f}^\star)$ and update $\mathcal{S}_3^{(i+1)} := (\mathbf{b}^\star, \mathbf{p}^\star, \mathbf{f}^\star)$;
6: Set $i := i+1$;
7: **Until** Convergence or $i > I^{\max}$.
8: **Output**: $\big(\mathbf{s}^\star, \boldsymbol{\alpha}^\star, \mathbf{b}^\star, \mathbf{p}^\star, \mathbf{f}^\star\big)$ and $\min \sum_{m=1}^{M}\{T_m^{\mathsf{e2e}}\}_{\forall m}$.

IV. Numerical Results

In our simulations, there are $M = 15$ UEs randomly distributed in a 100 m $\times$ 100 m square area [12] and the ES is located in the central position. Following [7], [13], the data size of computational tasks is set to 1354 bytes and the task complexity is $\eta_m \triangleq C_m/D_m = [100, 300]$ cycles/byte. The maximum processing rate of UEs and ES are set to $F_{\max}^{\mathsf{lo}} = 1.5$ GHz and $F_{\max}^{\mathsf{es}} = 30$ GHz, respectively. The maximum latency requirement for each task is $T_m^{\max} = 10$ ms [13]. Following [2], [7], [8] and [5], other parameters are set as $g_m = 10^{\mathsf{PL}(d_m)/10}$, $\mathsf{PL}(d_m) = -35.3 - 37.6\log_{10}$ dB, $N_0 = -174$ dBm/Hz, $B = 5$ MHz, $S_{\max}^{\mathsf{es}} = 60$ Kb, $E_m^{\max} = 3$ mJ, $\theta = 10^{-26}$ Watt.s^3/cycle^3, $\epsilon = 10^{-7}$, $L = 8$, and $P_m^{\max} = 23$ dB.

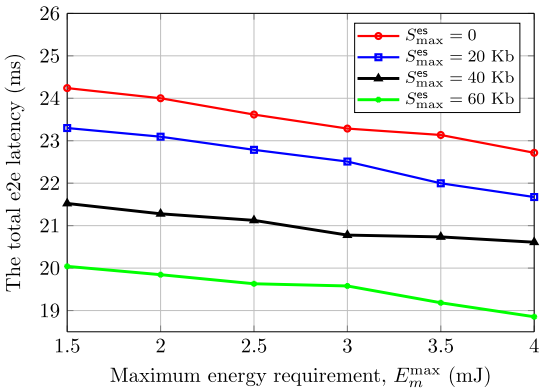

Fig. 2. The impact of the edge caching capacity ($S_{\max}^{\mathsf{es}}$) and UE's energy consumption budget ($E_m^{\max}$) in the scenarios of $M = 15$ UEs.

Impact of the ES caching capacity and UEs energy consumption budget: To investigate the impacts of the edge caching and the UEs energy requirement in reducing the latency, we have conducted simulations among different settings of the ES caching capacity and UEs energy budget. In particular, Fig. 2 clearly demonstrates that when the UEs energy consumption budget increases, the total e2e latency gradually declines. For instance, the minimal e2e latency in the scenarios of $S_{\max}^{\mathsf{es}} = 60$ Kb experiences a considerable decrease of approximately 10 ms when $E_m^{\max}$ approaches 4 mJ. Additionally, Fig. 2 also illustrates the effectiveness of the proposed task caching solutions by providing the obtained total latency with different levels of the ES caching capacity. In this regard, the higher the ES caching capacity is, the lower latency can be obtained. Importantly, we can clearly see that the gap between the $S_{\max}^{\mathsf{es}} = 60$ Kb and the conventional method (i.e., the non-caching scheme $S_{\max}^{\mathsf{es}} = 0$) is significantly large, which obviously proves that the task caching model is effective for the DT with time-sensitive metaverse applications.

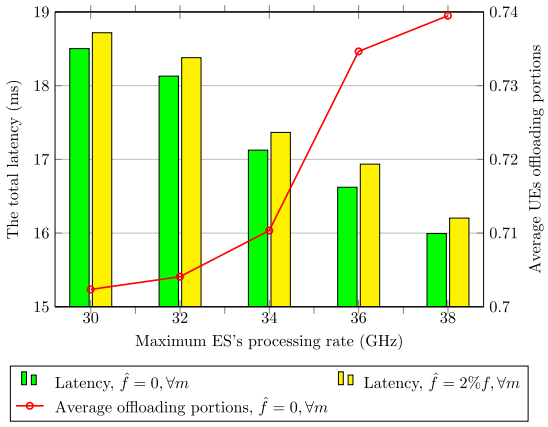

Fig. 3. The impact of ES's processing rate, deviation values and offloading behavior in the scenarios of $M = 15$ UEs and $S^{\mathsf{es}}_{\max} = 60$ Kb.

Impact of the ES processing rate: For the purpose of demonstrating the impact of the ES processing rate to the obtained latency, we run simulations with different settings of the maximum ES's processing rate ($F^{\mathsf{es}}_{\max}$). Fig. 3 reveals that there has been a gradual decline in the total e2e latency of UEs when the ES's computing capacity increases. In particular, the total latency decrease by nearly 2.5 ms when $F^{\mathsf{es}}_{\max}$ climbs to 38 GHz. Fig. 3 additionally indicates that the offloading portions of UEs steadily rises when the ES becomes more powerful, which proves that the proposed task offloading model works effectively. Finally, the impact of the deviation between the estimated and the real processing rate has been also displayed in Fig. 3. It can be clearly seen that the more accurately the DT estimates, the better performance can be obtained, and this, the proposed solution is highly applicable in practical scenarios.

