Computation Offloading and Trajectory Definition in UAV-MEC-based 5G Networks

Andson Balieiro, Jamilson Dantas, Elton Alves, Peterson Yoshioka, and Siba Narayan Swain .

Abstract— **UAV-based MEC (MEC-UAV) is a solution that provides computing resources close to users in 5G networks when ground-based infrastructure is not available, enabling computation offloading services. However, since the UAV is a battery-powered device, both the application (e.g., latency) and the UAV's energy consumption are impacted by computational and flight activities. Therefore, a careful analysis of the energy and performance tradeoff is required during the dimensioning and operation of UAV-MEC systems. This paper presents a MEC-UAV model that incorporates the data return and fusion at the user device, as well as considers errors that may occur during data transmission and processing at MEC-UAV, which have been overlooked in previous works. Utilizing the model, we conducted an analysis of energy consumption, latency, and the percentage of users served by MEC-UAV from the perspectives of computation offloading and trajectory optimization. The results emphasize the importance of considering these features when configuring the MEC-UAV system and defining supported applications.**

Keywords— **UAV, MEC, Computation Offloading, Trajectory Optimization, Modeling.**

I. INTRODUCTION

The Fifth Generation (5G) of mobile networks have been designed to encompass different applications organized into three categories: enhanced mobile broadband (eMBB), massive machine-type communication (mMTC), and ultra-reliable low latency communication (URLLC). To support such services, different technologies have been incorporated into 5G networks, including millimeter waves, software-defined networking, network function virtualization, and service-based architecture [1]. Among these technologies, Multi-Access Edge Computing (MEC) has been recognized as crucial for meeting applications with strict latency requirements (e.g., URLLC) or those that demand intensive computation but operate with limited resources, such as certain Internet of Things (IoT) ones [2], since MEC provides cloud resources (computing, storage, or connectivity) closer to the end-user.

MEC enables computational offloading applications in which users send their tasks to be processed at the edge servers, achieving not only low latency but also energy saving of user devices, which is crucial for battery-powered devices such as those used in IoT scenarios [3]. However, in disaster scenarios, remote or with high momentary demand locations, MEC services based on terrestrial infrastructures may not meet application requirements (e.g., latency), become unavailable (e.g., connectivity issues), or even be impractical to adopt (e.g., financial or physical deployment costs).

Andson Balieiro, Jamilson Dantas, Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Brasil, e-mail: {amb4, jrd}@cin.ufpe.br; Elton Alves, Universidade do Sul e Sudeste do Pará, Marabá, Brasil, e-mail: eltonalves@unifesspa.edu.br; Peterson Yoshioka, Instituto Federal de Ciência e Tecnologia do Pará (IFPA), Marabá, Brasil, e-mail: pmsyoshioka@gmail.com; Siba Narayan Swain, Indian Institute of Technology Dharwad, e-mail:sibaswain@iitdh.ac.in

In this context, embedding MEC into Unmanned Aerial Vehicles (MEC-UAV) is a prominent solution to overcome such problems, offering flexible and cost-effective deployment, as well as line-of-sight to users, which increases the channel gain and, consequently, reduces the users energy consumption in data offloading [2]. However, this combination brings challenges as the UAV is battery-powered and the energy consumption and user application (e.g., latency) are impacted by the computational offloading activities (computation and communication) and UAV flight (trajectory).

Several solutions for computation offloading and trajectory optimization in MEC-UAV have been proposed in the literature [3] [2] [4]. However, these solutions have relied on MEC-UAV models that overlook important aspects, such as the possibility of failure during data transmission and task processing at the UAV, as well as the data return and fusion at the user equipment (UE), which may lead to results that deviate from reality. This paper addresses these gaps by proposing an MEC-UAV model that considers the costs associated with the return of data processed in the UAV and its local fusion at the UE. Additionally, it admits transmission and processing failures in the MEC-UAV node. By adopting the model, we conducted an analysis of energy consumption, latency, and the percentage of users served by MEC-UAV from the perspectives of computation offloading and trajectory optimization. The results emphasize the importance of considering these features when configuring the MEC-UAV system and defining supported applications.

