IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 June 2024, accepted 16 July 2024, date of publication 19 July 2024, date of current version 29 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3431248

RESEARCH ARTICLE

Fog Computing Meets URLLC: Energy Minimization of Task Partial Offloading for URLLC Services

CHENHAO SHI[1], JINGRUI WEI[1], YAO ZHU[2], (Member, IEEE), AND ANKE SCHMEINK[2], (Senior Member, IEEE)

[1]School of Electronic Information, Wuhan University, Wuhan 430000, China
[2]Chair of Information Theory and Data Analytics, RWTH Aachen University, 52068 Aachen, Germany

Corresponding author: Yao Zhu (yao.zhu@isek.rwth-aachen.de)

This work was supported by the Federal Ministry of Education and Research of Germany (BMBF) in the Project "Open6GHub" under Grant 16KISK012.

ABSTRACT Ultra-high reliability and ultra-low latency communication (URLLC) are critical challenges for upcoming 6G applications. Cloud computing and mobile edge computing (MEC) offer potential solutions but incur high deployment and maintenance costs due to reliance on central or edge servers. Moreover, the surge in users and data exacerbates latency concerns. Therefore, with more flexible servers deployment, fog computing is more capable of URLLC requirements. In this work, we propose a fog computing model utilizing mobile devices' computing capabilities to mitigate latency delays. We characterise the problem as an optimisation problem in quadratic variables. And we reduce the problem to a mixed integer convex optimisation problem in two dimensions using decomposition subproblems. Based on this, we introduce a partial offloading algorithm based on the finite blocklength (FBL) mechanism, which improves the energy efficiency. Simulations demonstrate the efficiency of our algorithm in URLLC, with a 49% reduction in energy consumption compared to no retransmission and a 36% reduction in energy consumption compared to infinite blocklength (IBL) coding.

INDEX TERMS Fog computing, dynamic voltage and frequency scaling (DVFS), partial offloading, finite blocklength (FBL), ultra-high reliability and ultra-low latency communication (URLLC), 6G.

I. INTRODUCTION

As mobile smart devices become more and more embedded in the public's life, the wireless network is expected to support immediate application (e.g.,speech and image recognition, online game, virtual reality) [1], [2], which requires an ultra-high reliability and ultra-low latency communication (URLLC) service. Cloud computing delivers computing services over the Internet, providing businesses and individuals with a cost-effective, flexible, and efficient way to manage and use computing resources. Its key advantages include centralized management, large-scale resource pooling, and on-demand allocation, allowing users to access powerful computing capabilities via the Internet without worrying about hardware details.

However, traditional cloud computing may not be able to fulfill the stringent latency requirement, due to the latency and bandwidth limitations. To address the limitations of cloud computing, mobile edge computing (MEC) has emerged. MEC pushes computing and data processing closer to the data source, i.e., closer to where the data is generated and used [3]. This concept aims to reduce latency, improve bandwidth utilisation and support applications that require real-time decision making and processing. Typical scenarios

The associate editor coordinating the review of this manuscript and approving it for publication was Irfan Ahmed.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

for MEC include Internet of Things (IoT) devices, smart cities, industrial automation, etc [4], [5], [6]. Fog Computing is a further extension of MEC that focuses more on processing closer to the user. The very name of Fog Computing indicates its position between the ''cloud'' and the ''ground''. Unlike cloud computing, fog computing is more decentralised and can be deployed at the edge of the network close to the user's device, resulting in lower latency and higher real-time performance. Fog computing has huge advantages in terms of cost, latency and reliability. Specifically, on the cost side, bandwidth and data transfer costs are reduced through local processing and storage rather than server. In terms of latency, the advantage for fog computing is to further reduce the transmission latency due to the proximity and flexibility: any device has available computation capability can be the server, while the servers are usually dedicated in MEC. In terms of reliability, based on the cooperation between mobile devices, fog computing does not depend on specific servers, i.e., even if one device does not work, there may still be other reliable devices around the user. These features make fog computing particularly suitable for IoT, industrial automation, smart cities and other application scenarios that require low latency and high reliability.

URLLC is a specialized paradigm in cellular communication systems, designed to meet strict requirements for low-latency and high-reliability applications. It's crucial for mission-critical sectors like industrial automation, autonomous vehicles, and telemedicine, as well as immersive technologies such as AR/VR. Its importance extends to industries needing fast and dependable communication, establishing it as indispensable in modern technology and innovation [7].

In cloud computing, the considerable distance between the central server and the user often poses challenges in meeting the demands for URLLC. In such scenarios, MEC and fog computing emerge as more viable options, given their proximity to the computing resources essential for URLLC applications. Notably, fog computing stands even closer to end devices compared to MEC, rendering it poised for broader adoption in URLLC scenarios in the future [8]. The study in [9] investigates the integration of URLLC into cloud robotics (CR) applications, emphasizing the unlicensed band for potential implementation. Meanwhile, [10] employs URLLC links and utilizes digital twin (DT) to model edge server computing capacity, optimizing resource allocation in the system. Reference [11] introduces an investigation into the energy-aware task allocation issue within vehicular fog networks, taking URLLC into account. Reference [12] overcome the limitations of low latency requirements for intelligent transportation systems by proposing a novel architecture based on fog-cloud computing and software-defined networking (SDN). Additionally, [11] addresses the energy-aware task allocation problem in vehicular fog networks for URLLC in intelligent transportation systems. Moreover, to ensure deterministic and reliable performance for critical VEC services, [13] investigates the integration of Vehicle Edge Computing (VEC) into Intelligent Transportation Systems (ITS) to handle the substantial data generated by advanced vehicles. However, these studies concentrate on adapting the principles of Cloud Computing, MEC, and Fog Computing to construct pertinent models addressing real-world challenges, they do not enhance delay and reliability aspects from the channel and coding perspectives.

Indeed, although the fog computing networks provide a viable architecture for providing real-time computing services, in order to implement URLLC, we need to introduce Finite blocklength (FBL) Regime. FBL coding is a coding method proposed to solve the problem of high reliability and low latency for short packet data communication. Under the assumption of finite blocklength, the transmission is no longer arbitrarily reliable, and the error probably of the transmission remains large even when the coding rate is below the Shannon capacity while the blocklength is very short. Thus an exact approximation of the achievable coding rate for an additive Gaussian white noise (AWGN) channel under the FBL assumption is derived in [14], which allows the reliability problem to be solved. Moreover, [15], [16] mentions that in order to prevent data loss caused by FBL transmission error, the system can improve transmission reliability by means of retransmission. This modelling approach improves communication reliability while introducing additional energy consumption and delay costs. For multi-user MEC networks deploying FBL coding for wireless data transmission, [17] minimises the end-to-end error probability subject to FBL and energy consumption constraints. Reference [18] focuses on optimizing the decoding error probability and power allocation factor for joint decoding in NOMA systems, aiming to maximize effective throughput at the central user. However, there is insufficient work on how to design a fog computing network that applies the FBL mechanism to optimise energy consumption. Reference [19] proposes finite blocklength coding (FBC)-based strategies for joint beamforming and unmanned aerial vehicle (UAV) trajectory optimization, addressing statistical delay and error-rate bounded quality-of-service (QoS) challenges in URLLC with MEC. Considering finite blocklength channel codes, [20] explores a novel UAV-assisted URLLC service system for future wireless communication, addressing the UAV-deployment in achieving URLLC. Their main contribution is to consider the finite blocklength mechanism in Internet of Things (IoT) networks and optimise IoT device scheduling, power control, and resource allocation to minimise the average uplink transmit power, which provides important lessons for more flexible network deployments in the future. Moreover, [21] presents a UAV-enabled MEC system, where latency-sensitive tasks are offloaded from ground devices to a UAV-carried MEC server using URLLC. However, on one hand, these works achieve optimization at the policy and allocation level. In fact, we can achieve more flexible resource allocation by introducing more advanced hardware technologies. On the other hand, they overlook the trade-offs between system energy consumption, latency, and

reliability due to the limited blocklength, which is also crucial for system design [22].

Although FBL coding and retransmission mechanisms are good methods to implement URLLC, there are still risks. In general, as the system takes more retransmission attempts, the risk of latency violations increases, and the increase in communication time leads to a compression of the system computation time, which requires the system CPU to operate in a more flexible and high-frequency manner in order to complete the data processing as fast as possible. We therefore introduce dynamic voltage and frequency scaling (DVFS) techniques. DVFS technology adjusts the CPU operating frequency and voltage in real time according to the current workload [23]. The application of this technology gives mobile devices the ability to adaptively adjust computational resources to improve computing efficiency, reduce computing energy consumption and shorten computing time length. Furthermore, traditional cloud computing completely relies on cloud servers to perform computational tasks, this computational strategy is difficult to meet the requirements of URLLC, the combination of DVFS and fog network makes the task divisible, the computational task can be divided into two parts: the local computation and the server computation part. This enables the communication and computation strategy more flexible to reduce the overall energy consumption of the system. Reference [24] addresses the challenge of minimizing energy consumption in a fog network with dense terminal devices and servers. It formulates the problem of task offloading with DVFS and transmission power control. Reference [25] explores the impact of offloading portions of tasks to edge devices, considering the extra energy consumption for transmission and reception, and highlights the importance of optimizing this tradeoff for efficient MEC. Reference [26] investigates delay and energy efficiency in fog radio access networks with hybrid caching, exploring multiple caching and transmission strategies for enhanced flexibility in file placement and fetching.

However, none of these works have focused on the closed-loop communication process under the FBL regime. Although offloading computational tasks partially to fog computing networks is a good idea, the execution of downlink transmissions in closed-loop communication scenarios relies on successful uplink transmissions due to the assumption of FBL. In fact, most of the solutions focus on improving the reliability of the open-loop links and fail to address the resource allocation problem caused by the duplex asymmetry. To tackle this issue, a pioneering ARQ-based protocol was introduced in [27], functioning within a FBL regime. In this protocol, resources initially allocated for downlink slots can be flexibly reassigned to retransmit in the uplink in case of failures.

