Enhancing Service Continuity in Non-Terrestrial Networks via Multi-Connectivity Offloading

Yekaterina Sadovaya, *Graduate Student Member, IEEE*, Olga Vikhrova, *Member, IEEE*, Sergey Andreev, *Senior Member, IEEE*, and Halim Yanikomeroglu, *Fellow, IEEE*

***Abstract*— Non-terrestrial networks (NTNs) have recently emerged as a promising paradigm for computation-intensive six-generation (6G) applications, which may range from augmented reality to disaster relief. Moreover, NTNs can cater to uninterrupted connectivity needs in both rural and urban areas. In urban settings, uncrewed aerial vehicles (UAVs) and high-altitude platform station (HAPS) play crucial roles in supporting delay-sensitive computation applications for terrestrial users when terrestrial networks face limitations. Given the emerging interest in multi-connectivity for NTNs, this letter investigates UAV- and HAPS-assisted multi-connectivity computation offloading in urban areas. Specifically, we propose two novel multi-connectivity offloading strategies to improve the probability of timely task computation, along with a framework for optimizing the corresponding offloading probabilities onto HAPS and UAVs. Our results demonstrate that utilizing multi-connectivity in NTN-assisted offloading can achieve a 75% reduction in task computation delay as compared to scenarios with no offloading.**

***Index Terms*— HAPS, UAV, NTN, multi-connectivity, MEC, offloading.**

I. Introduction

MOBILITY of user devices causes spatial and temporal demand fluctuations across terrestrial networks. To prevent service outages, mobile network operators (MNOs) often over-provision by increasing the density of their terrestrial base stations (BSs). These BSs are strategically clustered in urban areas likely to experience peak demand. However, the densification significantly raises both capital and operational expenditures for terrestrial networks and the network's energy consumption due to the underutilization of BSs when demand is low. While various network energy saving techniques have been developed to address this issue, terrestrial densification remains unsustainable in the long run. This problem can be mitigated by integrating non-terrestrial network (NTN) with existing radio access networks (RANs). High altitude platform station (HAPS) and uncrewed aerial vehicle (UAV) are particularly attractive to this aim as they can be deployed *on-demand* to quickly respond to surges in terrestrial network load [1].

In addition to providing on-demand connectivity, NTNs can host mobile edge computing (MEC) servers for processing and analyzing information from power-constrained devices, thereby facilitating the transition toward safer and smarter environments [2]. NTNs can support MEC by allowing user equipment (UE) and Internet of Things (IoT) devices to offload their computationally intensive tasks such as object detection, recognition, tracking, or trajectory prediction for, e.g., urban augmentation, smart city, and public safety applications. Processing these tasks on the device side is often restricted by its small battery capacity. A critical requirement for these applications is the support for real-time latency to ensure timely decision making. Both HAPS and UAVs, as part of an NTN, have the potential for such applications.

HAPS systems are typically deployed at altitudes of 20 km, which reduces propagation latency to less than 1 ms. They have a large enough payload to support a high capacity MEC on board and can be equipped with powerful energy sources, including solar and wind energy converters. Moreover, the atmospheric temperature at HAPS's operating altitudes helps save energy for cooling and allows for scaling up the computation capacity. Different aspects of HAPS-aided computation offloading have been studied including joint offloading and caching for improved offloading delay [3], as well as joint offloading and resource allocation for reduced UE energy consumption [4]. However, as pointed out in [1], aggregating all the load on a single HAPS may result in congestion and significant performance degradation. This issue can be addressed via offloading diversity, by introducing less powerful but more adaptive UAV-aided MEC [5]. Cooperation between HAPS and UAVs alleviates limited computation resources and endurance time of UAVs, thereby satisfying stringent application requirements.

Another approach for enhancing the offloading diversity is through *multi-connectivity*, which enables devices to connect to multiple BSs simultaneously. Recent works on multi-connectivity in NTNs have demonstrated promising results for load balancing [6] and enhanced service continuity [7]. While multi-connectivity is standardized and widely used for terrestrial networks, its implementation in NTN presents several challenges [8], one of which is efficient task and traffic steering between nodes.