This paper is organized as follows. Section II presents studies that address computation offloading and trajectory optimization in MEC-UAV. Section III describes the computation and communication models, considering the features of communication and processing failures besides data return and fusion. The results are analyzed in Section V, and the conclusion and future directions are highlighted in Section VI.

II. RELATED WORK

The computation offloading and trajectory optimization problems in MEC-UAV have received attention from academia. For instance, [5], [3], and [2] address both problems. The first work proposes a heuristic that includes bit allocation and aims to minimize energy consumption while satisfying the latency constraints of applications and UAV energy budget. The second one offers a solution based on

Primal-Dual Particle Swarm Optimization and Deep Deterministic Policy Gradient to minimize the weighted sum of latency and energy consumption. The third study focuses on reducing global energy consumption through resource allocation optimization (e.g., bit allocation, bandwidth, computation, CPU frequency, and transmission power) and UAV trajectory. In [6], in turn, the computation offloading problem is addressed from the perspective of URLLC reliability and latency. The authors associate a failure probability with each UAV and focus on optimizing the number of attended applications while satisfying latency and reliability constraints. However, although all four solutions present interesting features and results, they neglect the possibility of transmission failures and the data return and fusion. Additionally, the first three solutions also overlook processing failures. In contrast, this paper proposes a model that addresses these gaps in the analysis of energy consumption and latency in computation offloading and trajectory optimization problems in MEC-UAV-based 5G networks.

III. System Model

We consider an MEC-UAV node v that moves from left to right in a 3-D space with limited dimensions $\{X_{size}, Y_{size}, H\}$, where, at time t, the user i and UAV v have positions given by the coordinates $Q_{user_i}(t) = \{x_i(t), y_i(t), 0\}$, with a null height, and $Q_{uav_v}(t) = \{X_v(t), Y_v(t), H\}$, with a fixed height H, respectively, similar to [2]. Additionally, at time t, the amount of data to be sent by the k users to the MEC-UAV is denoted as $L(t) = \{L_1(t), L_2(t), ..., L_k(t)\}$. Failures may occur during data transmission in both directions, uplink (from user to UAV) and downlink (from UAV to user),as well as in the data processing at the MEC-UAV. Furthermore, our system allows for data fusion returned from MEC-UAV. Sections III-A and III-B describe the communication and computation models of the MEC-UAV system, as well as the formulations for latency and energy consumption.

A. Communication Model

The user communicates with the MEC-UAV in both directions, sending data to be processed (uplink), and receiving processed data (downlink). We assume that the channel gain (h) between the user i and the MEC-UAV v at time t for both directions is given by Equation 1, which takes into account the channel gain at 1 meter reference distance (β_0) and the Euclidian distance between i and v.

$$h_{iv}(t) = \frac{\beta_0}{H^2 + \|Q_{uav_v}(t) - Q_{user_i}(t)\|^2} \tag{1}$$

Considering that the communication between the user i and the MEC-UAV v adopts B_{iv} Hz, the uplink and downlink data rates are given by Equations 2 and 3, where P_{user_i}, P_{uav_v} e δ_0 denote the transmission power of user, UAV, and noise, respectively.

$$r_{up_{iv}}(t) = B_{iv} log_2 \left(1 + \frac{P_{user_i} h_{iv}(t)}{\delta_0^2}\right) \tag{2}$$

$$r_{down_{iv}}(t) = B_{iv} log_2 \left(1 + \frac{P_{uav_v} h_{iv}(t)}{\delta_0^2}\right) \tag{3}$$

The uplink and downlink transmission latencies may be obtained by dividing the amount of data to be transmitted by the respective link data rate. However, the occurrence of failures during transmission may result in retransmissions, leading to higher latency and energy consumption. Let Pf_{up} and Pf_{down} be the probabilities of failure during uplink and downlink transmissions, respectively, and assuming the average case for the occurrence moment (in the middle of transmission period), the uplink and downlink tranmisison latencies between user i and MEC-UAV v are given by Equations 4 and 5. Here, $\alpha_i(t)$ and $\theta_i(t)$ denote the ratios of data $L_i(t)$ to be processed in MEC-UAV v and returned to user i for fusion, respectively. It is worth mentioning that the failure probabilities may be defined via transmission history analysis, signal-to-noise ratio, or probabilities distributions, for example.

