Information Freshness in Orbital Edge Computing Through Dual-Queue Offloading

Umut Guloglu
Inst. of Comm. and Navigation
German Aerosp. Center (DLR), Germany
email: umut.gueloglu@dlr.de

Leonardo Badia
Dept. of Information Engineering
University of Padova, Italy
email: badia@dei.unipd.it

Andrea Munari
Inst. of Comm. and Navigation
German Aerosp. Center (DLR), Germany
email: andrea.munari@dlr.de

***Abstract*—We study the minimization of age of information (AoI) in a dual-queue architecture for satellite-assisted Internet of things systems. We consider multiple source–monitor pairs that generate status updates that can be processed locally or offloaded to a shared common server on a satellite, introducing a trade-off between queueing delay and propagation latency. The objective is to determine the optimal offloading probability that minimizes the average AoI under shared resource contention. To address the structural complexity of heterogeneous queues, we propose a non-iterative heuristic policy with an explicit expression based on the optimal operating points of individual queues. The resulting policy adopts a piecewise workload allocation strategy that proportionally distributes traffic at low load and prioritizes the most AoI-efficient server at high load. The numerical results show that the proposed scheme closely approximates the minimum-AoI performance across different service rates and propagation delay regimes, providing a scalable and practical solution for AoI-aware task offloading in orbital edge computing systems.**

***Index Terms*—Orbital Edge Computing, Age of Information, Internet of Things, Heuristic algorithms, Computation Offloading.**

I. Introduction

With the decreasing costs of launching and maintaining non-terrestrial infrastructure, the adoption of satellite-based systems has surged in recent years. Among these, orbital edge computing (OEC) has emerged as a prominent application in which satellites are equipped with onboard processing [1]. This advancement is especially impactful for Internet of things (IoT) applications operating in areas where terrestrial network coverage is absent or unreliable, such as rural villages, maritime platforms, or mountainous terrains.

In these environments, IoT devices generate time-sensitive data streams for applications such as environmental monitoring, precision agriculture, or remote tracking. However, due to limited local computational resources, many IoT nodes cannot process data locally in an efficient way, and stale data can lead to inaccurate analysis or delayed decision-making. In these scenarios, ensuring up-to-date, processed information becomes more critical than minimizing individual packet delays, as data freshness jointly depends on how information is generated and delivered and is further influenced by update frequency [2]. To this end, OEC can enable real-time task offloading from IoT devices to available satellites, which process data and return results, as illustrated in Fig. 1.

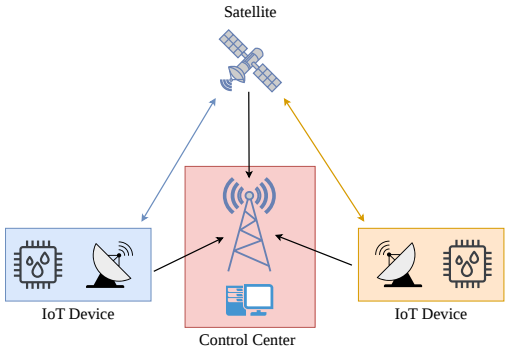


Fig. 1. An offloading scenario. The IoT devices can process time-sensitive data locally or offload to a satellite. The processed data is transmitted to a monitor, which can be: (a) the IoT device itself (colored lines), or (b) a centralized control center (colored and black lines).

To capture the freshness aspect of system performance, Age of Information (AoI) is commonly used, measuring the time elapsed since the generation of the most recently completed processing available at the endpoint [3]. Offloading AoI-sensitive tasks to edge servers requires additional effort due to the increased latency induced by the physical separation between end users and edge infrastructure. This challenge is particularly pronounced in OEC, where high propagation delays significantly impact timely data delivery.

In multi-user OEC environments, multiple users accessing the same edge satellite lead to shared resource contention, resulting in increased queueing delays and longer buffer occupancy times. In such scenarios, the selection of an appropriate offloading rate becomes a critical design decision. High rates can indeed overload the common (edge) queue, increasing flow staleness even when the queue remains stable, while leaving the local queues that process non-offloaded tasks underutilized. Conversely, an overly conservative offloading strategy may result in local queues becoming congested, while the common queue remains underutilized. The optimal offloading policy is in general complex and depends on the arrival rates of the sources and the service capacities of both local and common queues, along with the number of users and propagation delay.