To the best of our knowledge, this is the first study of multi-connectivity offloading that addresses the computation task steering challenge in NTNs. In this letter, we benchmark

Manuscript received 2 July 2024; accepted 18 July 2024. Date of publication 29 July 2024; date of current version 11 October 2024. This work was supported by the Research Council of Finland (Projects ALL-ON, ECO-NEWS, SOLID, and RADIANT). The associate editor coordinating the review of this letter and approving it for publication was M. Elhattab. *(Corresponding author: Yekaterina Sadovaya.)*

Yekaterina Sadovaya and Olga Vikhrova are with the Unit of Electrical Engineering, Tampere University, 33014 Tampere, Finland (e-mail: yekaterina.sadovaya@tuni.fi; olga.vikhrova@tuni.fi).

Sergey Andreev is with the Unit of Electrical Engineering, Tampere University, 33014 Tampere, Finland, and also with the Department of Telecommunications, Brno University of Technology, 601 90 Brno, Czech Republic (e-mail: sergey.andreev@tuni.fi).

Halim Yanikomeroglu is with the Non-Terrestrial Networks (NTN) Laboratory, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: halim@sce.carleton.ca).

Digital Object Identifier 10.1109/LCOMM.2024.3434400

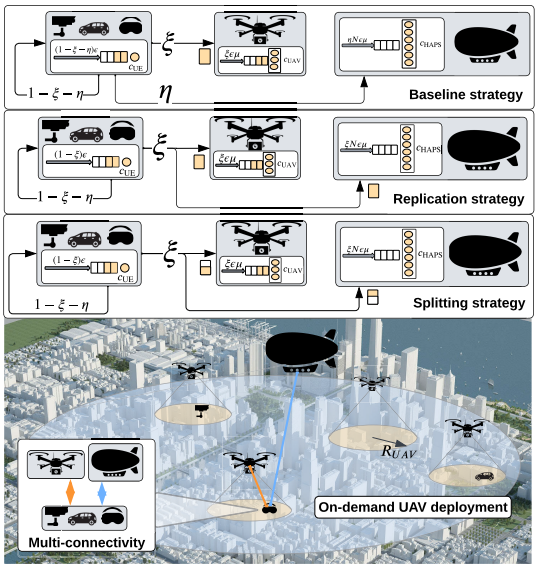

Fig. 1. Considered NTN deployment with HAPS- and UAV-assisted MEC.

two proposed multi-connectivity-specific offloading strategies against a baseline approach from [9] adapted for fair comparison. All the strategies are optimized to maximize the probability of in-time task computation based on the developed analytical model, evaluated, and compared through extensive simulations. Our results demonstrate that multi-connectivity offloading improves in-time computation of UE tasks at a marginal increase in UE energy consumption.

II. System Model

A. Deployment and System Assumptions

We consider an NTN deployment featuring a HAPS and N UAVs, which provide MEC services on demand [10] to terrestrial UEs with delay-aware and compute-intensive applications when the capacity of terrestrial RANs is insufficient. As illustrated in Fig. 1, the area of interest is modeled as a disk with a radius R_{HAPS}, which corresponds to the coverage area of the HAPS. The on-demand UAV coverage area is modeled by a disk with a radius R_{UAV}, which depends on the deployment site's environment. Our targeted UEs, being served by the NTN, are assumed to be uniformly distributed within the coverage area of the UAVs, hence ensuring that each UE is within the coverage of at least one UAV. The HAPS is stationary and accessible to any of the considered UEs.

UEs select sensory perception of the environment according to the generate-at-will model [11] with intensity ϵ. The latter means that the samples are generated at the UEs according to a Poisson process. The sensory information is represented as stacked video frames, which are processed as a batch, while the batches are of the fixed size n [12]. Each batch requires a constant computational load C for, e.g., object detection and recognition task, which has to be executed within a deadline t^*. This deadline may correspond to the interval between two consecutive environment samplings.