In this paper, we focus on partial offloading schemes in closed-loop communication scenarios under the FBL regime. We take the total expected energy consumption of the overall closed-loop system as the optimisation objective and optimise the CPU computing frequency, the transmission length for finite blocklength, the ratio of system offloading to local computation and the maximum allowable number of retransmission. We disassemble the problem into several sub-problems by analysing the feasible domains of the optimization variables, followed by a proof-of-convexity analysis that transforms the problem into a mixed integer convex optimization problem solved by the Karush-Kuhn-Tucke (KKT) condition. To the best of our knowledge, we are the first to study the partial offloading problem for closed-loop communication under the FBL regime. Specifically, our contribution can be summarised as follows:

- We develop a closed-loop communication model for fog computing under the FBL regime and investigate the partial offloading energy minimisation problem under this model, formulating the problem as a mixed integer convex problem.
- We analyse the feasible domains of each optimisation variable in the problem, giving specific upper and lower bounds. Moreover, we prove the convexity of the total expected energy consumption and the total transmission error probability with respect to the transmission blocklength and the offloading ratio, giving an expression for the optimal CPU computing frequency. Based on these, we reduce the problem to a convex problem in two variables and propose an algorithm based on the Block Coordinate Descent (BCD) method.
- We investigate the characteristics of the optimal solution by KKT condition, deducing the conditions for obtaining an optimal solution. In addition we provide a partial offloading algorithm under FBL regime. The results indicate that our algorithm produces a more accurate solution than traditional iterative algorithm. Quantitative experiments indicate that our design improves

TABLE 1. Abbreviation and definitions.

Abbreviation	Definition
MEC	Mobile Edge Computing
NACK	Negative Acknowledgement
FBL	Finite Blocklength
URLLC	Ultra-Reliable and Low-Latency Communication
HARQ	Hybrid Automatic Repeat reQuest
NOMA	Non-Orthogonal Multiple Access
DVFS	Dynamic Voltage and Frequency Scaling technology
IBL	Infinite Blocklength
IoT	Internet of Things
CR	Cloud Robotics
DT	Digital Twin
VEC	Vehicle Edge Computing
SDN	Software-Defined Networking
UAV	Unmanned Aerial Vehicle
QoS	Quality-of-Service
AWGN	Additive Gaussian White Noise
FS	Fog Server
FU	Fog User
SNR	Signal to Noise Ratio
KKT	Karush-Kuhn-Tucker
BCD	Block Coordinate Descent

significantly in terms of energy consumption. Compared to no retransmission, the retransmission mechanism improves energy consumption by 49%, and compared to infinite blocklength (IBL) coding, the FBL regime improves by 38%.

For readability, we give a list of Abbreviations TABLE 1 and a list of Notations TABLE 2 (Here $i \in [u, d]$, which represents the uplink or downlink.).

The rest of this paper is composed as fallows: In Section II, we demonstrate the overall model, then demonstrate the original problem formulation in Section III. Next, We analyse the feasible domains of the variables in detail in Section IV and prove the convexity of the expected total energy consumption and the total error probability over the variables, according to which we propose an optimisation framework, and at the end of this section we give an iterative algorithm based on the bisection approach. In Section V, we give the simulation results for the model and make an analytical validation. Furthermore, we summarise the full work in Section VI.

II. SYSTEM MODEL

We consider a small-scale fog-cooperative network, which is more widely utilized in future home cloud scenarios [28]. The network is composed of multiple devices with limited computational power cooperating with each other. Each wireless device in the network can be either a fog server (FS) or a computing-needed fog user (FU). Specifically, when a device has available computational resources and no pending tasks, it functions as a fog server by offering its own resources. However,once the device has tasks to compute and its local resources are insufficient, it switches to being a fog user, requesting resources from the FS. In this scenario, the server is no longer a central server but another device with comparable computing capability to the FU. If the device has a task to compute, then it becomes a FU and request resource from FS. Once the offloading is scheduled, the FU judiciously determines what portion of the task will be computed at FS, and what portion of the task will be offloaded to FU, what size a blocklength to send packets in, and the local computing frequency. Computing resource is abstracted as a profile with two parameters, i.e., (D, $T_{\max}$), where D and $T_{\max}$ denote the amount of input data to be computed and the latency requirement associated with the application, respectively. We model the number of cycles C required for the application as the number of input data for the calculation multiplied by the factor, i.e., $C = \alpha D$, where α is related to the computational complexity of the application. Furthermore, we define λ ($0 \leq \lambda \leq 1$) as the ratio of the number of locally executed size to the total input data packet. In addition, the data is transmitted in blocklength of length m (symbols).

Particularly, we exert a time slot model to demonstrate the entire structure, i.e., the system divides the time of the entire process into individual frames of length T, where each symbol is of length T_s. In this model, the frame length should

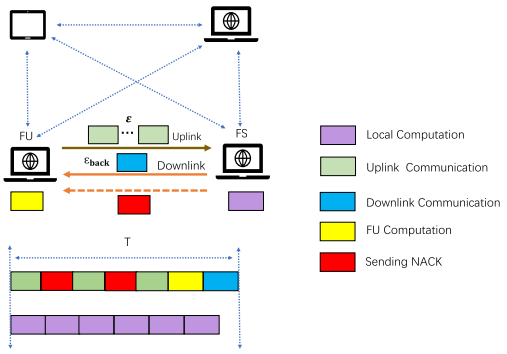

FIGURE 1. System model (End-to-End communication for fog computing networks within time slot model, where each device has comparable computing capacity).

gratify $T \leq T_{\max}$. Hence, the process can be described as follows: In the first step, a portion of the data processing (λD bits) is offloaded locally, and the rest of the data ((1-λ)D bits) is sent to the FS over the uplink with a finite blocklength of length m. In the second step, the FS processes the received data. In the third step, the FS dispatches the data back to the FU also with a blocklength of length m_{back}. In addition, the communication of data and the FS computation time are parallel to the local computation time, which represents that the maximum time spent by the system is the larger one of the two. The whole model is demonstrated in Fig. 1.

We assume that the channel experiences quasi-static frequency-flat Rayleigh fading. Therefore, it is assumed that the channel state remains constant from one frame to the next, and that the changes between frames follow a certain relationship. We characterize the channel gain as z. Then, the channel gain can be denoted as z_{u} and z_{d} respectively. The signal-to-noise ratio (SNR) of the links can be presented as

$$\gamma_{\text{u}} = \frac{\phi_{\text{u}} z_{\text{u}} P_{\text{u}}}{\sigma_{\text{u}}^2}, \tag{1}$$

$$\gamma_{\text{d}} = \frac{\phi_{\text{d}} z_{\text{d}} P_{\text{r}}}{\sigma_{\text{d}}^2}, \tag{2}$$

where ϕ_{u} and ϕ_{d} represent the path loss of uplink and downlink, σ_{u} and σ_{d} denote the noise power of the uplink and downlink.

A. RELIABILITY MODEL

Note that due to the FBL regime, error may arise during communication, and once an error occurs, the FS will send a Negative Acknowledgement (NACK) to the FU at a fixed time length t_{NK}. The FU will resend the data after receiving the NACK, and this process continues until the FS successfully decodes the data or reaches the maximum retransmission tolerance number $N_{\max}$.

1) PROBABILITY OF DECODING ERROR IN FBL REGIME

We adopt $C(\gamma) = \log_2(1+\gamma)$ for the Shannon Capacity and $V(\gamma) = 1-(1+\gamma)^{-2}$ for channel dispersion under a complex

AWGN channel. According to the FBL regime in [14], the error probability of each (re)transmission can be modeled as

$$\varepsilon = P(r, \gamma, m) \approx Q(\sqrt{\frac{m}{V(\gamma)}}(C(\gamma) - r)\ln 2), \quad (3)$$

where r is coding rate given by $r = \frac{(1-\lambda)D}{m}$ and $Q(.)$ is complementary error function, which satisfies $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$.

2) OVERALL RELIABILITY–TOTAL DECODING ERROR

Once the FS decodes incorrectly, it sends NACK to the FU requesting retransmission. Therefore, the overall reliability of the system is inextricably linked to the retransmission mechanism. Due to the small size of NACK, here we assume that FU decoding of NACK is error-free, meaning that the downlink is a control link. According to (3), we have

(a.) $N = 1$: No retransmission is considered, which indicates that the decoding error probability at this point is the total decoding error probability, i.e., $\varepsilon_{\text{tot}} = \varepsilon$.

(b.) $N \geq 1$: We employ $n = 1$ to represent the initial transmission, and the error probability associated with the initial transmission is denoted by ε. Extending this analogy, considering the independence of each transmission, the n^{th} decoding failure for the fog server (FS) occurs when all preceding n retransmissions fail. In other words, the probability of the n^{th} retransmission is ε^n. The cumulative error probability, without accounting for data backhaul, is then given by ε^N. Furthermore, to adhere to the communication's latency constraints, we allocate the remaining time, post the completion of transmission and fog server computation, exclusively for the return transmission process. In this process, a decoding error $\varepsilon_{\text{back}} = \varepsilon$ occurs. Consequently, the comprehensive error probability for the closed-loop communications is given by:

$$\varepsilon_{\text{tot}}(m, \lambda; N) = \varepsilon^N + \varepsilon_{\text{back}}. \quad (4)$$

Here, the error probability consists of two components, i.e., the retransmission error probability of the uplink and the backhaul error probability of the downlink.

B. ENERGY COST MODEL

In practice, the fog node can take various forms such as a cell phone, laptop, smartwatch, etc., each with limited battery energy. Consequently, our design objective is to minimize the energy consumption of these devices, enhancing the endurance of mobile devices while achieving URLLC. In our model, the overall energy consumption of the system encompasses several components: the energy consumption of transmitting data E_{t}, the energy consumption of receiving data E_{r}, the energy consumption related to NACK-reception/transmission E_{k}, and the energy consumption associated with local computation E_{c}. Moreover, given that FS is another device with finite computational resources, we also have to consider the expected energy consumption of FS. What demands emphasis is that the energy consumption of these processes is influenced by the number of retransmissions and the probability of error in decoding. Subsequently, we will compute the expected energy consumption for each component separately, contributing to the construction of our comprehensive model.