From this perspective, optimizing the use of local or remote queue alone may in general not lead to the minimum overall AoI. For example, the authors in [4] derived that in one user

no-propagation dual queue systems, the optimal offered load of each queue is 0.533, slightly higher than the optimal value for individual M/M/1 queues, which is around 0.531. However, the difference may be negligible in practice, indicating that decoupled optimization may provide a robust approach to near-optimal performance under typical conditions. Motivated by this observation, the main novelty of this work is the design of a simple and practical heuristic that strikes a balance between AoI performance and implementation complexity in identical multi-user OEC systems with consistent satellite connectivity. Specifically, we define a non-iterative heuristic for OEC environments that can be derived directly, lowering the computational burden on IoT nodes, which is crucial given their resource constraints. By means of numerical results, we show that the solution offers good performance in a wide range of system configurations in terms of service rates and propagation delays, providing a scalable solution for AoI-aware task-offloading.

II. Related Work

The concept of AoI was introduced in [3] to characterize information freshness in status update systems. Since then, it has been extensively studied, including revisits of queueing theory and extension to wireless network applications, especially including optimal scheduling and energy-aware transmission policies [5], [6]. Beyond the approaches based on centralized optimization, the literature has also extended AoI analysis to random-access scenarios, considering competition for shared wireless resources [7]. At the same time, the expected increase of satellite and non-terrestrial networks (NTNs) prompts a discussion towards orbital edge computing, i.e., offloading to satellite servers [1]. In NTNs, even basic latency considerations differ fundamentally from terrestrial systems due to large propagation delays and dynamic topology. Round-trip times can be significantly longer, especially in GEO satellites, and LEO constellations may introduce multi-hop relaying and lack of coverage [8], [9]. 3GPP NTN standardization [10] explicitly acknowledges extended propagation delays and intermittent link availability as key design challenges.

OEC has been explored from various perspectives in the literature [11]. Existing studies primarily focus on energy [12] or latency [13], which are critical to ensuring reliable and cost-effective system operation. Some studies developed solutions to maximize the profit of the OEC-MEC service provider [14]. For instance, [15] investigates offloading decisions in satellite–aerial integrated computing for disaster scenarios, while [16] presents a multi-layer offloading solution tailored for autonomous farm vehicles.

Although these works provide valuable information on optimizing traditional performance metrics, the freshness of information has received comparatively limited attention in the OEC. Among the few contributions existing, [17] proposes an online Lyapunov based framework for joint communication/computation offloading in LEO satellite integrated 6G networks, targeting minimization of the average peak AoI under energy constraints. A further distinguishing feature of satellite networks is intermittent connectivity, due to limited satellite visibility windows in LEO constellations, leading to service interruptions and time-varying access rates. This phenomenon has been extensively studied in delay-tolerant networking (DTN), where contact plans and store-and-forward strategies are required [18]. In the context of freshness-sensitive traffic, intermittent link availability implies that the evolution of AoI must account for deterministic contact windows and potential service outages [19].

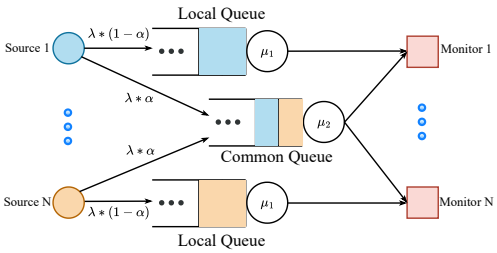


Fig. 2. Dual queueing model illustration.