$$T_{up_{iv}}(t) = \Delta T_{up_{iv}}(t) \left((1 - Pf_{up}) + \frac{3Pf_{up}}{2}\right) Pf_{up}, \tag{4}$$

$$T_{down_{iv}}(t) = \Delta T_{down_{iv}}(t) \left((1 - Pf_{down}) + \frac{3Pf_{down}}{2}\right) \tag{5}$$

$$\Delta T_{up_{iv}}(t) = \frac{\alpha_i(t) L_i(t)}{r_{up_{iv}}(t)}; \Delta T_{down_{iv}}(t) = \frac{\theta_i \alpha_i(t) L_i(t)}{r_{down_{iv}}(t)} \tag{6}$$

The energy consumption, in Joules (J), for uplink and downlink transmissions are obtained via 7, which consider the product of transmission power and latency.

$$E_{up_{iv}}(t) = P_{user_i} T_{up_{iv}}(t); E_{down_{iv}}(t) = P_{uav_v} T_{down_{iv}}(t) \tag{7}$$

B. Computation Model

There may be situations where processing data in the MEC-UAV requires more energy and time compared to the user device. Therefore, our model allows for two alternatives, local processing only, indicated by $\varphi_i(t) = 0$; and partial processing in both user device i and the MEC-UAV, denoted by $\varphi_i(t) = 1$.

1) Local Processing Only: In this case, it is assumed that the user device is free from failure [2] [3]. Thus, the processing latency is determined by the number of CPU cycles to process one bit (C_{user_i}) and the CPU frequency of the user device i (f_{user_i}), as described in Eq. 8.

$$T_{local_i}(t) = \frac{(1 - \varphi_i(t)) L_i(t) C_{user_i}}{f_{user_i}} \tag{8}$$

The energy spent in processing, in turn, is computed by using the user capacitance coefficient (K_{user_i}) and CPU frequency ($f_{user)_i}$), and the the processing latency [2] [3], as shown in Eq. 9.

$$E_{local_i}(t) = K_{user_i} (f_{user_i})^3 T_{local_i}(t) \tag{9}$$

2) Partial Processing: MEC-UAV and User: In this case, the data processing takes place in both the user device and the MEC-UAV node.

- **Processing at the UE side**: data processing on the user device i comprises two stages: processing of a parcel of the total data $L_i(t)$ (i.e., $\alpha_i(t)L_i(t)$) and merging them with the result that returns from the MEC-UAV v. Eqs. 10 and 11 denote the processing latency for these stages, respectively, while Eqs. 12 and 13 describe the energy consumption.

$$T_{user_i}(t) = \frac{(1-\alpha_i(t))L_i(t)C_{user_i}}{f_{user_i}}\varphi_i(t) \tag{10}$$

$$T_{fusao_i}(t) = \frac{\theta_i(t)\alpha_i(t)L_i(t))C_{user_i}}{f_{user_i}}\varphi_i(t) \tag{11}$$

$$E_{user_i}(t) = K_{user_i}(f_{user_i})^3 T_{user_i}(t) \tag{12}$$

$$E_{user_i}(t) = K_{user_i}(f_{user_i})^3 T_{fusao_i}(t) \tag{13}$$

- **Processing at the MEC-UAV side**: this proposed model also considers that failures may occur during the data processing on the MEC-UAV, which may be caused by hardware, software or non-optimized solutions [7]. It assumes the average case for the occurrence moment, which is halfway through the data amount to be processed. Eq. 14 defines the MEC-UAV processing latency, where Pf_{uav} and C_{uav_v} denote the MEC-UAV failure probability and C_{uav_v} the numbers of CPU cycles to process one bit on the MEC-UAV v, respectively. The failure probability may be obtained by analyzing the failure history and calculating the time between failures and repair times [8].