We assume that UEs as well as MEC servers at the HAPS and UAVs have different numbers of central processing units (CPUs) and CPU frequencies. Specifically, each UE is equipped with a single CPU [12], while HAPS and UAVs have c_{UAV} and c_{HAPS} CPUs onboard, respectively [9]. The CPU frequencies at UEs, UAVs, and HAPS are denoted by C_{UE}, C_{UAV}, and C_{HAPS}, respectively, being measured in GFLOPs. The task processing time, consequently, varies across the computing nodes and can be given as $D_{\text{UE}} = C/C_{\text{UE}}$, $D_{\text{UAV}} = C/C_{\text{UAV}}$, and $D_{\text{HAPS}} = C/C_{\text{HAPS}}$.

UE can compute its tasks locally or offload them onto UAV and/or HAPS. We consider a computation offloading strategy from [9] as a baseline, by extending it to include the capability of task offloading to UAVs. In this baseline strategy, UEs offload their tasks to the associated UAV with probability ξ, to the HAPS with probability η, or compute them locally. We also propose replication and splitting offloading strategies that leverage multi-connectivity capabilities at the UEs. The replication strategy involves the UE sending task replicas to both HAPS and UAV, and the splitting strategy involves splitting the tasks equally between HAPS and UAV. In both cases, the probability of offloading is denoted by ξ.

The time instances of task generation are identical and independent across all UEs. When a new task is generated, a UE immediately makes an offloading decision, which results in the task arrival at the HAPS and UAVs being a superposition of several thinned Poisson processes. Let μ be the average number of UEs within the coverage of a UAV. Under the baseline strategy, tasks arrive at this UAV with intensity $\xi\mu\epsilon$ and at the HAPS with intensity $\eta N\mu\epsilon$. For the replication and splitting strategies, the intensities of task arrivals at the HAPS and UAV are $\xi\mu\epsilon$ and $\xi N\mu\epsilon$, respectively. Although the average number of tasks offloaded onto UAVs and HAPS is the same for these strategies, the payload differs because the tasks are divided in the splitting strategy.

B. Transmission Delay

The signal-to-noise ratio (SNR) between a transmitter and a receiver in the considered system is given as

$$\Gamma = \frac{P_{\text{TX}}(G/N)_{\text{RX}}}{kWL}, \tag{1}$$

where P_{TX} is the effective radiated power, $(G/N)_{\text{RX}}$ is the receiver antenna-gain-to-noise-temperature, L is the path loss in NTN given by [12], k is the Boltzmann constant, and W is the system bandwidth.

Let $\Gamma^{\text{UL}}_{\text{UAV}}$, $\Gamma^{\text{DL}}_{\text{UAV}}$, $\Gamma^{\text{UL}}_{\text{HAPS}}$, and $\Gamma^{\text{DL}}_{\text{HAPS}}$ be the SNR for uplink (UL) and downlink (DL) communication channels between UE, UAV, and HAPS according to the deployment. The corresponding link capacity therefore yields

$$R^i_j = W \log_2\left(1 + \Gamma^i_j\right), \tag{2}$$

where $i \in \{\text{UL}, \text{DL}\}$ and $j \in \{\text{UAV}, \text{HAPS}\}$.

The communication delays between a UE its associated UAV or the HAPS in the cases of the baseline and replication strategies can be expressed as $T^i_j = n/R^i_j$. For the splitting strategy, the delay is $T^i_j = \delta_j n/R^i_j$, where δ_j is the task splitting ratio such that $\delta_{\text{UAV}} + \delta_{\text{HAPS}} = 1$.

C. Propagation Delay

Due to a significant difference in the deployment altitudes between HAPS and UAVs, the propagation delay to the HAPS may impact the overall communication delay, thereby influencing the choice of an offloading strategy. Therefore, we include the propagation delay $\tau = d/c_l$ into the computation of the overall task compute delay, where d is the average distance between UEs and HAPS and c_l is the speed of light.