V. Conclusion

In conclusion, we have proposed a DT framework to enable metaverse applications by jointly considering the communication, computing, and storage to minimise the latency performance. The optimal latency has been obtained by jointly optimizing various edge caching, communication and computation variables including offloading tasks, caching policies, bandwidth allocations, transmit powers, and the processing rate at EU and ES. The proposed iterative algorithm has effectively solved the problem in the fashion of the AO-based approach with three subproblems, namely the edge caching optimization, the task offloading optimization, and the resources allocation optimization. Finally, the selective numerical results have effectively validated the proposed solution.

Appendix

By applying the following inequality [14, eq. (73)]:

$$z\ln(1+\frac{x}{y}) \geq 2\bar{z}\ln(1+\frac{\bar{x}}{\bar{y}}) + \frac{\bar{z}\bar{x}}{\bar{x}+\bar{y}}\left(2-\frac{\bar{x}}{x}-\frac{y}{\bar{y}}\right) - \frac{\ln(1+\bar{x}/\bar{y})}{z}\bar{z}^2, \quad (16)$$

with $z = b_m$, $\bar{z} = b_m^{(i)}$, $x = p_m\|\mathbf{h}_m\|^2$, $\bar{x} = p_m^{(i)}\|\mathbf{h}_m\|^2$, $y = b_m BN_0$, and $\bar{y} = b_m^{(i)} BN_0$ for $G_m(b_m, p_m)$, we can innerly approximate $G_m(b_m, p_m)$ as follows

$$\begin{aligned} G_m(b_m,p_m) &\geq 2b_m^{(i)}\ln\left(1+\frac{p_m^{(i)}\|\mathbf{h}_m\|^2}{b_m^{(i)}BN_0}\right) \\ &+ \frac{b_m^{(i)}p_m^{(i)}\|\mathbf{h}_m\|^2}{p_m^{(i)}\|\mathbf{h}_m\|^2+b_m^{(i)}BN_0}\left(2-\frac{p_m^{(i)}\|\mathbf{h}_m\|^2}{p_m\|\mathbf{h}_m\|^2}-\frac{b_m}{b_m^{(i)}}\right) \\ &- \frac{\ln\left(1+p_m^{(i)}\|\mathbf{h}_m\|^2/b_m^{(i)}BN_0\right)(b_m^{(i)})^2}{b_m} \\ &\triangleq \mathcal{G}_{mk}^{(i)}(b_m,p_m). \end{aligned} \quad (17)$$

To handle $W_m(b_m)$, we apply the following inequality

$$\sqrt{x} \leq \frac{\sqrt{\bar{x}}}{2} + \frac{x}{2\sqrt{\bar{x}}}, \quad (18)$$

with $x = b_m$, $\bar{x} = b_m^{(i)}$ to approximate $W_m(b_m)$ as

$$W_m(b_m) \leq \frac{Q^{-1}(\epsilon_m)}{\sqrt{\phi B}}\left(\frac{\sqrt{b_m^{(i)}}}{2}+\frac{b_m}{2\sqrt{b_m^{(i)}}}\right) \triangleq \mathcal{W}_m^{(i)}. \quad (19)$$

References

[1] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12240–12251, Oct. 2020.

[2] T. Do-Duy, D. V. Huynh, O. A. Dobre, B. Canberk, and T. Q. Duong, "Digital twin-aided intelligent offloading with edge selection in mobile edge computing," *IEEE Wireless Commun. Lett.*, vol. 11, no. 4, pp. 806–810, Apr. 2022.

[3] T. Liu, L. Tang, W. Wang, Q. Chen, and X. Zeng, "Digital twin assisted task offloading based on edge collaboration in the digital twin edge network," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1427–1444, Jan. 2022.

[4] Y. Li, D. Van Huynh, T. Do-Duy, E. Garcia-Palacios, and T. Q. Duong, "Unmanned aerial vehicles-aided edge networks with ultra-reliable low-latency communications: A digital twin approach," *IET Signal Process.*, to be published.

[5] D. V. Huynh, V. D. Nguyen, V. Sharma, O. A. Dobre, and T. Q. Duong, "Digital twin empowered ultra-reliable and low-latency communications-based edge networks in industrial IoT environment," in *Proc. IEEE Int. Conf. Commun.*, Seoul, South Korea, May 2022, pp. 1–6.

[6] C. She, C. Yang, and T. Q. S. Quek, "Radio resource management for ultra-reliable and low-latency communications," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 72–78, Jun. 2017.

[7] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4692–4707, Oct. 2019.

[8] A. A. Nasir, "Latency optimization of UAV-enabled MEC system for virtual reality applications under Rician fading channels," *IEEE Wireless Commun. Lett.*, vol. 10, no. 8, pp. 1633–1637, Aug. 2021.

[9] M. Grant and S. Boyd. "CVX: MATLAB software for disciplined convex programming, version 2.1." Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[10] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Philadelphia, PA, USA: SIAM, 2001.

[11] C. Sun, C. She, C. Yang, T. Q. S. Quek, Y. Li, and B. Vucetic, "Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 402–415, Jan. 2019.

[12] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.

[13] "Study on scenarios and requirements for next generation access technologies, version 15.0.0," 3GPP, Sophia Antipolis, France, Rep. TR 38.913, 2018.

[14] A. A. Nasir, H. D. Tuan, H. H. Nguyen, M. Debbah, and H. V. Poor, "Resource allocation and beamforming design in the short blocklength regime for URLLC," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1321–1335, Feb. 2021.