$$T_{uav_{iv}}(t) = \left[\frac{\alpha_i(t)\varphi_i(t)L_i(t)C_{uav_v}}{f_{uav_v}}\right][(1-Pf_{uav}) + \frac{3}{2}Pf_{uav}] \tag{14}$$

  The energy consumption to process data of the user i on the MEC-UAV v is given by Eq. 15, where K_{uav_v} and f_{uav_v} represent the capacitance coefficient and MEC-UAV processor clock frequency, respectively.

$$E_{uav_{iv}}(t) = K_{uav_v}(f_{uav_v})^3 T_{uav_{iv}}(t) \tag{15}$$

- **Total Energy Consumption and Latency**: Since the user data are processed in parallel on the UAV and the user device , the total latency is obtained by Eq. 16, which emphasizes that the data merging only happens after the reception of the result from the MEC-UAV.

$$T_{total_{iv}}(t) = max[T_{user_i}(t), T_{up_{iv}}(t) + T_{uav_{iv}}(t) + T_{down_{iv}}(t)] + T_{fusao_i}(t) \tag{16}$$

  The energy consumption is given by Eq. 17, which is the sum of the energy spent in each stage.

$$E_{total_{iv}}(t) = E_{user_i}(t) + E_{up_{iv}}(t) + E_{uav_{iv}}(t) + E_{down_{iv}}(t) + E_{fusao_i}(t) \tag{17}$$

IV. Computational Offloading and Trajectory Optimization Problems

The computational offloading (Eq. 18) and trajectory optimization problems (Eq. 19) are formulated to minimize the weighted sum of energy consumption and latency through the decision of each user i at each instant t to do offloading to the MEC-UAV v and the definition of the trajectory of the UAV at each step p, flying from a starting point to an ending point.

$$\begin{aligned} &minimize\ \omega\sum_{i=1}^{k} E_i(t) + (1-\omega)\sum_{i=1}^{k} T_i(t) \\ &subject\ to: Q_{uav_v}, Q_{user_i} \leq \{X_{size}, Y_{size}\}, \quad \forall i \in k, \\ &\varphi_i(t) \in \{0,1\}, \quad \forall i \in k, \\ &min[max(T_{total_i})]. \end{aligned} \tag{18}$$

$$\begin{aligned} &minimize\ \omega\sum_{p=1}^{\zeta} E_i(t,p) + (1-\omega)\sum_{p=1}^{\zeta} T_i(t,p) \\ &subject\ to: Q_{uav_v}, Q_{user_i} \leq \{X_{size}, Y_{size}\}, \quad \forall i \in k, \\ &Q_{uav_v}: Q_{uav}^{inicio} \to Q_{uav}^{fim}. \end{aligned} \tag{19}$$

$$\begin{aligned} T_i(t) &= (1-\varphi_i(t))T_{local_i}(t) + \varphi_i(t)T_{total_{iv}}(t) \\ E_i(t) &= (1-\varphi_i(t))E_{local_i}(t) + \varphi_i(t)E_{total_{iv}}(t) \end{aligned} \tag{20}$$

V. Numerical Results and Discussion

To address the problems presented in Section IV, we adopted the algorithms outlined in [3] while considering the the merging of data returned from MEC-UAV as well as accounting for processing and transmission failures. We conducted a comparative analysis of our results against those obtained using [3] in terms of latency, energy consumption, and the percentage of users served by MEC-UAV under different (θ) values for data return and error probabilities. Additionally, we adopted the the same input parameter values of [3], which are summarized in Table I.

The algorithms described in [3] employ Primal-Dual Particle Swarm Optimization and Deep Deterministic Policy Gradient to make decisions regarding computation offloading and optimization trajectory. Our work integrates these algorithms into the proposed model to analyze the energy consumption, response time, and the number of users served by the MEC-UAV. Our model takes into account crucial features that have been overlooked in prior research and that impact on the solution results. To highlight these impacts, we present the results in Sections V-A and V-B, which respectively analyze scenarios with (i) data return and merging, and (ii) failure occurrence. Additionally, Section V-C combines both scenarios. The average and total sum of the results are shown, considering ten runs of the algorithms, each with 1000 episodes. Results obtained using the model of [3] are denoted as "Gan" for comparison purposes.