D. Task Compute Delay

Upon the UE offloading decision, tasks arrive at a shared compute queue and are processed according to the first come first served (FCFS) discipline. If a newly arrived task finds an available CPU, it immediately starts being processed; otherwise, it waits in the queue [13].

1) Onboard Compute: Delay of local task compute T_{UE} includes waiting and processing times at UE's CPU

$$T_{\text{UE}} = W_{\text{UE}} + D_{\text{UE}}, \tag{3}$$

where W_{UE} is the task waiting time at the UE's queue and D_{UE} is the task processing time. As D_{UE} is deterministic and tasks arrive according to the Poisson process, the waiting time can be modeled according to the $M/D/1$ queue [13]. The cumulative distribution function (CDF) of T_{UE} is given by

$$\mathbb{P}\{T_{\text{UE}} \le t\} = F_{W_{\text{UE}}}(t - D_{\text{UE}})u(t - D_{\text{UE}}), \tag{4}$$

where $u(\cdot)$ is the Heaviside step function, $F_{W_{\text{UE}}}(t)$ is the CDF of the waiting time given by (9) for $D = D_{\text{UE}}$, $\lambda = (1-\xi-\eta)\epsilon$ for the baseline strategy and $\lambda = (1-\xi)\epsilon$ for the replication and splitting strategies.

2) UAV-Assisted Compute: Task offloading to UAVs requires both UL and DL transmissions, which may experience different delays. Therefore, the overall compute delay is given as follows:

$$T_{\text{UAV}} = T^{\text{UL}}_{\text{UAV}} + T^{\text{DL}}_{\text{UAV}} + W_{\text{UAV}} + D_{\text{UAV}}, \tag{5}$$

where $T^{\text{UL}}_{\text{UAV}}$ and $T^{\text{DL}}_{\text{UAV}}$ are UL and DL transmission delays, D_{UAV} is the processing delay at a UAV. Since D_{UAV} is deterministic, tasks arrive according to the Poisson process, and UAV is equipped with several CPUs, the waiting time can be modeled as the $M/D/c$ queue [13]. The CDF of the UAV-assisted compute delay T_{UAV} is expressed as

$$\mathbb{P}\{T_{\text{UAV}} \le t\} = F_{W_{\text{UAV}}}(t - D_{\text{UAV}} - T^{\text{UL}}_{\text{UAV}} - T^{\text{DL}}_{\text{UAV}})\times u(t - D_{\text{UAV}} - T^{\text{UL}}_{\text{UAV}} - T^{\text{DL}}_{\text{UAV}}), \tag{6}$$

where the CDF of the waiting time $F_{W_{\text{UAV}}}(x)$ is given by (10) for $D = D_{\text{UAV}}$, $\lambda = \xi\epsilon\mu$, and $c = c_{\text{UAV}}$ for all strategies.

3) HAPS-Assisted Compute: If a task is offloaded to HAPS, the compute delay also includes the propagation delay as defined in subsection II-C. Therefore, the overall delay for the HAPS-assisted compute can be written as follows:

$$T_{\text{HAPS}} = 2\tau + T^{\text{UL}}_{\text{HAPS}} + T^{\text{DL}}_{\text{HAPS}} + W_{\text{HAPS}} + D_{\text{HAPS}}, \tag{7}$$

where $T^{\text{UL}}_{\text{HAPS}}$, $T^{\text{DL}}_{\text{HAPS}}$, and D_{HAPS} are the UL and DL transmission delays and the processing delay at the HAPS, which are assumed deterministic. Similarly to the UAV-assisted compute case, the CDF of the HAPS-assisted compute delay T_{HAPS} is

$$\mathbb{P}\{T_{\text{HAPS}} \le t\} = F_{W_{\text{HAPS}}}(t - D_{\text{HAPS}} - T^{\text{UL}}_{\text{HAPS}} - T^{\text{DL}}_{\text{HAPS}} - 2\tau)\times u(t - D_{\text{HAPS}} - T^{\text{UL}}_{\text{HAPS}} - T^{\text{DL}}_{\text{HAPS}} - 2\tau), \tag{8}$$

where the CDF of the waiting time $F_{W_{\text{HAPS}}}(x)$ is given by (10) for $D = D_{\text{HAPS}}$, $\lambda = \eta N\mu\epsilon$ for the baseline strategy, $\lambda = \xi N\mu\epsilon$ for both splitting and replication strategies, and $c = c_{\text{HAPS}}$.