1) LOCAL COMPUTATION ENERGY CONSUMPTION

To enhance the computational efficiency of devices, we utilize DVFS technology. This innovative approach dynamically adjusts the CPU computing frequency f_{c} based on the computational demand of the system. The primary objective is to optimize and conserve electric power consumption across the devices, ensuring a more efficient use of resources. This adaptive adjustment allows the devices to operate at varying levels of performance, aligning with the real-time computational requirements and contributing to overall energy conservation. We adopt a CPU nonlinear power consumption model [29] as follows:

$$P_{\text{c}} = \kappa f_{\text{c}}^3. \quad (5)$$

Here, f_{c} represents the local CPU's computing frequency, and κ is a coefficient dependent on chip architecture. f_{c} spans the range from 0 to $f_{\text{c,max}}$ as it denotes the number of computational cycles per second. After receiving the λD bits data, the CPU utilizes the DVFS technology to flexibly adjust the computing frequency to execute, so the CPU executing time is

$$t_{\text{c}} = \frac{\alpha\lambda D}{f_{\text{c}}}. \quad (6)$$

The local computation phase is parallel to the communication phase. The FU only needs to execute the data within the specified time delay requirement $T_{\max}$, i.e., in one frame, the expected value of the local computational energy consumption always is

$$E_{\text{c}} = \alpha\lambda\kappa D f_{\text{c}}^2. \quad (7)$$

2) FS COMPUTATION ENERGY COST

Given that FS is another device with limited available computing resources, it also integrates DVFS technology. Consequently, its computational energy consumption model should align with that of the FU, i.e.

$$E_{\text{m}} = \alpha(1-\lambda)\kappa D f_{\text{m}}^2, \quad (8)$$

where f_{m} represents the computing frequency of FS and is limited as $0 \leq f_{\text{m}} \leq f_{\text{m,max}}$. It is worth noting that the FS can only start the computation when the transmission decoding is successful. Obviously, n = 1 indicates that the initial transmission is successful with probability $1-\varepsilon$, so the probability of the n^{th} computation is $\varepsilon^{n-1}(1-\varepsilon)$. Therefore, the expected computational energy consumption of the FS is

$$\begin{aligned}\bar{E}_{\text{m}} &= E_{\text{m},0} + \varepsilon E_{\text{m},0} + \ldots + \varepsilon^{N-1} E_{\text{m},0} \\ &= \sum_{n=1}^{N} \varepsilon^{n-1} E_{\text{m},0}, \end{aligned} \quad (9)$$

where the $E_{\mathrm{m},0}$ is computational energy consumption of the FS, same in a frame for each transmission.

3) ENERGY CONSUMPTION FOR SENDING/RECEIVING DATA PACKET

The FU uploads $(1-\lambda)D$ bits data to FS, and the energy consumption of the first transmitted data is defined as $E_{\mathrm{t},0} = t_{\mathrm{u}}P_{\mathrm{u}} + E_{\mathrm{t},1}$, where t_{u} represents the transmitting time of FU and $E_{\mathrm{t},1}$ is the energy cost for receiving data at FS. In the case of a failure in the $(\mathrm{n}-1)^{th}$ retransmission decoding attempt, the n^{th} transmission occurs. Hence, we can obtain the expected energy cost for transmitting data as follows:

$$\begin{aligned}\bar{E}_{\mathrm{t}} &= E_{\mathrm{t},0} + \varepsilon E_{\mathrm{t},0} + \ldots + \varepsilon^{N-1}E_{\mathrm{t},0} \\ &= \sum_{n=1}^{N} \varepsilon^{n-1}E_{\mathrm{t},0}. \end{aligned} \tag{10}$$

According to [27], the optimal transmission scheme for closed-loop communication in the FBL regime is to devote all the remaining time length in a frame to backhauling for better reliability. Specifically, the system allocates the remaining time length of this frame to perform a backhaul after completing data offloading and FS computation. Therefore the energy consumption of the backhaul is:

$$E_{\mathrm{r}} = P_{\mathrm{r}}t_{\mathrm{r}} + E'_{\mathrm{r},0}, \tag{11}$$

where $t_{\mathrm{r}} = T_s m_{\mathrm{back}}$ denotes reception time length of FU and $E'_{\mathrm{r},0}$ is the energy consumption for transmitting data at FS.

4) ENERGY COST FOR SENDING/RECEIVING NACK AT FU

In the FBL regime, the transmission is no longer reliable, and the FS may encounter an error while decoding the data packet from FS. When an error occurs, the FS sends a NACK to the FU to request a data retransmission. For a single transmission, the energy consumption for receiving a NACK at the FU can be exhibited as $E_{\mathrm{k},0} = t_{\mathrm{NK}}P_{\mathrm{r}} + E_{\mathrm{k},1}$, $E_{\mathrm{k},1}$ is the energy cost for receiving at FU. Note that for the initial transmission, no NACK is sent, i.e., $E_{\mathrm{k}} = 0$. The first NACK occurs only if the FS decodes incorrectly for the first time, i.e., $E_{\mathrm{k}}^{(1)} = \varepsilon E_{\mathrm{k},0}$. So, the expected energy consumption for sending NACK can be given by

$$\begin{aligned}\bar{E}_{\mathrm{k}} &= \varepsilon E_{\mathrm{k},0} + \varepsilon^2 E_{\mathrm{k},0} + \ldots + \varepsilon^{N} E_{\mathrm{k},0} \\ &= \sum_{n=1}^{N} \varepsilon^{n} E_{\mathrm{k},0}. \end{aligned} \tag{12}$$

Therefore, the total expected energy cost of the system can be given as

$$\bar{E}_{\mathrm{tot}} = \eta_k\bar{E}_{\mathrm{k}} + \eta_r E_{\mathrm{r}} + \eta_t\bar{E}_{\mathrm{t}} + \eta_c E_{\mathrm{c}} + \eta_m\bar{E}_{\mathrm{m}}. \tag{13}$$

Here, the energy consumption for transmitting NACK $\bar{E}_{\mathrm{k}}$, the energy consumption for transmitting data in the uplink $\bar{E}_{\mathrm{t}}$, and the computational energy consumption of the FS $\bar{E}_{\mathrm{m}}$ are correlated with the decoding error probability, whereas the local computational energy consumption of the FU E_{c} and the backhaul energy consumption E_{r} are independent of the decoding error probability ε. $\eta_j, j \in [k, r, t, c, m]$, is the cost factor of each component.

C. TOTAL DELAY

Because FBL coding results in systematic retransmissions, we define $n = 1$ as the initial transmission. Therefore, we can determine the time length of the n^{th} (re)transmission as $nmT_{\mathrm{s}} + (n-1)t_{\mathrm{NK}} + m_{\mathrm{back}}T_{\mathrm{s}}$ (which includes the time length when the FS is sent back to the FU after processing $t_{\mathrm{r}} = m_{\mathrm{back}}T_{\mathrm{s}}$). Additionally, taking into account the FS computation time, we can calculate the total execution time length of each offload as

$$D_{\mathrm{T}} = \max\left\{(n+1)t_{\mathrm{s}} + t_{\mathrm{d}}^{(n)} + (n-1)t_{\mathrm{NK}} + t_r, t_{\mathrm{c}}\right\}, \tag{14}$$

where t_{d} is the computational time length in the FS and satisfies $t_{\mathrm{d}} = \frac{\alpha(1-\lambda)D}{f_{\mathrm{m}}}$, with f_{m} representing the computing frequency of FS. Therefore, at a maximum of N retransmission attempts, the limiting execution time of the system should satisfy

$$\max\left\{(N+1)t_{\mathrm{s}} + t_{\mathrm{d}}^{(N)} + (N-1)t_{\mathrm{NK}} + t_{\mathrm{r}}, t_{\mathrm{c}}\right\} \le T_{\max}. \tag{15}$$

In addition, when the uplink transmission and computation is complete, all the remaining time length are used for the downlink backhaul operation, which can be calculated with $t_{\mathrm{r}} = T_{\max} - (N+1)t_{\mathrm{s}} + t_{\mathrm{d}}^{(N)} + (N-1)t_{\mathrm{NK}} = m_{\mathrm{back}}T_{\mathrm{s}}$. In particular, the transmission blocklength of the downlink is

$$m_{\mathrm{back}} = \left\lceil \frac{T_{\max} - (N+1)\,mT_{\mathrm{s}} + \frac{\alpha D(1-\lambda)}{f_{\mathrm{m}}} + (N-1)t_{\mathrm{NK}}}{T_{\mathrm{s}}} \right\rceil. \tag{16}$$

$\lceil . \rceil$ is the ceiling function.

III. PROBLEM FORMULATION

We expect to optimize the total expected energy consumption of the system by finding the optimal single transmission blocklength m, local computing frequency f_{c}, maximum tolerable number of retransmissions N, and offloading ratio λ. Note that the system reliability requirement is $\varepsilon_{\mathrm{tot}} \le \varepsilon_{\max}$, following (15), which stands for the ultra-low latency constraints. We have the following problem:

$$\min_{m,N,f_{\mathrm{c}},\lambda} \bar{E}_{\mathrm{tot}} \tag{17}$$

$$s.t.\ \varepsilon_{\mathrm{tot}} \le \varepsilon_{\max}, \tag{17a}$$

$$D_T \le T_{\max}, \quad n = N, \tag{17b}$$

$$0 \le \lambda \le 1, \tag{17c}$$

$$0 \le f_{\mathrm{c}} \le f_{\mathrm{c},\ \max}, \tag{17d}$$

$$N \in \mathbb{Z}. \tag{17e}$$

IV. OPTIMIZATION DESIGN

A. FEASIBLE DOMAIN ANALYSIS

Given the intricacy of the problem, our next step involves conducting a feasible domain analysis for each variable

to elucidate the interrelationships, thereby facilitating the optimization of the primary problem.

1) THE FEASIBLE DOMAIN ANALYSIS OF BLOCKLENGTH M

According to (3), the blocklength of a single transmission affects the probability of decoding error in each transmission and thus impacting the reliability of the system. Notably, a shorter blocklength leads to a higher probability of FS decoding error, rendering (17a) unfulfilled. Conversely, an excessively long blocklength, failing to satisfy $(N+1)t_s \leq T_{\max}$, also renders the system devoid of feasible solutions for problem (17). Referring to (17b), we derive

$$m_{\text{upper}} = \begin{cases} \left\lceil \dfrac{\frac{D\lambda\alpha}{f_{c,\min}} - (N-1)t_{\text{NK}} - \frac{(1-\lambda)D\alpha}{f_m}}{(N+1)T_s} \right\rceil, & \text{if } D_T = t_c, \\ \left\lceil \dfrac{T_{\max} - (N-1)t_{\text{NK}} - \frac{(1-\lambda)D\alpha}{f_m}}{(N+1)T_s} \right\rceil, & \text{otherwise}, \end{cases} \tag{18}$$

which implies the upper limit of blocklength and $\lceil . \rceil$ is the ceiling function. Moreover, considering (17a), we ascertain that $\varepsilon^{(n)} \leq \varepsilon_{\max}$. In conjunction with (3), where ε diminishes in m [22], i.e., the lower bound of blocklength should occur when m is too short to violate the reliability constraint.Thus, we can ascertain the lower bound on the blocklength m by:

$$m_{\text{low}} = \frac{H + \sqrt{H^2 - \frac{4C(\gamma)^2 D^2}{V(\gamma)^2}}}{2\frac{C(\gamma)^2}{V(\gamma)}}, \tag{19}$$

where $H = (\frac{Q^{-1}(\varepsilon_{\max})}{ln2})^2 + \frac{2DC(\gamma)}{V(\gamma)}$. Specifically, the longest blocklength must adhere to the delay constraint, while the shortest must maintain the reliability constraint. Consequently, for the problem to be solvable, the blocklength should adhere to $m \in (m_{\text{low}}, m_{\text{upper}})$.