Parameter	Value
Number of users (k)	10
Data size per user (L)	[1,10]Mbits
UAV height (H)	100 m
Channel Gain - 1m reference distance (β_0)	-50 dB
Bandwidth for each user(B_0)	10 MHz
User Transmission Power P_{user_i}	0,5 W
UAV Transmission Power (P_{uav_v})	0,6 W
UAV Noise Power(δ_0^2)	-70 dBm/Hz
User Capacitance Coefficient (K_{user_i})	10^{-27}
UAV Capacitance Coefficient (K_{uav_v})	10^{-28}
User CPU cycles per bit (C_{user_i})	800 cycles/bit
UAV CPU Cycles per bit (C_{uav_v})	1000 cycles/bit
User CPU Clock Frequency (f_{user_i})	1 GHz
MEC-UAV CPU Clock Frequency (f_{uav_v})	3 GHz
Position Limits ($\{X_{size}, Y_{size}\}$)	[100m, 100m]
Energy consumption x response time weight (ω)	0,75

TABLE I: Simulation Parameters

A. *Data Return and Merging*

In this failure-free scenario, we consider three different percentages of data returning from MEC-UAV to be merged on the user device, denoted as (θ), namely 10%, 30% and 60%. This values may correspond to different applications. Fig. 1 present the total sum for total latency (T_{total}) and total energy consumption (E_{total}), as defined by Eq. 16 and 17, respectively, for different θ values.It is noteworthy that as the amount of data returning to the user increases, both energy consumption and response time also increase. When θ raises from 10% to 30%, the energy consumption and increase by 0.44% and 14.07%, respectively. Comparing these results to the model of [3], we observe a difference of 7.3% and 31.38%, indicating the underestimation when [3] is employed, as id does not consider data return and fusion.

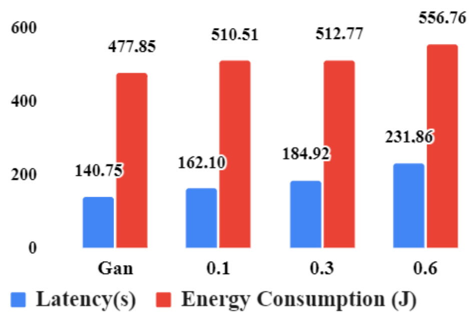
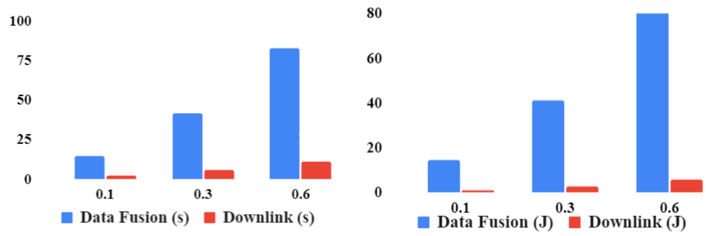


Fig. 1: Total Sum for Energy Consumption and Latency.

Figures 2a and 2b illustrate the individual effect of the data fusion and downlink transmission on energy consumption and latency. We may observe that data fusion has greater impact on both metrics. When θ increases from 30% to 60%, the energy consumption and latency experience a significant increase of 99.95% in the data fusion stage while in downlink transmission stage, they increase by 99.64% and 100%, respectively. These results highlight the potential burden on achieving low latency or the limitations it imposes on energy-constrained applications. Table II presents the percentage of users exclusively performed local processing.It is worth noting that for applications with a data return rate of at least 30%, the predominant decision is to process the data locally. This can potentially lead to underutilization of the MEC-UAV systems.



(a) Latency. (b) Energy Consumption

Fig. 2: Data Return and Fusion Impact for different θ values.