III. Analysis of Offloading Strategies

A. Waiting Time Distribution

Remark 1: The CDF of the waiting time for an $M/D/1$ queue with arrival rate λ and service time D is expressed as

$$F_W(x) = (1-\lambda D)\sum_{k=0}^{\lfloor \frac{x}{D} \rfloor} \frac{(-\lambda(x-kD))^k e^{\lambda(x-kD)}}{k!}. \tag{9}$$

Proof: See the derivation in [14]. □

Remark 2: The CDF of the waiting time for an $M/D/c$ queue with arrival rate λ and service time D is given by

$$F_W(x) = e^{\lambda(x-kD)}\sum_{j=0}^{kc-1} Q_{kc-j-1}\frac{(-\lambda(x-kD))^j}{j!}, \tag{10}$$

where $Q_m = \sum_{i=0}^{m+c} p_i$ and

$$p_j = e^{-\lambda D}\frac{(\lambda D)^j}{j!}\sum_{j=0}^{c} p_k + \sum_{k=c+1}^{c+j} p_k e^{\lambda D}\frac{(\lambda D)^{j-k+c}}{(j-k+c)!}.$$

Proof: See the derivation in [15]. □

To obtain the probabilities p_j in a computationally efficient manner, we employ the geometric tail approach. For more details on this method, we refer our readers to [16, p. 378].

B. Baseline Strategy

The baseline strategy assumes that UEs offload their tasks to the associated UAVs with probability ξ, to HAPS with probability η, or compute them locally with probability $1-\xi-\eta$. Given the probability $\mathbb{P}\{T_k \le t^*\}$ for $k \in \{\text{UE}, \text{UAV}, \text{HAPS}\}$, which denotes the probability that a task is completed within the deadline t^* either locally, at a UAV, or at the HAPS, the probability $P(\xi,\eta)$ of a task being computed in-time following the baseline strategy can be expressed as follows:

$$P(\xi,\eta) = (1-\xi-\eta)\mathbb{P}\{T_{\text{UE}} \le t^*\} + \xi\mathbb{P}\{T_{\text{UAV}} \le t^*\} + \eta\mathbb{P}\{T_{\text{HAPS}} \le t^*\}. \tag{11}$$

Since $\mathbb{P}\{T \le t^*\} = F_T(t^*)$ by definition, one can compute $P(\xi,\eta)$ by evaluating the CDF values from (4), (6), and (8) at t^*, respectively.

We seek ξ^* and η^* that maximize $P(\xi,\eta)$:

$$(\xi^*,\eta^*) = \arg\max P(\xi,\eta), \quad 0 \le \xi \le 1, 0 \le \eta \le 1, \quad \xi+\eta \le 1. \tag{12}$$

The optimization problem in (12) is NP-hard, and the objective function $P(\xi, \eta)$ is non-differentiable at every point in its domain due to the presence of the piece-wise Heaviside function. To tackle the problem in (12), we use a derivative-free Nelder-Mead solver [17] with different initial simpleces and the stopping condition of $\epsilon_{\text{stop}} = 0.01$ to ensure robust convergence close to the global optimum.