III. System Model

We consider a system with N identical source–monitor pairs. The sources are IoT nodes, which generate tasks for processing, and the corresponding monitors track the status of the processed tasks. Two processing servers are available: one located on a satellite and one co-located with each IoT node, both operating on a first-in-first-out (FIFO) basis. Each IoT node decides whether to process a task locally or offload it to the satellite. In the case of offloading, tasks are executed directly on the satellite without being forwarded to any other entity. Satellite connectivity is considered to be consistently available; this can be relaxed according to [20] but our analysis still works within the boundary of a specific availability window. Upon completion (local or satellite), the processed task result is forwarded to the monitor, which may be collocated with the IoT node or placed at a different location (see Fig. 1). The satellite server is shared among all N users and serves as a common resource. The system model is schematically represented in Fig. 2.

Task arrivals from each source follow a general arrival process with average rate λ, which is constant and time-invariant for all source-monitor pairs. While λ remains fixed, the arrival process itself may exhibit different memoryless characteristics within the model such as geometric or exponential inter-arrival times. Each new arrival is independently offloaded to the common queue with probability α or processed locally with probability $1-\alpha$. Consequently, the effective average arrival rate to each local queue is $\lambda_L = \lambda(1-\alpha)$, while each source-monitor pair contributes arrivals to the common queue at rate $\lambda_{C,S} = \lambda\alpha$. The service rates of each local queue and the common queue are denoted by μ_L and μ_C, respectively, with no assumptions made about service time distributions.

We adopt AoI as the performance metric to quantify the freshness of updates at the monitor. The metric is generally

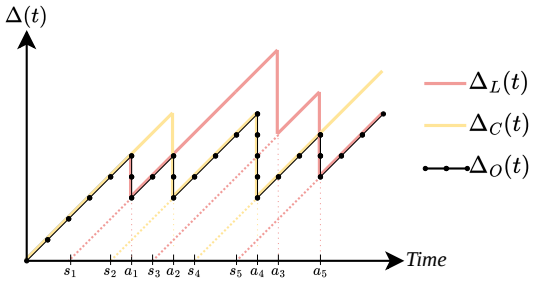


Fig. 3. Status age graph for local, common and overall system.

defined as the time elapsed since the most recent task completion, i.e., $\Delta(t) = t - U(t)$, where t is time, and $U(t)$ is the generation time of the most recently delivered processed task at time t. In our setting, some remarks are in order. To this aim, we provide in Fig. 3 an example of AoI evolution over time. In the plot, the i-th task is generated at time s_i, processed either locally or offloaded to the satellite, and received at the monitor at time a_i. Due to the presence of two processing servers, the AoI for tasks processed on each follows a distinct evolution. We denote the AoI for locally processed tasks as $\Delta_L(t)$ and for offloaded tasks as $\Delta_C(t)$. The overall AoI at the monitor is given by

$$\Delta_O(t) = \min\{\Delta_L(t), \Delta_C(t)\}$$

representing the most recent update available. In classical FIFO queueing models, the AoI typically drops immediately after a new update is delivered. However, in our model, the most recently processed task on one server may be older than the most recent update on the other, leading to a situation where the overall AoI does not decrease upon a new delivery. This situation is exemplified in the figure, where the third task is processed locally and arrives at the monitor at a_3, but does not reduce the overall status age because a fresher update is already available from the common server.

Our goal is to minimize the average AoI, calculated as

$$\Delta = \lim_{T\to\infty} \frac{1}{T} \int_0^T \Delta_O(t) dt.$$

The quantity depends on the absolute propagation delays for both the local and satellite processing. In case (a), where the monitor is co-located with the IoT node, no propagation is required after local processing. In case (b), however, the result must be transmitted to a remote monitor. In this case, the local path involves a short transmission from the IoT node to the monitor, while the offloaded path involves a longer transmission in both cases. However, in the calculation of the AoI, any common delay component that affects both servers equally cancels out in the relative comparison, leaving only the difference in processing and queuing delays as relevant. To simplify the analysis without loss of generality, we define D as the difference in propagation delay between the satellite and local servers, treating it as a deterministic parameter. Assuming the satellite path has a longer delay, $D > 0$ represents the additional time required to deliver an offloaded task to the monitor compared to a locally processed one.