θ	0	0,1	0,3	0,6
% of users	13	26	53	62

TABLE II: Percentage of users with local processing

B. *Transmission and Processing Failures*

This scenario incorporates the possibility of failure occurrences during the transmission (user-UAV) and processing (MEC-UAV) stages, without any data return. We consider three transmission failure probabilities: 10%, 30%, and 50%, which are determined based on the distance between the user and the MEC-UAV, as shown in Table III, where d is the projection on the ground of the euclidean distance between the MEC-UAV and the user device. Similarly, the processing failure probability assumes three values, defined based on the amount of data to be processed, as illustrated in Table IV. It is important to note that such values are chosen to show the impact of these failure types on the system performance. Probability distributions, signal-to-noise ratio, regressive models [9], or analyst knowledge [6] may be used to determine appropriate values for these probabilities considering the target applications.

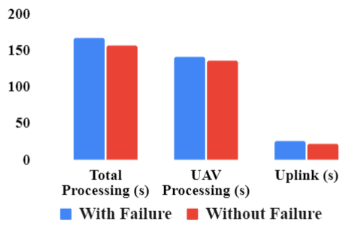
Distance	Pf_{up}
$0m \leq d < 20m$	0%
$20m \leq d < 80m$	10%
$80m \leq d < 90m$	30%
$90m \leq d \leq 141m$	50%

TABLE III: Transmission Failure Probability

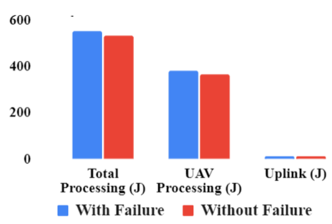
Data amount	Pf_{uav}
$0Mb \leq L < 3Mb$	0%
$3Mb \leq L < 7Mb$	10%
$7Mb \leq L < 9Mb$	30%
$9Mb \leq L \leq 10Mb$	50%

TABLE IV: Processing Failure Probability

Fig. 3a shows that the failure has a lower impact on energy consumption and latency compared to data fusion and return. When comparing a system with failure to a failure-free one, the latency different was 10.23s, with 5.87s and 4.36s caused by failures during processing and transmission. Analyzing the model of [3], we note that it underestimates the latency and energy consumption in 21.35s (15.17%) and 32.66J (6.83%), respectively, which may compromise the system operation and the application support. Additionally, out of the 23% of users not served by the MEC-UAV, 23% opted to process data locally.



(a) Latency.

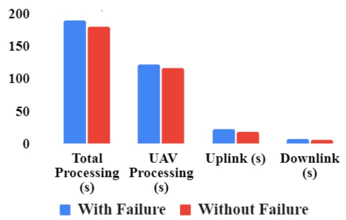


(b) Energy.

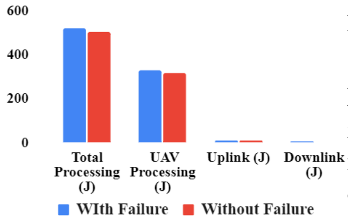
Fig. 3: Latency and Energy Consumption.

C. Failure and Data Return and Fusion

This scenario comprises transmission (both directions) and processing (MEC-UAV) failures along with data merging. We adopted the failures probabilities shown in Table III and IV, with a (θ) value of 30%. In Figs.4a and 4b, it can be observed that the downlink tranmission resulted in a 21.15% increase in both metrics, while the downlink one caused a similar increment of 21.10%. When considering the failure processing, there was an addition of 4.00% in the metrics. Overall, total result showed a uniform elevation of 5.34% in both metrics.



(a) Latency.



(b) Energy.

Fig. 4: Consumption for both features

Fig. 5 compare all components that contribute to the metrics. It is noted that the data fusion accounts for 21.25% and 7.82% of the total latency and energy consumption, respectively. This energy consumption is attributed to the user equipment (UE), which may lead to faster battery discharge. Table V presents average results, where the downlink transmission contributes with 660 ms and 0.33J for the energy and latency, respectively, while data fusion contributes 4.06s and 4.06J, which may significantly impact certain applications (e.g., URLLC services [10]).