C. Replication Strategy

According to the replication strategy, UEs send task replicas to the UAVs and the HAPS with probability ξ_R and compute tasks locally with probability $1 - \xi_R$. Due to task replication, the overall system compute load is higher as compared to the baseline scenario under the same environment sampling rate ϵ. The task is completed if either the UAV or the HAPS returns a result to the UE before the deadline t^*. The probability $P(\xi)$ of such an event is given by

$$P_R(\xi) = (1 - \xi_R)\mathbb{P}\{T_{\text{UE}} \leq t^*\} + \xi_R\Big[\mathbb{P}\{T_{\text{UAV}} \leq t^*\} + \mathbb{P}\{T_{\text{HAPS}} \leq t^*\}\Big]. \quad (13)$$

To find the optimal offloading probability ξ_R^*, we numerically solve the following optimization problem:

$$\xi_R^* = \arg\max P_R(\xi_R), \quad 0 \leq \xi_R \leq 1. \quad (14)$$

D. Splitting Strategy

Instead of sending task replicas to the UAV and the HAPS, a UEs can split its task between them. The task is completed if the UE receives results in time from both the UAV and the HAPS. The probability of in-time compute for the splitting strategy is given by

$$P_S(\xi_S) = (1 - \xi_S)\mathbb{P}\{T_{\text{UE}} \leq t^*\} + \xi_S\mathbb{P}\{T_{\text{UAV}} \leq t^*\}\mathbb{P}\{T_{\text{HAPS}} \leq t^*\}. \quad (15)$$

We obtain the optimal offloading probability ξ_S^* using the Nelder-Mead solver for the following optimization problem:

$$\xi_S^* = \arg\max P_S(\xi_S), \quad 0 \leq \xi_S \leq 1. \quad (16)$$

The task arrival rate at the UAVs and HAPS is the same as that for the replication strategy, but the task sizes differ. Assuming a splitting factor δ, the proportion of tasks offloaded to the UAV $\delta_{\text{UAV}} = \delta$, while the proportion offloaded to the HAPS is $\delta_{\text{HAPS}} = 1 - \delta$.

IV. Numerical Results

The parameters used in our numerical assessment are provided in Table I. Each UE produces video frames by capturing a sensory representation of the environment with a size of 0.375 MB at a frame rate of 5 frames per second (FPS). We assume that the deadline for processing the tasks is inversely proportional to their sampling rate, i.e., $t^* = 1/\epsilon$, which is a reasonable assumption for the real-time video processing applications [9]. Each task has a constant computational load of $C = 100$ GFLOPs, which is typical for applications such as object detection or semantic segmentation [12].

TABLE I
Simulation Parameters

Parameter name	Parameter value
Bandwidth, W	400 MHz
Antenna gain of UE/UAV/HAPS, G	3/10/30 dBi
Transmit power, P	20 dBm
Noise temperature, T	300 K
CPU frequency of UEs, C_{UE}	200 GFLOPs
CPU frequency of UAVs, C_{UAV}	500 GFLOPs
CPU frequency of HAPS, C_{HAPS}	3000 GFLOPs
Number of CPUs at UAVs, c_{UAV}	5
Number of CPUs at HAPS, c_{HAPS}	15
Task arrival rate at UE, ϵ	5 FPS
Deadline, t^*	0.2 s
Task size, n	0.375 MB

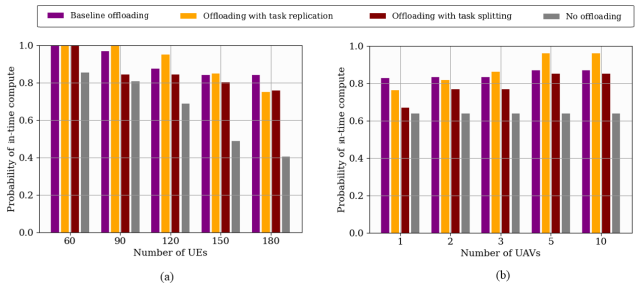

Fig. 2. Proportion of timely computed tasks as a function of the number of (a) UEs and (b) UAVs. There are 5 UAVs in (a) and 120 UEs in (b).