2) THE FEASIBLE DOMAIN ANALYSIS OF THE OFFLOADING RATIO λ

The data offloading ratio λ, representing the ratio of locally offloaded data to the total data volume, must adhere to $\lambda \in [\lambda_{\text{low}}, \lambda_{\text{up}}]$. Specifically, the influence of λ on the system is primarily evident in the dynamic between offloading more data for local processing versus transmitting more for processing in the FS. This dynamic is inherently tied to CPU processing. Combining (6) and (14), to ensure a feasible solution for f_c, we deduce that ($\lambda \in [0, 1]$)

$$\lambda_{\text{up}} = \max\left\{\frac{f_{c,\max} T_{\max}}{D\alpha}, 1\right\}. \tag{20}$$

In addition, the offloading ratio correlates with the duration of time allocation: a higher offloading ratio implies more time allocated for local computation, whereas the system allocates more time to the communication phase conversely. Thus, we define $t_l = (N+1)t_s + t_d^{(N)} + (N-1)t_{\text{NK}} + t_r$, which should satisfy $t_l \leq T_{\max}$. We derive

$$\lambda_{\text{low}} = \min\left\{0, 1 - \frac{f_m\left(T_{\max} - (N-1)t_{\text{NK}}\right)}{D\alpha} + \frac{((N+1)t_s - t_r)}{D\alpha}\right\}. \tag{21}$$

3) THE FEASIBLE DOMAIN ANALYSIS OF DELAY REQUIREMENT T_{MAX}

The latency requirement at the FU should satisfy $T_{\max} \in [T_p, T_q]$. In other words, if the FU delay falls outside this range, a system update becomes necessary for a viable solution. Given that $\lambda_{\text{low}} \leq \lambda_{\text{up}}$, we can determine a feasible domain concerning the maximum delay as follows:

$$T_p = \frac{D\alpha + (N+1)t_s f_m + f_m(N-1)t_{\text{NK}} + f_m t_r}{f_m + f_{c,\max}}. \tag{22}$$

The upper bound of ($T_{\max}$ can be determined by considering two extreme scenarios: when all computations are performed locally and when all data is sent to the FS for computing, i.e.

$$T_q = \min\left\{\frac{D\alpha}{f_{c,\min}}, \frac{D(1-\lambda)\alpha}{f_m} + (N+1)t_s + (N-1)t_{\text{NK}} + t_r\right\}. \tag{23}$$

4) THE FEASIBLE DOMAIN ANALYSIS OF COMPUTATIONAL SPEED F_C

The minimum value of the local computing frequency is realized when tasks are offloaded locally, and the computation process is completed, i.e., $t_c \leq T_{\max}$. So we have

$$\frac{\lambda D\alpha}{T_{\max}} \leq f_c \leq f_{c,\max}, \tag{24}$$

which indicates that $f_{c,\text{optimal}}$ will occur at endpoint because of (7), i.e, the monotonically increasing properties of E_c with respect to f_c.

5) THE FEASIBLE DOMAIN ANALYSIS OF MAXIMUM TOLERABLE NUMBER OF RETRANSMISSION N

Retransmission occurs in the communication phase, we have $t_l \leq T_{\max}$, thus an upper bound on N can be given:

$$N_{\max} \leq \left\lfloor \frac{T_{\max} - t_d + t_{\text{NK}} - mT_s - t_r}{(mT_s + t_{\text{NK}})} \right\rfloor, \tag{25}$$

i.e., the number of (re)transmissions N should satisfy $1 \leq N \leq N_{\max}, \forall N \in \mathbf{Z}$. $\lfloor . \rfloor$ is the floor function.

B. OPTIMAL SOLUTION

1) SUBPROBLEM DECOMPOSITION AND VARIABLE CONVEXITY ANALYSIS

Primarily, (25) indicates that Problem (17) can be decomposed into an upper bound of $N_{\max}$ subproblems. Note that $N_{\max}$ is linearly related to m, λ and f_c. Therefore, the

Problem (17) can be expressed as

$$\min_{m,f_c,\lambda} \quad \bar{E}_{tot} \tag{26}$$

$$s.t.\ (17a), (17b), (17c), (17d), \tag{26a}$$

$$N_{\max} \le \lfloor \frac{T_{\max} - t_d + t_{NK} - mT_s - t_r}{(mT_s + t_{NK})} \rfloor. \tag{26e}$$

Secondly, according to [30], there is

$$\inf_{a,b} f(a,b) = \inf_{b} \widetilde{f}(b), \tag{27}$$

where the $\widetilde{f}$ is $\inf_a f(a,b)$. To determine the global minimum energy consumption, we aim to find the lower bound of $\bar{E}_{\text{tot}}$ with respect to each variable. Note that (7) tells us that $\bar{E}_{\text{tot}}$ is monotonically increasing with respect to f_c. Additionally, (24) represents the feasible domain of f_c based on the constraints. Thus, the locally optimal computing frequency can be expressed in closed form as follows:

$$f_c^* = \frac{\lambda D \alpha}{T_{\max}}, \tag{28}$$

which also is the lower bound for f_c.

In this way, we can reduce the problem to one about λ and m as

$$\min_{m,\lambda} \ \bar{E}_{tot} \tag{29}$$

$$s.t.\ (17a), (17c), (26e), \tag{29a}$$

$$t_l \le T_{\max}. \tag{29b}$$

The optimal solution of this problem can be obtained by using the Block Coordinate Descent (BCD) iterative method. Next, we will demonstrate and justify the BCD method. To start with, problems (29) can be simplified when the λ is fixed:

$$\min_{m} \quad \bar{E}_{tot} \tag{30}$$

$$s.t.\ (17a), (17c), (26e), \tag{30a}$$

$$t_l \le T_{\max}, \tag{30b}$$

$$\lambda = \lambda^{(k)}. \tag{30c}$$

Here, $\lambda^{(k)}$ represents the λ^* at the k^{th} iteration in Algorithm 1. Our next goal is to prove the convexity of $\bar{E}_{\text{tot}}$ and ε_{tot} in m, which can be obtained by the following Lemmas.

Lemma 1: The decoding error ε is convex in blocklength m.

Proof 1: We have $Q(\omega) = \int_{\omega}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$, where $\omega(m) = \sqrt{\frac{m}{V(\gamma)}} \left(C(\gamma) - \frac{(1-\lambda)D}{m} \right) \ln 2$. According to [22], we have

$$\frac{\partial^2 \varepsilon}{\partial m^2} = \frac{\partial^2 \varepsilon}{\partial \omega^2} \left(\frac{\partial \omega}{\partial m} \right)^2 + \frac{\partial \varepsilon}{\partial m} \frac{\partial^2 \omega}{\partial m^2} \ge 0, \tag{31}$$

which indicates that ε is convex in blocklength m. So we have

$$\frac{\partial^2 \varepsilon_{\text{tot}}}{\partial m^2} = N \frac{\partial^2 \varepsilon}{\partial m^2} \varepsilon^{N-1} + N(N-1) \left(\frac{\partial \varepsilon}{\partial m} \right)^2 \varepsilon^{N-2}, \tag{32}$$

since $\frac{\partial^2 \varepsilon}{\partial m^2} \ge 0$, we obtain $\frac{\partial^2 \varepsilon_{\text{tot}}}{\partial m^2} \ge 0$, i.e., the total decoding error is convex in m.

Since the total expected energy consumption is related to the decoding error probability, the convexity of the decoding error probability in the blocklength facilitates our analysis of the total expected energy consumption in m. Hence, we obtain the following lemma.

Lemma 2: Total expected energy consumption $\bar{E}_{\text{tot}}$ is convex in blocklength m.

Proof 2: The total energy consumption comprises four components, and we can establish the validity of Lemma 2 by demonstrating the convexity of each component in m. We observe that the second-order derivative of the locally expected computational energy consumption with respect to m is consistently equal to 0, i.e., E_c is liner in m. Meanwhile, the expected energy consumption $\bar{E}_m$ of the FS satisfies

$$\frac{\partial^2 \bar{E}_m}{\partial m^2} = \sum_{n=1}^{N} \frac{\partial^2 \varepsilon^{n-1}}{\partial m^2}. \tag{33}$$

Due to Lemma 1, we have $\frac{\partial^2 \bar{E}_m}{\partial m^2} \ge 0$, i.e., $\bar{E}_m$ is convex in m. So our next step is to proof the convexity of $\bar{E}_t$ and $\bar{E}_k$ in m. With (12), we have

$$\frac{\partial^2 \bar{E}_k}{\partial m^2} = \frac{\partial^2 \varepsilon}{\partial m^2} E_{k,0} + \sum_{n=1}^{N} (n-1)(n-2)\varepsilon^{n-3} \left(\frac{\partial \varepsilon}{\partial m} \right)^2 + (n-1)\varepsilon^{n-2} + \frac{\partial^2 \varepsilon}{\partial m^2} \ge 0, \tag{34}$$

which indicates that $\bar{E}_k$ is convex in m. Moreover, we have

$$\frac{\partial^2 \bar{E}_t}{\partial m^2} = P_u T_s \sum_{n=1}^{N} [\, (n-1)(n-2) m \varepsilon^{n-3} \frac{\partial^2 \varepsilon}{\partial m^2} + 2(n-1) \frac{\partial \varepsilon}{\partial m} \,] \ge 0, \tag{35}$$

so $\bar{E}_t$ is convex in m. As a result, total expected energy consumption $\bar{E}_{\text{tot}}$ is convex in m.

Thanks to Lemma 2, the optimal blocklength m^* can be give as

$$m^* = \begin{cases} m_{\text{upper}}, & \widehat{m} \ge m_{\text{upper}}, \\ \widehat{m}, & m_{\text{low}} \le \widehat{m} \le m_{\text{upper}}, \\ m_{\text{low}}, & \widehat{m} \le m_{\text{low}}, \end{cases} \tag{36}$$

where $\widehat{m}$ is minimal points of the curve, i.e., $\frac{\partial \bar{E}_{\text{tot}}(m)}{\partial m}|_{m=\widehat{m}} = 0$. So far, the objective function and the set of domains of definition of problem (30) are convex in m, i.e., when λ is fixed, (30) can be found to have an optimal solution using the BCD method.