IV. Proposed Heuristic Solution

The authors in [4] studied a similar setup considering two M/M/1 queues with no propagation delay, and derived Δ leveraging the cumulative distribution function (CDF) of AoI in M/M/1 systems [21]. However, computing the optimal offloading rate from this expression is challenging, as it involves high-degree multinomials. Furthermore, closed-form expressions for the CDF of AoI in other queueing models are difficult to obtain, which makes the analysis more complex in general settings. To address these challenges, we propose a simple heuristic solution that can be applied across different system configurations and achieves performance close to the optimal, while maintaining low computational complexity.

The proposed solution adopts a piecewise workload allocation strategy parameterized by the arrival rate λ and the optimal offered loads $\rho_{L,\text{opt}}$ and $\rho_{C,\text{opt}}$, i.e., the loads that minimize the average AoI in the local and common queue when considered in isolation, respectively.

Specifically, when the arrival rate λ is not high enough to drive the queues to their optimal operating points, the offloading decision is based on the observed service rates. The strategy aims to balance the load across both queues by maintaining a consistent ratio of utilization relative to the optimal operating point, rather than completely filling one queue before the other. For the local queue, the observed service rate remains μ_L, as there is no propagation delay. For the common queue, the observed service rate accounts for the additional delay and is given by $\mu_{C,\text{obs}} = \frac{\mu_C}{1+D\mu_C}$, where D is the deterministic delay difference. By equating the ratio of the load in each queue to its optimal utilization target, the offloading rate is computed as

$$\alpha(\lambda) = \frac{\mu_{C,\text{obs}} \cdot \rho_{C,\text{opt}}}{\mu_{C,\text{obs}} \cdot \rho_{C,\text{opt}} + \mu_L \cdot \rho_{L,\text{opt}}},$$

where $\rho_{L,\text{opt}}$ and $\rho_{C,\text{opt}}$ are the optimal utilization targets for the local and common queues, respectively.

Within this strategy, the common queue remains underutilized due to its reduced observed service rate, $\mu_{C,\text{obs}} = \frac{\mu_C}{1+D\mu_C} \leq \mu_C$ for all $D \in \mathbb{R}_{\geq 0}$. In other words, the observed service rate of the common server is effectively lower, so offloading is reduced to maintain the desired load ratio. At $\lambda_{\text{Bound}} = \mu_L \rho_{L,\text{opt}} + \mu_{C,\text{obs}} \rho_{C,\text{opt}}$, the local queue operates at optimal utilization while the common queue is underutilized.

For arrival rates higher than this boundary point, the proposed strategy identifies the queue that achieves a lower average AoI at optimal utilization, designated as the primary processing resource, while the other serves as a secondary overflow buffer for excess workload. This prioritizes AoI minimization over load balancing. The choice depends on the parameters: if the common queue yields better performance, meaning the average AoI of the common server at optimal utilization is lower than that of the local one ($\Delta_{C,\text{opt}} < \Delta_{L,\text{opt}}$),

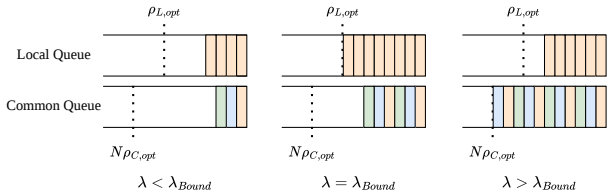


Fig. 4. Normalized load in queues.

it is used at its optimal load and the offloading rate becomes $\alpha = \frac{\mu_C \rho_{C,\text{opt}}}{\lambda}$. This rate ensures that the load on the common queue is exactly $\rho_{C,\text{opt}}$. Conversely, if the local queue is superior ($\Delta_{C,\text{opt}} \geq \Delta_{L,\text{opt}}$), it operates at its optimal point, and all remaining tasks are offloaded to the common queue. In this case, the offloading rate becomes $1 - \frac{\mu_L \rho_{L,\text{opt}}}{\lambda}$ to drive the queue at optimal load.