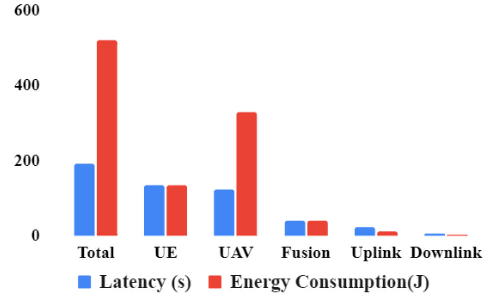


Fig. 5: Average Latency and Energy Consumption.

Table. VI presents the percentage of users who processed data locally. It is noteworth that the data return has a higher impact than the failure events. The downlink with all features and θ equals 0.3 showed only a 3% increase compared to the case without transmission failures.

	Total	UE	UAV	Fusion	*Up*	*Down*
Latency	19,10	13,53	12,18	4,06	2,2	0,66
Energy Cons.(J)	51,91	13,53	32,89	4,06	1,1	0,33

TABLE V: Average Latency and Energy Consumption

Proposal	% of users
Gan	13%
Downlink with $\theta = 0.1$	26%
Downlink with $\theta = 0.3$	53%
Downlink with $\theta = 0.6$	62%
Transmission and Processing Probs.	23%
All proposals with $\theta = 0.3$	59%

TABLE VI: Percentage of users who processed data locally

VI. Conclusion

This work proposed a 5G UAV-MEC model that incorporates processing and transmission failures as well as data return and fusion data, to analyze computation offloading and trajectory optimization problems. The results showed the significant impact of data fusion and return on energy consumption and latency, with the UE playing a crucial role in this process. The proposed model aims to facilitate the desing of solutions for these problems, yielding more realistic results, and assisting service operators in the analysis and dimensioning of UAV-MEC systems and supported applications. Future directions includes incorporating failure probabilities associated with modulation and coding schemes and transport block size, exploring computation offloading with multiple UAVs, and considering the energy consumption of the UAV flight.

References

[1] Hasneen, J. and Sadique, K. M., “A survey on 5g architecture and security scopes in sdn and nfv. In Iyer, B., Ghosh, D., and Balas, V. E., editors, Applied Information Processing Systems," *Singapore. Springer Singapore*,pp. 447–460, 2022.

[2] Qin, X., Song, Z., Hao, Y., and Sun, X., “Joint resource allocation and trajectory optimization for multi-uav-assisted multi-access mobile edge computing," *IEEE Wireless Communications Letters*, 10(7):1400–1404, 2021.

[3] Gan, Y. and He, Y., “Trajectory optimization and computing offloading strategy in uav-assisted mec system," *Computing, Communications and IoT Applications (ComComAp)*, pp. 132–137, 2021.

[4] Liu, J. and Zhang, Q., “Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, v. 6, pp. 12825—12837, 2018.

[5] Xiong, J., Guo, H., and Liu, J., “Task offloading in uav-aided edge computing: Bit allocation and trajectory optimization," *IEEE Communications Letters*, 23(3):538–541, 2019.

[6] Haber, E. E., Alameddine, H. A., Assi, C., and Sharafeddine, S., “A reliability-aware computation offloading solution via uav-mounted cloudlets," *IEEE 8th International Conference on Cloud Networking (CloudNet)*, pp. 1–6, 2019.

[7] Teng, X., Pham, H., and Jeske, D. R. “Reliability modeling of hardware and software interactions, and its applications," *IEEE Trans. on Reliability*, 55:571–577, 2006.

[8] Zhang, Q., Zhani, M. F., Jabri, M., and Boutaba, R., “Venice: Reliable virtual data center embedding in clouds.," *In IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 289-–297, 2014.

[9] Boutiba, K., Bagaa, M., and Ksentini, A. (2021) “Radio link failure prediction in 5g networks," *IEEE Global Communications Conference (GLOBECOM)*, 2021.

[10] El-Moghazi, M. and Whalley, J. , “the itu imt-2020 standardization: lessons from 5g and future perspectives for 6g," *Journal of Information Policy*, 2022.