Next, we demonstrate that the problem when m is fixed.

$$\min_{\lambda} \quad \bar{E}_{tot} \tag{37}$$

$$s.t.\ (17a), (17c), (26e). \tag{37a}$$

$$t_l \le T_{\max}, \tag{37b}$$

$$m = m^{(k)}. \tag{37c}$$

Similarly, $m^{(k)}$ represents the m^* at $k^{(\text{th})}$ iteration in Algorithm 1. The convexity of $\bar{E}_{\text{tot}}$ and ε_{tot} in λ can be obtained by the following Lemmas.

Lemma 3: The decoding error ε is convex in offloading ratio λ.

Proof 3: From (3), we have

$$\frac{\partial \omega}{\partial \lambda} = \ln 2 \frac{D}{\sqrt{mV(\gamma)}} \geq 0. \tag{38}$$

Hence we have

$$\frac{\partial^2 \varepsilon}{\partial \lambda^2} = \frac{\partial^2 \varepsilon}{\partial \omega^2}\left(\frac{\partial \omega}{\partial \lambda}\right)^2 + \frac{\partial \varepsilon}{\partial \omega}\frac{\partial^2 \omega}{\partial \lambda^2} \geq 0, \tag{39}$$

which tells us that

$$\frac{\partial^2 \varepsilon_{\text{tot}}}{\partial \lambda^2} = N\frac{\partial^2 \varepsilon}{\partial \lambda^2}\varepsilon^{N-1} + N(N-1)\left(\frac{\partial \varepsilon}{\partial \lambda}\right)^2 \varepsilon^{N-2} \geq 0. \tag{40}$$

So the total decoding error is convex in m and λ separately. For (37), the constraints form a convex set. Next, we prove the convexity of the objective function in λ by the following Lemma.

Lemma 4: Total energy consumption $\bar{E}_{\text{tot}}$ is convex in offloading ratio λ.

Proof 4: Since we have

$$\frac{\partial^2 \bar{E}_t}{\partial \lambda^2} = \frac{\partial^2 \bar{E}_{\text{k}}}{\partial \lambda^2} = \frac{\partial^2 \bar{E}_{\text{c}}}{\partial \lambda^2} \iff \frac{\partial^2 \varepsilon}{\partial \lambda^2} \geq 0, \tag{41}$$

so the convexity of $\bar{E}_{\text{tot}}$ in λ depends on (8), i.e.

$$\frac{\partial^2 \bar{E}_m}{\partial \lambda^2} = \frac{\partial^2 \varepsilon}{\partial \lambda^2}E_{\text{m},0} + 2\frac{\partial \varepsilon}{\partial \lambda}\left(-\alpha\kappa D f_{\text{m}}^2\right). \tag{42}$$

Moreover from (3), we have

$$\begin{aligned} \frac{\partial \omega}{\partial \lambda} &= \ln 2 \frac{D}{\sqrt{mV(\gamma)}} \geq 0, \\ \frac{\partial^2 \omega}{\partial \lambda^2} &= 0. \end{aligned} \tag{43}$$

Hence we have

$$\frac{\partial^2 \varepsilon}{\partial \lambda^2} = \frac{\partial^2 \varepsilon}{\partial \omega^2}\left(\frac{\partial \omega}{\partial \lambda}\right)^2 + \frac{\partial \varepsilon}{\partial \omega}\frac{\partial^2 \omega}{\partial \lambda^2} \geq 0, \tag{44}$$

where $\frac{\partial \varepsilon}{\partial \omega} \leq 0$, $\frac{\partial^2 \varepsilon}{\partial \omega^2} \geq 0$ according to [22]. Thus we have proved that ε is convex in λ and constrain (17a) is convex in λ. Based on these, there is $\frac{\partial^2 \bar{E}_m}{\partial \lambda^2} \geq 0$, which indicates $\bar{E}_m$ is convex in λ.

So far, we have demonstrated that $\bar{E}_{\text{tot}}$ and the constrains set are convex in λ. Hence, the problem (37) can be similarly approximated to the optimal solution by the BCD method.

Lemma 1- 4 reveals us that problems (30) and (37) are two separated convex problems, then problem (29) can be solved by using the BCD method to obtain the optimal blocklength m^* and offloading ratio λ^* simultaneously. In particular, the value of $m^{(k)}$ and $\lambda^{(k)}$ at our k^{th} iteration is the optimal solution (m^*, λ^*) in the $(k-1)^{th}$ iteration, which can be obtained by Algorithm 1.

2) OPTIMAL SOLUTION ANALYSIS BASED ON KKT METHOD

Although Algorithm 1 can efficiently approximate the optimal solution to problem (29), we still wish to investigate the relevant characteristics of the exact solutions. In this section, we investigate the features as well as the conditions of the optimal solutions with KKT method. First, we give the Lagrangian function as

$$\begin{aligned} \Gamma = &\bar{E}(\lambda^*, m^*) + \mu_1(\varepsilon_{\text{tot}} - \varepsilon_{\max}) + \mu_2(t_{\text{l}} - T_{\max}) \\ &+ \mu_3(N_{\max} - \lfloor \frac{T_{\max} - t_{\text{d}} + t_{\text{NK}} - mT_{\text{s}} - t_{\text{r}}}{(mT_{\text{s}} + t_{\text{NK}})} \rfloor) \\ &+ \mu_4\lambda - \mu_5(\lambda - 1), \end{aligned} \tag{45}$$

so the KKT can be given as

$$\begin{aligned} \nabla\Gamma(m, \lambda) &= 0, \\ \mu_1(\varepsilon_{\text{tot}} - \varepsilon_{\max}) &= 0, \\ \mu_2(t_{\text{l}} - T_{\max}) &= 0, \\ \mu_3\left(N_{\max} - \lfloor \frac{T_{\max} - t_{\text{d}} + t_{\text{NK}} - mT_{\text{s}} - t_{\text{r}}}{(mT_{\text{s}} + t_{\text{NK}})} \rfloor\right) &= 0, \\ \mu_4\lambda &= 0, \\ \mu_5(\lambda - 1) &= 0, \\ \mu_j \geq 05, \\ j \in (1, 5), \\ \text{(17a), (17b)}, \end{aligned} \tag{46}$$

where $\nabla\Gamma(m, \lambda)$ is

$$\begin{aligned} \frac{\partial \Gamma}{\partial m} &= \sum_{n=2}^{N}(n-1)\frac{\partial \varepsilon}{\partial m}\varepsilon^{n-2}\left(E_{\text{m},0} + E_{\text{t},0}\right) \\ &\quad + \sum_{n=1}^{N}\varepsilon^{n-1}\left(nE_{\text{k},0}\frac{\partial \varepsilon}{\partial m} + \varepsilon P_{\text{u}}T_s\right) + B \\ &= \frac{\partial \varepsilon}{\partial m}\sum_{n=1}^{N}\varepsilon^{n-1}\left(\varepsilon P_{\text{u}}T_{\text{s}} + n\left(E_{\text{m},0} + E_{\text{t},0} + E_{\text{k},0}\right)\right) + B, \end{aligned} \tag{47}$$

and

$$\begin{aligned} \frac{\partial \Gamma}{\partial \lambda} &= \frac{\partial \varepsilon}{\partial \lambda}\sum_{n=1}^{N}\varepsilon^{n-1}\left((n-1)E_{\text{m},0} - \alpha\kappa D f_{\text{m}}^2\right) + \alpha\kappa D f_{\text{c}}^2 \\ &\quad + \frac{\partial \varepsilon}{\partial \lambda}\sum_{n=1}^{N}n\varepsilon^{n-1}(E_{\text{t},0} + nE_{\text{k},0}) - \frac{\alpha\kappa D P_{\text{r}}}{f_{\text{m}}} + A \\ &= \frac{\partial \varepsilon}{\partial \lambda}\sum_{n=1}^{N}\varepsilon^{n-1}\left(n\left(E_{\text{m},0} + E_{\text{t},0} + E_{\text{k},0}\right) - E_{\text{m},0} - E_{\text{c},1}\right) \\ &\quad + \frac{\alpha\kappa D P_{\text{r}}}{f_{\text{m}}} + E_{\text{c},1} + A, \end{aligned} \tag{48}$$

where $B = \mu_1\frac{\partial \varepsilon_{\text{tot}}}{\partial m} + \mu_3 T_{\text{s}}\left(\lfloor \frac{T_{\max} + t_{\text{d}} + t_{\text{r}}}{(mT_{\text{s}} + t_{\text{NK}})^2} \rfloor\right)$ and $A = \mu_1\frac{\partial \varepsilon_{\text{tot}}}{\partial \lambda} - \mu_2\frac{\alpha\kappa D}{f_{\text{m}}} + \mu_3\lfloor \frac{\frac{\alpha\kappa D}{f_{\text{m}}}}{(mT_{\text{s}} + t_{\text{NK}})} \rfloor + \mu_4 - \mu_5$. $E_{\text{c},1}$ represents the energy

cost in local at $\lambda = 1$ according to (7). Interestingly, B reveals that μ_2 and the corresponding constraints have no effect on the optimal solution of m, i.e. when m is an optimal solution, (26e) has already been satisfied. (47) reveals that the partial derivatives of Γ in m depend only on μ_1 and μ_3, whose relationship can be demonstrated as:

$$\mu_1 = -\frac{1}{\frac{\partial \varepsilon_{\text{tot}}}{\partial m}} \left\{ \frac{\partial \varepsilon}{\partial m} \sum_{n=1}^{N} \varepsilon^{n-1} \left(\varepsilon P_{\text{u}} T_{\text{s}} + n E_{\text{t,m,k}} \right) + \mu_3 T_{\text{s}} \left\lfloor \frac{T_{max+t_{\text{d}}+t_{\text{r}}}}{(mT_{\text{s}} + t_{\text{NK}})^2} \right\rfloor \right\},$$

where μ_1 and μ_3 have a linear relationship. Therefore, the equation about μ_1 in (46) can be reformulated as:

$$-\frac{(\varepsilon_{\text{tot}} - \varepsilon_{\max})}{\frac{\partial \varepsilon_{\text{tot}}}{\partial m}} H_1 = 0, \tag{49}$$

where $H_1 = \left\{ \frac{\partial \varepsilon}{\partial m} \sum_{n=1}^{N} \varepsilon^{n-1} \left(\varepsilon P_{\text{u}} T_{\text{s}} + n E_{\text{t,m,k}} \right) + \mu_3 T_{\text{s}} \left\lfloor \frac{T_{\max}+t_{\text{d}}+t_{\text{r}}}{(mT_{\text{s}}+t_{\text{NK}})^2} \right\rfloor \right\}$. Moreover, at $\mu_1 = 0$, we can derive μ_3 as:

$$\mu_3(m|\mu_1 = 0) = -\frac{\partial \varepsilon}{\partial m} \sum_{n=1}^{N} \varepsilon^{n-1} \frac{\left(\varepsilon P_{\text{u}} T_{\text{s}} + n E_{\text{t,m,k}} \right)}{T_{\text{s}} \left\lfloor \frac{T_{\max}+t_{\text{d}}+t_{\text{r}}}{(mT_{\text{s}}+t_{\text{NK}})^2} \right\rfloor}. \tag{50}$$

Here, $E_{\text{t,m,k}} = \left(E_{\text{m,0}} + E_{\text{t,0}} + E_{\text{k,0}} \right)$. $\mu_1 = 0$ indicates that $\varepsilon_{\text{tot}} - \varepsilon_{\max} \neq 0$, signifying that ε_{tot} must satisfy the following two conditions: the reliability constraints are met, and the optimal solution is not on the boundary. If the reliability constraint (17a) is violated, there is no optimal solution.