We evaluate three reference offloading strategies against a strategy with no offloading under different system configurations. For the offloading strategies, we determine the optimal offloading probabilities as described in Section III and employ them in our extensive simulations. For each configuration, we collect statistics from multiple simulation runs until a 95% confidence level in the estimates is achieved. In the simulation, the channel realizations between UEs, UAVs, and HAPS follow the methodology from [12].

Multi-connectivity can be implemented at various levels, including the PHY, MAC, PDCP, and core network layers [8]. In terrestrial networks, multi-radio dual connectivity has been standardized as an appealing PDCP-layer solution due to its adaptability to changing radio conditions. We consider this approach for our multi-connectivity setup, even though it has not yet been standardized for NTNs.

Fig. 2 shows the probability of in-time computation as a function of (a) the number of UEs and (b) the number of UAVs. This probability remains above 0.7 for all the offloading strategies and UE deployment densities considered. As the number of UEs increases, the probability of in-time computation without offloading decreases more rapidly than that with any offloading strategy, by eventually dropping below 0.4. Meanwhile, the average UE energy consumption without offloading, as illustrated in Fig. 3, is consistently higher than the energy consumption with any offloading strategy.

Fig. 2 suggests that for deployments with fewer UEs, the replication strategy achieves the highest probability of

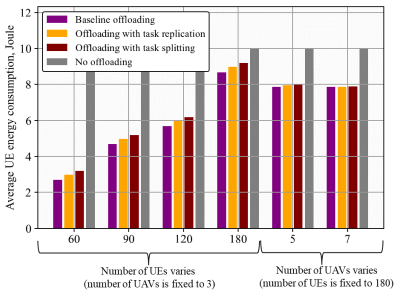

Fig. 3. Average UE energy consumption.

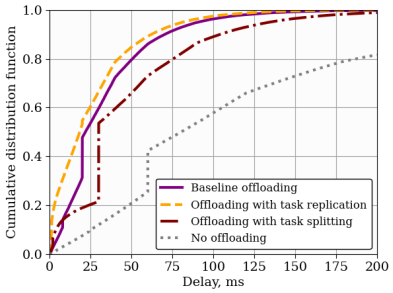

Fig. 4. CDF of the task compute delay.

timely computation due to the benefits of compute diversity. This trend persists up until the combined load on the UAVs and HAPS, along with the replication overheads, becomes sufficiently high. At this point, both task splitting and baseline offloading become more preferable solutions, which is particularly evident when there are 180 UEs. While the task replication strategy consistently consumes more energy than the baseline strategy due to a lower offloading probability, this increase in energy consumption is negligible as compared to the significant improvement in the numbers of tasks computed in time.

Similarly, the replication strategy delivers the highest probability of timely computation as more UAVs are deployed, thereby offering additional compute capacity. The task splitting strategy performs comparably to the baseline strategy and does not show a significant improvement even with the deployment of more UAVs. For instance, when the number of UAVs exceeds 2, the task replication strategy offers a 5-17% gain over both the baseline and the task splitting offloading strategies. However, when there are fewer than 3 UAVs, as illustrated in Fig. 2, the baseline strategy outperforms the replication and the task splitting strategies by 3-5% and 3-7%, respectively.

Fig. 4 presents the CDF of the task compute delay as determined by (3), (5), and (7) for 60 UEs and 5 UAVs as an example. In this deployment, all tasks are computed before the deadline of 500 ms. Offloading strategies significantly reduce both the average and the worst-case task compute delay. The replication strategy exhibits the lowest delays among all strategies, while the baseline strategy, although being slightly inferior to the replication strategy, outperforms the task splitting strategy. By utilizing multi-connectivity offloading mechanisms, the system can efficiently distribute tasks across available resources, thereby reducing computation delay and mitigating service interruptions.