In essence, finding an optimal solution at $\mu_1 = 0$ implies that the solution must adhere to the reliability constraint. Similarly, we can obtain μ_1 at $\mu_3 = 0$ as:

$$\mu_1(m|\mu_3 = 0) = -\frac{1}{\frac{\partial \varepsilon_{\text{tot}}}{\partial m}} \frac{\partial \varepsilon}{\partial m} \sum_{n=1}^{N} \varepsilon^{n-1} \left(\varepsilon P_{\text{u}} T_{\text{s}} + n \left(E_{\text{m,0}} + E_{\text{t,0}} + E_{\text{k,0}} \right) \right). \tag{51}$$

$\mu_3 = 0$ indicates that $N_{\max} - \lfloor \frac{T_{\max}-t_{\text{d}}+t_{\text{NK}}-mT_{\text{s}}-t_{\text{r}}}{(mT_{\text{s}}+t_{\text{NK}})} \rfloor \neq 0$, signifying that the maximum allowable number of retransmissions either exceeds the system limit or must meet performance criteria. In the case of $\mu_3 = 0$, finding the optimal solution implies that it must adhere to the retransmission requirement within the system's capacity.

Since the condition expressed in equation (49) relies on the relationship between μ_1 and μ_3 when subjected to the inverse substitution applied to the μ_1's KKT condition, the solution derived from equation (49) represents an optimal solution for m, nominated as m^+.

Specially, when $\mu_1 = \mu_3 = 0$, since we have

$$\frac{\partial \varepsilon}{\partial m} = -\frac{\ln 2 \left(C(\gamma)\sqrt{m} + (1-\lambda)D \right)}{2\sqrt{2\pi V(\gamma)} m} e^{-\frac{\omega^2}{2}}, \tag{52}$$

$$\frac{\partial \varepsilon}{\partial \lambda} = -\frac{D \ln 2}{\sqrt{2\pi m V(\gamma)}} e^{-\frac{\omega^2}{2}}, \tag{53}$$

Algorithm 1 Partial Offloading Optimal Energy With FBL Regime

```
Obtain the data amount D and latency requirement T_max;
if T_max ≤ T_q then
    Drop the application or Update the system;
else
    Initialize N*, Ē_tot,min;
    for N = 1:N_max do
        Initialize λ* and m* as λ^(0), m^(0);
        for k = 0, 1, 2, 3,... do
            repeat
                With given λ^(k), find m* = m^(k+1) to solve Problem (30);
                With given m^k, find λ* = λ^(k+1) to solve Problem (37);
                Update k = k + 1;
            until |m^(k+1) − m^(k)| ≤ δ and |λ^(k+1)-λ^(k)| ≤ ν;
        end for
        return m* = m^(k), λ* = λ^(k);
        Keep a record of Ē*_tot;
        if Ē*_tot ≤ Ē_tot,min then
            Update Optimal Ē_tot,min;
            Update N* = N^(k);
        end if
    end for
    Calculate f*_c by (28);
end if
```

where $\omega = \sqrt{\frac{m}{V(\gamma)}} \left(C(\gamma) - \frac{(1-\lambda)D}{m} \right) \ln 2$. We provide a form of m^* as

$$m^* = \lceil \frac{(1-\lambda)^2 D^2}{C(\gamma)^2} \rceil. \tag{54}$$

Here, $\mu_3 = \mu_1 = 0$ represents that when we can find an optimal solution, it must satisfy both the system retransmission limit and the delay constraint. Hence m^* is obtained by:

$$m^* = \begin{cases} \lceil \frac{(1-\lambda)^2 D^2}{C(\gamma)^2} \rceil, & \mu_1 = \mu_3 = 0, \\ m^+, & \text{otherwise.} \end{cases} \tag{55}$$

Moreover, considering that $(-\mu_2 \frac{\alpha\kappa D}{f_{\text{m}}} + \mu_3 \lfloor \frac{\frac{\alpha\kappa D}{f_{\text{m}}}}{(mT_{\text{s}}+t_{\text{NK}})} \rfloor + \mu_4 - \mu_5)$ remains constant and does not impact the optimal solution, λ^* is determined as the solution to the following equation

$$\frac{\partial \varepsilon}{\partial \lambda} \sum_{n=1}^{N} \varepsilon^{n-1} \left(n \left(E_{\text{m,0}} + E_{\text{t,0}} + E_{\text{k,0}} \right) - E_{\text{m,0}} - E_{\text{c,1}} \right) + \mu_1 \frac{\partial \varepsilon_{\text{tot}}}{\partial \lambda} = -H_2. \tag{56}$$

Here $H_2 = \frac{\alpha\kappa D P_{\text{r}}}{f_{\text{m}}} + E_{\text{c,1}} - \mu_2 \frac{\alpha\kappa D}{f_{\text{m}}} + \mu_3 \lfloor \frac{\frac{\alpha\kappa D}{f_{\text{m}}}}{(mT_{\text{s}}+t_{\text{NK}})} \rfloor + \mu_4 - \mu_5$. The functions affecting the position of λ^* are represented on the left side of the equation, while the right side holds the constant terms, i.e., the terms that do not influence the optimal solution's position.

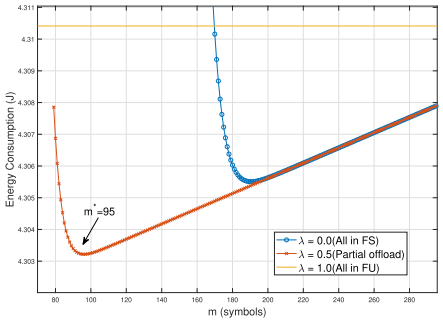

FIGURE 2. Energy cost in blocklength within different λ (The m^*, labelled in the figure, matches our analytical verification in (54)).

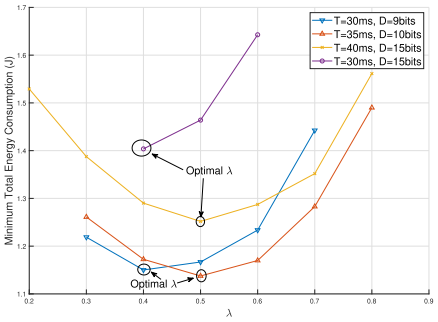

FIGURE 3. Energy cost vs λ within different data packet size and frame length T.

V. PARAMETER SETTING AND NUMERICAL SIMULATION

A. PARAMETER SETUP

First we consider setting the hardware constant κ of the CPU to 10^{-11}, aligning the energy consumption with the measurements in [31]. In addition, we set $\alpha = 10^4$ cycles/bit, to match the workload magnitude as reported in [32]. Apart from this, we constrain the frame length T to 45 ms with a symbol length $T_s = 0.03$ ms. The NACK-Transmission time length is $t_{NK} = 3$ ms. The path-loss model adheres to the NLOS path-loss model in [33], demonstrated as $\phi = 17.0 + 40.0\log_{10}(x)$, where x is the distance from FU to the FS. Furthermore, we set the bandwidth to $B = 5$ MHz, while the transmit power to $P_u = P_r = 0.3$ W and noise power to $\sigma_u^2 = \sigma_d^2 = -174$ dBm. The maximum computing frequency of the CPU, $f_{c,max}$, is set to 3×10^6 cycles/s, the FS server CPU has an idle arithmetic cap $f_{m,max}$ of 3.5×10^6 cycles/s. We set the cost factor as $\eta = 1$, $\eta \in [\eta_t, \eta_k, \eta_r, \eta_m, \eta_c]$. Additionally, the maximum allowed transmission error probability ε_{max} is 10^{-5} as a constraint for ultra-high reliability scenarios.

B. SIMULATION RESULTS

First we illustrate the variation of expected energy cost with respect to blocklength for different offloading ratio λ in Fig. 2. It's evident that the expected energy cost $\bar{E}_{tot}$ is convex in m, consistent with Lemma 2. Interestingly, the results also demonstrate that partial offloading yields lower expected energy consumption compared to full uploading to the FS or complete local computation. However, as the blocklength increases, the curve for full offloading to FS converges with the curve for partial offloading, i.e., when the length of a single transmission is long enough, partial offloading to local can be dispensed with in order to conserve the energy consumption of local mobile devices. In addition, the local computation energy consumption is independent of the transmission, so it has remained stable. Moreover, the optimal solution value derived from the curve in the context of partial offloading aligns with the analytical expression presented in (54). This convergence reinforces the fact that (54) serves as the analytical solution specifically tailored for the partial offloading scenario.

In contrast, we promptly demonstrate the variation of expected energy consumption in the offloading ratio λ for

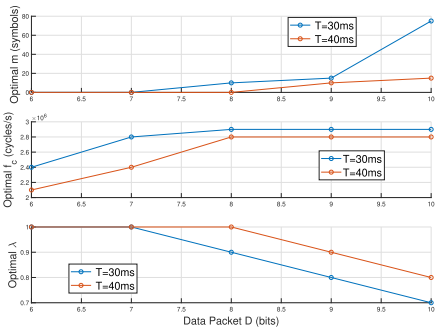

FIGURE 4. Optimal m, λ and f_c versus D at different frame length T.

different application packet sizes D and frame length T. The curves demonstrate the convexity of the energy consumption in the offloading ratio in line with our Lemma 4. The convexity demonstrated by Fig. 2 and Fig. 3 also provides a proof for the validity of our final optimisation algorithm 1. In addition, we find that the tighter the latency constraint and the larger the amount of data to be processed, the more the system tends to choose to offload to the FS, which is mainly due to the fact that the computational power of FS is slightly better than that of the local CPU and the transmission cost is lower than that of local computation when the latency constraint is strict. The feasible domain of λ is also closely related to the FU's state (D, T_{max}), which is consistent with our analysis of λ in (21).

In Fig. 4, we present a series of optimal solutions tailored to the system's diverse computational demands. Initially, for scenarios where the workload is below the local device's computational capacity (characterized by smaller values of D and α), the system opts for local computation. This choice is driven by the local CPU's ability to execute the entire task within the specified delay frame. During such frame, the blocklength remains at 0 symbol, indicating that the optimal solution is to compute all data locally. As the workload escalates, the system adapts by employing larger blocklength to facilitate the efficient offloading and uploading of tasks. Second, the local CPU tends to choose a slower computing frequency when the workload and latency requirements are relaxed, but as the task volume increases, the longer frame length allows the local CPU to use a faster computing frequency, completing the local computation faster and

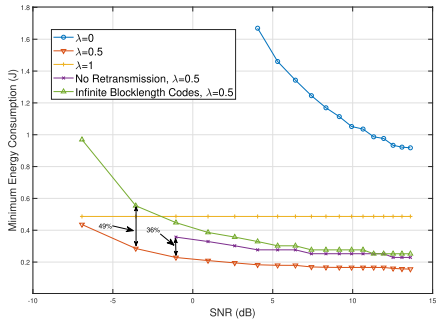

FIGURE 5. Optimal energy cost in Signal-to-noise ratio (*SNR*) within different λ and benchmarks (The figure demonstrates the benchmarks of IBL regime and No-retransmission).

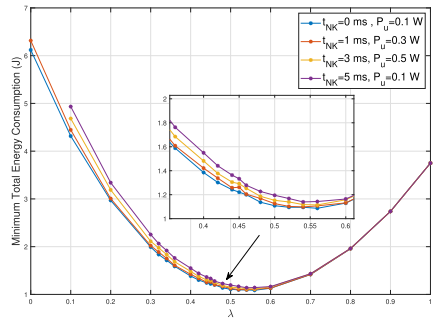

FIGURE 6. Optimal energy cost vs λ within different transmit power P_u and NACK-Transmission length t_{NK}.

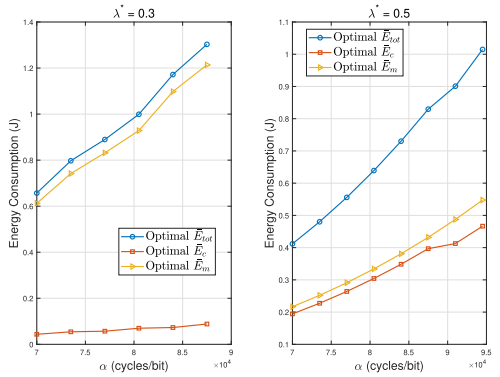

FIGURE 7. Parts of optimal energy cost versus workload ratio α in different optimal λ.

leaving the time for communication and FS computation. As the workload intensifies, the optimal configuration involves the local CPU operating at its maximum computing frequency, while concurrently offloading a portion of the processing load to the FS for auxiliary support. Moreover, the stringency of latency constraints directly influences the schedule of offloading, with tighter constraints prompting earlier offloading occurrences.

In Fig. 5, we delve into the correlation between the system's optimal energy consumption and the channel SNR across varying offloading ratio. Our analysis reveals that enhancing the SNR can effectively optimize the system's energy consumption whenever offloading operations are in play. However, this optimization is bounded, primarily due

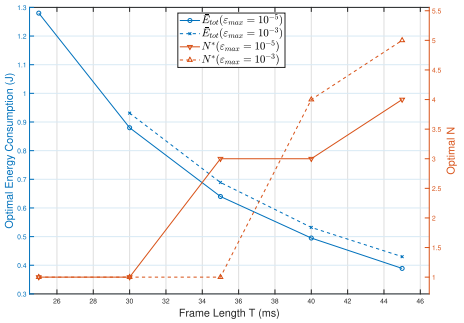

FIGURE 8. Optimal energy cost and optimal transmission number *N* vs frame length *T* within different allowed-error ε_{max}.

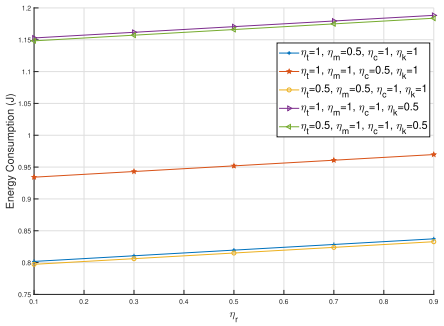

FIGURE 9. Variation of total consumption in η_t with different cost factor settings.

to the direct impact of the SNR on upload-back and NACK-transmission processes, while exerting minimal influence on the comparison between the FS and the local CPU, which serves as the predominant factor affecting energy consumption. In addition, the smaller the offloading ratio i.e. the more uploads to the FS, the more significant the role of the optimal energy consumption is influenced by the SNR γ. In addition, the retransmission mechanism can effectively reduce the expected energy consumption of the system, mainly because when the frame length is fixed, retransmission can lead to a more flexible blocklength allocation, resulting in less energy consumption per transmission. It should be noted that the no retransmission setting is also considered in our design, i.e., $N = 1$. Furthermore, in the context of the IBL assumption, the coding rate of the system aligns with the Shannon capacity. As per equation (3), the optimal blocklength is determined by $m = \frac{(1-\lambda)D}{C(\gamma)}$, with the error probability of transmission being $Q(0) = 0.5$. However, despite this optimal setup, the system still necessitates further retransmissions. Consequently, energy consumption escalates during transmission under the infinite blocklength assumption.

For a more quantitative analysis, we juxtapose our design against benchmarks. The results reveal a notable enhancement in the performance of the retransmission mechanism, showcasing a 49% improvement compared to scenarios without retransmission. Similarly, the performance of FBL coding exhibits a 36% enhancement in contrast to IBL coding.

TABLE 2. Notations.

Symbol	Definition	Symbol	Definition
T	Frame length	t_{NK}	The time length for NACK-transmission in downlink
$T_{\max}$	Maximum Allowable Latency	P_{r}	The power of transmitting NACK at FS
D	Data packet size	T_{s}	The time length of a symbol
α	Workload factor	t_{s}	Duration of a single transmission of uplink
C	Total computational cycles	t_{d}	The computational time length in the FS
m	Length of a blocklength	t_{r}	Duration of a single transmission of downlink
λ	Percentage of total data volume offloaded to local processing	n	Index of the number of transmissions
z_i	Gain of the uplink/downlink	t_{c}	Computation time of FU
γ_i	Signal-to-noise ratio of the uplink/downlink channel	f_c	CPU frequency of FU
ϕ_i	Channel path-loss of the uplink/downlink	f_m	CPU frequency of FS
P_{u}	The transmit power of the user	$f_{\mathrm{c,max}}$	The maximum available CPU frequency of FU
σ_i^2	The noise power of the uplink/downlink	$f_{\mathrm{m,max}}$	The maximum available CPU frequency of FS
ε	Probability of decoding error for a single transmission of the uplink	κ	Hardware constant of the CPU
$\varepsilon_{\mathrm{back}}$	Probability of decoding error for a single transmission of the downlink	r	Coding rate
$\varepsilon_{\mathrm{tot}}$	The total probability of the system decoding error	$C(\gamma)$	Shannon capacity of the channel
$\bar{E}_{\mathrm{t}}$	Energy consumption for transmitting data of FU	$V(\gamma)$	Channel dispersion of the channel
$\bar{E}_{\mathrm{r}}$	Energy consumption for transmitting data of FS	$\bar{E}_{\mathrm{k}}$	Energy consumption for transmitting NACK at FS
E_{c}	Energy consumption for computation of FU	E_{m}	Energy consumption for computation of FS
N	Number of retransmissions in a frame	$N_{\max}$	Maximum number of retransmissions for system
$E_{\mathrm{t,1}}$	The energy cost for receiving data at FS	$E_{\mathrm{t,0}}$	The initial energy cost for transmitting data at FU
$E_{\mathrm{k,1}}$	The energy cost for receiving NACK at FU	$E_{\mathrm{k,0}}$	The initial energy cost for transmitting NACK at FS
$\varepsilon_{\max}$	System reliability constraint	D_{T}	Length of full process execution time in a frame
m_{back}	The blocklength of the backhaul transmission	m_{upper}	The upper limit of blocklength
m_{low}	The lower bound of the blocklength	λ_{up}	The upper limit of the offloading ratio
m^+	Optimal blocklength derived for $\mu_1 \neq \mu_3 \neq 0$ in the KKT condition	$\widehat{m}$	The extremal point of $\bar{E}_{\mathrm{tot}}$ in m
λ_{low}	The lower bound of the offloading ratio	T_{p}	The lower limit of the frame length
T_{q}	The upper bound of the frame length	f_c^*	The locally optimal computational frequency
t_l	The communication time length	m^*	The optimal blocklength
λ^*	The optimal offloading ratio	η_j	The cost factor of the different energy cost, where $j \in \{t, k, r, m, c\}$

Moreover, we demonstrate curves of optimal energy consumption versus offloading ratio for different transmit power and NACK-Transmission time length in Fig. 6. Both the transmit power and the time length of the NACK-Transmission affect the value of the optimal energy consumption, with higher transmit power and longer NACK-Transmission time length corresponding to elevated optimal energy consumption values. Notably, the impact of these two factors differs more significantly when the offloading ratio λ is smaller. As λ increases, the distinctions in the changes of P_{u} and t_{NK} become less pronounced. This phenomenon can be elucidated by considering the direct impact of both P_{u} and t_{NK} on the communication transmission process. A smaller λ implies increased data

upload to the FS for processing, amplifying differences in transmission parameters. Conversely, as the system offloads more data for local processing, the diminishing data volume mitigates disparities in transmission characteristics. At $\lambda = 1$ (i.e., all data are computed locally) the curves will converge completely.

Furthermore, in Fig. 7, we analyse part of the optimal energy consumption versus the workload ratio within different optimal offloading ratio λ^* corresponding to each workload ratio α. Consistent with our expectations, computational energy consumption $\bar{E}_{\mathrm{c}}$, $\bar{E}_{\mathrm{m}}$ is the main factor affecting total energy consumption. In addition, when the optimal offloading ratio λ^* at this point is 0.5, i.e., the system offloads packets equally to the local and FS for processing and both have similar computational capability, $\bar{E}_{\mathrm{m}}$ and $\bar{E}_{\mathrm{c}}$ are close, which is consistent with our assumptions (7), (8) in Section II. Furthermore, the offloading ratio plays a crucial role in resource allocation, and despite the slightly superior computational power of the FS, the optimal solution does not advocate maximal data offloading to the FS. This observation underscores the significance of our work in determining optimal offloading strategies rather than pursuing maximal offloading to the FS.