V. Conclusion

Given the challenges of deploying dense terrestrial networks and the growing need for MEC, this letter offers a novel analysis of different computation task offloading approaches in UAV- and HAPS-assisted NTNs. The considered offloading strategies leverage multi-connectivity capabilities between the target UEs and the NTN nodes to increase the probability of timely computation for the UE tasks. Our results indicate that multi-connectivity offloading significantly improves the probability of in-time compute, by ensuring that 70% of tasks are completed before the deadline. The replication strategy achieves the highest probability of timely computation and the lowest task compute delay when the number of UEs is smaller. Conversely, the baseline strategy outperforms both task replication and splitting strategies as the computation load grows.

References

[1] C. E. Kement et al., "Sustaining dynamic traffic in dense urban areas with HAPS," *IEEE Commun. Mag.*, vol. 61, no. 7, pp. 150–156, Jul. 2023.

[2] G. K. Kurt et al., "A vision and framework for the HAPS networks of the future," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 729–779, 2nd Quart., 2021.

[3] Q. Ren, O. Abbasi, G. K. Kurt, H. Yanikomeroglu, and J. Chen, "Caching and computation offloading in HAPS assisted intelligent transportation systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 9010–9024, Nov. 2022.

[4] D. S. Lakew, A.-T. Tran, N.-N. Dao, and S. Cho, "Intelligent offloading and resource allocation in HAP-assisted MEC networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 1582–1587.

[5] Z. Jia, Q. Wu, C. Dong, C. Yuen, and Z. Han, "Hierarchical aerial computing for Internet of Things via cooperation of HAPs and UAVs," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5676–5688, Apr. 2023.

[6] S. M. Shahid, Y. T. Seyoum, S. H. Won, and S. Kwon, "Load balancing for 5G integrated satellite-terrestrial networks," *IEEE Access*, vol. 8, pp. 132144–132156, 2020.

[7] M. López, S. B. Damsgaard, I. Rodríguez, and P. Mogensen, "Connecting rural areas: An empirical assessment of 5G terrestrial-LEO satellite multi-connectivity," in *Proc. IEEE 97th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2023, pp. 1–5.

[8] M. Majamaa, "Toward multi-connectivity in beyond 5G non-terrestrial networks: Challenges and possible solutions," *IEEE Commun. Mag.*, early access, Jan. 15, 2024, doi: 10.1109/MCOM.001.2300581.

[9] A. Traspadini, M. Giordani, G. Giambene, and M. Zorzi, "Real-time HAP-assisted vehicular edge computing for rural areas," *IEEE Wireless Commun. Lett.*, vol. 12, no. 4, pp. 674–678, Apr. 2023.

[10] Z. Lou, R. Wang, B. E. Y. Belmekki, M. A. Kishk, and M.-S. Alouini, "Terrain-based UAV deployment: Providing coverage for outdoor users," *IEEE Trans. Veh. Technol.*, vol. 73, no. 6, pp. 8988–9002, Jun. 2024.

[11] N. Sathyavageeswaran, R. D. Yates, A. D. Sarwate, and N. Mandayam, "Timely offloading in mobile edge cloud systems," 2024, *arXiv:2405.07274*.

[12] G. Pan, H. Zhang, S. Xu, S. Zhang, and X. Chen, "Joint optimization of video-based AI inference tasks in MEC-assisted augmented reality systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 2, pp. 479–493, Apr. 2023.

[13] Z. Xie et al., "A Markovian queueing model for end-to-end delay analysis in computation offloading system," *IEEE Commun. Lett.*, vol. 27, no. 10, pp. 2687–2691, Oct. 2023.

[14] A. K. Erlang, "The theory of probabilities and telephone conversations," *Nyt. Tidsskr. Mat. Ser. B*, vol. 20, pp. 33–39, 1909.

[15] G. J. Franx, "A simple solution for the M/D/*c* waiting time distribution," *Oper. Res. Lett.*, vol. 29, no. 5, pp. 221–229, Dec. 2001.

[16] H. C. Tijms, *A First Course in Stochastic Models*. Hoboken, NJ, USA: Wiley, 2003.

[17] S. Singer and J. Nelder, "Nelder–Mead algorithm," *Scholarpedia*, vol. 4, no. 7, p. 2928, 2009.