Additionally, we demonstrate the impact curves of different cost factor settings on the optimal energy consumption of the system in Fig. 9. Since the total energy consumption is the weighted sum of the energy consumption of each component, varying the cost factor of each component does not affect the trend of the energy consumption, which is consistent with equation 13. An interesting observation is that the different settings of cost factor for computational energy consumption (η_c, η_m) and communication energy consumption (η_t, η_k, η_r) differ significantly, while the adjustment of their internal weights does not have a significant effect, which implies that in practice communication-dominated and computation-dominated networks will be optimised differently results.

Finally, in Fig. 8, we present curves depicting the optimal energy consumption and the optimal number of retransmissions within the system, varying with the frame length for different transmission-reliability requirements. Notably, as the frame length extends, the system's energy consumption diminishes, while concurrently witnessing an upsurge in the optimal number of retransmissions. This phenomenon can be elucidated by considering (15), where a fixed frame length necessitates shorter blocklength for individual transmissions with increased retransmissions. Therefore, when extending the frame length, the system, in a bid to reduce energy consumption, opts for more retransmissions, corresponding to shorter transmission blocklength. An intriguing observation is that under stringent transmission reliability requirements, denoted by $\varepsilon_{\max}$, the system must opt for longer transmission blocklength m to enhance decoding accuracy. Consequently, this choice results in heightened expected energy consumption and a decrease in the optimal transmission number N^*. Additionally, in scenarios where the frame length T is short, retransmissions are precluded to meet latency requirements.

VI. CONCLUSION

In this paper, we focus on partial offloading scenarios of fog computing networks equipped with DVFS technology. The emphasis is on optimizing the expected total energy consumption of the system under the FBL regime. By analyzing the variables, we transform the original problem into a bivariate convex optimization problem and propose the corresponding BCD algorithm. Specifically, we determine the explicit feasible domain of the system variables, establish the convexity of the expected energy consumption concerning the error rate in terms of the transmission blocklength, and offloading ratio, and analyse the form of the optimal solution using KKT conditions. Finally, we evaluate the proposed framework and algorithms through simulations, and we have found that our design is significantly improved compared to benchmarks. In addition, our work has significant potential for extension, particularly in determining the offloading ratio of each FU with the allocation strategy of the FBL codes in multi-FS-to-multi-FU scenarios. Moreover, based on our simplification of the original problem, the application of deep learning in similar models will be more convenient.

REFERENCES

[1] M. De Donno and N. Dragoni, "Combining AntibIoTic with fog computing: AntibIoTic 2.0," in *Proc. IEEE 3rd Int. Conf. Fog Edge Comput. (ICFEC)*, May 2019, pp. 1–6.

[2] G. Jin, "Review and case study of the impact of VR technology on Internet 3D game design," in *Proc. Int. Conf. Intell. Transp., Big Data Smart City (ICITBS)*, Mar. 2021, pp. 333–337.

[3] A. S. Al-Ahmad and H. Kahtan, "Cloud computing review: Features and issues," in *Proc. Int. Conf. Smart Comput. Electron. Enterprise (ICSCEE)*, Jul. 2018, pp. 1–5.

[4] A. Jain and D. S. Jat, "An edge computing paradigm for time-sensitive applications," in *Proc. 4th World Conf. Smart Trends Syst., Secur. Sustainability (WorldS)*, Jul. 2020, pp. 798–803.

[5] S. Lu, J. Lu, K. An, X. Wang, and Q. He, "Edge computing on IoT for machine signal processing and fault diagnosis: A review," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11093–11116, Aug. 2023.

[6] S. Lanka, T. Aung Win, and S. Eshan, "A review on edge computing and 5G in IoT: Architecture & applications," in *Proc. 5th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Dec. 2021, pp. 532–536.

[7] M. U. A. Siddiqui, H. Abumarshoud, L. Bariah, S. Muhaidat, M. A. Imran, and L. Mohjazi, "URLLC in beyond 5G and 6G networks: An interference management perspective," *IEEE Access*, vol. 11, pp. 54639–54663, 2023.

[8] Ravva. S. Sanketh, Y. MohanaRoopa, and Panati. V. N. Reddy, "A survey of fog computing: Fundamental, architecture, applications and challenges," in *Proc. 3rd Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Dec. 2019, pp. 512–516.

[9] R. Bajracharya, R. Shrestha, S. A. Hassan, H. Jung, R. I. Ansari, and M. Guizani, "Unlocking unlicensed band potential to enable URLLC in cloud robotics for ubiquitous IoT," *IEEE Netw.*, vol. 35, no. 5, pp. 107–113, Sep. 2021.

[10] D. Van Huynh, V.-D. Nguyen, S. R. Khosravirad, V. Sharma, O. A. Dobre, H. Shin, and T. Q. Duong, "URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7669–7682, Nov. 2022.

[11] T. Liu, J. Li, F. Shu, and Z. Han, "Optimal task allocation in vehicular fog networks requiring URLLC: An energy-aware perspective," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1879–1890, Jul. 2020.

[12] B. Cao, Z. Sun, J. Zhang, and Y. Gu, "Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3832–3840, Jun. 2021.

[13] R. Rosmaninho, D. Raposo, P. Rito, and S. Sargento, "Time constraints on vehicular edge computing: A performance analysis," in *Proc. NOMS - IEEE/IFIP Netw. Oper. Manage. Symp.*, May 2023, pp. 1–7.

[14] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[15] Y. Hu, J. Gross, and A. Schmeink, "On the capacity of relaying with finite blocklength," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1790–1794, Mar. 2016.

[16] Q. He, Y. Zhu, P. Zheng, Y. Hu, and A. Schmeink, "Multi-device low-latency IoT networks with blind retransmissions in the finite blocklength regime," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 12782–12795, Dec. 2021.

[17] Y. Yang, Y. Hu, and M. C. Gursoy, "Reliability-optimal designs in MEC networks with finite blocklength codes and outdated CSI: (Invited paper)," in *Proc. 17th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2021, pp. 1–6.

[18] J. Yao, Q. Zhang, and J. Qin, "Joint decoding in downlink NOMA systems with finite blocklength transmissions for ultrareliable low-latency tasks," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17705–17713, Sep. 2022.

[19] X. Zhang, J. Wang, and H. V. Poor, "Joint beamforming and trajectory optimizations for statistical delay and error-rate bounded QoS over MIMO-UAV/IRS-based 6G mobile edge computing networks using FBC," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2022, pp. 983–993.

[20] K. Chen, Y. Wang, J. Zhao, X. Wang, and Z. Fei, "URLLC-oriented joint power control and resource allocation in UAV-assisted networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10103–10116, Jun. 2021.

[21] Q. Wu, M. Cui, G. Zhang, F. Wang, Q. Wu, and X. Chu, "Latency minimization for UAV-enabled URLLC-based mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3298–3311, Apr. 2024.

[22] Y. Hu, Y. Zhu, M. C. Gursoy, and A. Schmeink, "SWIPT-enabled relaying in IoT networks operating with finite blocklength codes," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 1, pp. 74–88, Jan. 2019.

[23] G. Qu, "What is the limit of energy saving by dynamic voltage scaling?" in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2001, pp. 560–563.

[24] H. Zhao, J. Xu, P. Li, W. Feng, X. Xu, and Y. Yao, "Energy minimization partial task offloading with joint dynamic voltage scaling and transmission power control in fog computing," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9740–9751, Mar. 2024.

[25] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "Centralized and distributed architectures for energy and delay efficient fog network-based edge computing services," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 250–263, Mar. 2019.

[26] Y. Jiang, C. Wan, M. Tao, F.-C. Zheng, P. Zhu, X. Gao, and X. You, "Analysis and optimization of fog radio access networks with hybrid caching: Delay and energy efficiency," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 69–82, Jan. 2021.

[27] B. Han, Y. Zhu, M. Sun, V. Sciancalepore, Y. Hu, and H. D. Schotten, "CLARQ: A dynamic ARQ solution for ultra-high closed-loop reliability," *IEEE Trans. Wireless Commun.*, vol. 21, no. 1, pp. 280–294, Jan. 2022.

[28] A. V. Chandak and N. K. Ray, "Adaptive resource provisioning for smart home using fog computing," in *Proc. OITS Int. Conf. Inf. Technol. (OCIT)*, Dec. 2022, pp. 519–524.

[29] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[30] S. Boyd, L. Vandenberghe, and L. Faybusovich, "Convex optimization," *IEEE Trans. Autom. Control*, vol. 51, no. 11, p. 1859, Nov. 2006.

[31] T. L. Vinh, R. Pallavali, F. Houacine, and S. Bouzefrane, "Energy efficiency in mobile cloud computing architectures," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud Workshops (FiCloudW)*, Aug. 2016, pp. 326–331.

[32] Y. Zhu, Y. Hu, A. Schmeink, and J. Gross, "Energy minimization of mobile edge computing networks with HARQ in the finite blocklength regime," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7105–7120, Sep. 2022.

[33] Y. Corre, J. Stephan, and Y. Lostanlen, "Indoor-to-outdoor path-loss models for femtocell predictions," in *Proc. IEEE 22nd Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2011, pp. 824–828.

CHENHAO SHI is currently pursuing the bachelor's degree in communication engineering with the School of Electronics and Information, Wuhan University. His research interests include edge computing, deep learning, and ultra-high reliability and low-latency communications.

JINGRUI WEI is currently pursuing the bachelor's degree with the Department of Electronic Engineering, College of Electronic Information, Wuhan University. His research interests include ultra-high reliability and low latency communications, and wireless network design.

YAO ZHU (Member, IEEE) received the B.S. degree in electrical engineering from the University of Bremen, Bremen, Germany, in 2015, and the master's degree in information technology and computer engineering from RWTH Aachen University, Aachen, Germany, in 2018, where he is currently pursuing the Ph.D. degree with the ISEK Research Group. His research interests include ultra-reliable and low-latency communications, and mobile edge networks.

ANKE SCHMEINK (Senior Member, IEEE) received the Diploma degree in mathematics with a minor in medicine and the Ph.D. degree in electrical engineering and information technology from RWTH Aachen University, Germany, in 2002 and 2006, respectively. She was a Research Scientist with Philips Research before joining RWTH Aachen University. She spent several research visits with the University of Melbourne and the University of York. She is currently leading the Chair of Information Theory and Data Analytics, RWTH Aachen University. She is the co-author of more than 270 publications. Her research interests include information theory, machine learning, data analytics, and optimization with a focus on wireless communications and medical applications. She is an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. She is an Editor of the books *Big Data Analytics for Cyber-Physical Systems: Machine Learning for the Internet of Things* and *Smart Transportation: AI Enabled Mobility and Autonomous Driving*.