The overall behavior of the proposed heuristic can then be summarized in terms of the splitting factor α as

$$\alpha(\lambda) = \begin{cases} \dfrac{\mu_{C,\text{obs}} \rho_{C,\text{opt}}}{\mu_{C,\text{obs}} \rho_{C,\text{opt}} + \mu_L \rho_{L,\text{opt}}} & \text{if } \lambda \leq \lambda_{\text{Bound}} \\ 1 - \dfrac{\mu_L \rho_{L,\text{opt}}}{\lambda} & \text{if } \lambda > \lambda_{\text{Bound}} \\ & \& \ \Delta_{C,\text{opt}} \geq \Delta_{L,\text{opt}} \\ \dfrac{\mu_C \rho_{C,\text{opt}}}{\lambda} & \text{if } \lambda > \lambda_{\text{Bound}} \\ & \& \ \Delta_{C,\text{opt}} < \Delta_{L,\text{opt}} \end{cases}$$

An exemplary illustration is also provided in Fig. 4 for the three operating regions, assuming the common server has lower average AoI than the local server at optimal points.

We remark that identifying the queue with lower average AoI at its optimal load is not trivial in general, and deriving a universal decision rule across arbitrary queue types remains challenging. While analytical expressions can be obtained for specific configurations, such derivations are highly dependent on system-specific parameters and are therefore not scalable to a general framework. As this work focuses on a unified, broadly applicable solution applicable to FIFO queues, we exclude case-specific performance analyses from the scope. Instead, we rely on the known optimal traffic intensities $\rho_{L,\text{opt}}$ and $\rho_{C,\text{opt}}$, one for each queue, as guiding principles for offloading decisions. It is also assumed that users possess sufficiently accurate probing capabilities. Potential discrepancies between the probed and the true optimal traffic intensities are not considered within the scope of this work.

A. Benchmark Solutions

To evaluate the performance of the proposed offloading policy, several benchmark solutions are considered and compared. Specifically, we evaluate the following baselines:

- **Local-Optimized Offloading**: Each pair optimizes its own local queue to minimize average AoI. The offloading rate is equal to $1 - \frac{\mu_L \rho_{L,\text{opt}}}{\lambda}$ for all λ values.
- **Common-Optimized Offloading**: The shared queue is optimized to minimize average AoI. The offloading rate is equal to $\alpha = \frac{\mu_C \rho_{C,\text{opt}}}{\lambda}$ for all λ values.
- **Empirical Optimal Offloading**: Simulation results are used to identify the offloading rate that yields the lowest average AoI for each fixed configuration.

V. Results and Discussion

Extensive simulations were carried out to evaluate the proposed solution. Targeting practically relevant configurations, we base the local service rate μ_L on empirical measurements from real hardware. Specifically, we focus on an average processing time of 50 ms, as reported in [22] from experimental measures for processing of a video frame across multiple IoT platforms on Raspberry Pi 3, and set $\mu_L = \frac{1}{0.05\ \text{s}} = 20$ packets s^{-1}. Additionally, the M/M/1 model was employed for the shared queue in all experiments, because the optimal traffic load $\rho_{C,\text{opt}}$ can be derived without resorting to Monte-Carlo simulations.

In the absence of propagation delays, we first examine a baseline of the average AoI in the dual-queue mode. In this initial set of experiments, we consider 4 source-monitor pairs, **without** other latencies than queueing delays. For each pair of arrival rate (λ) and offloading rate (α) values we run 100 Monte-Carlo simulations and record the resulting average AoI.

Using these data, we construct Figs. 5 and 6, which show the average AoI and the corresponding offloading rate, respectively, for three distinct scenarios:

- **M/M/1 local queue with** $\mu_C = 20$ (Fig. 5a and Fig. 6a).

In this scenario, the local and common servers are identical. For each arrival rate, the heuristic solution yields average AoI results that are extremely close to the optimal ones. Although the heuristic offloading rate is slightly higher than the optimal rate when λ is low, the performance gap remains minimal due to the low arrival rates. This is further confirmed by the maximum relative difference between the heuristic and optimal solutions, which is only about 0.5%.

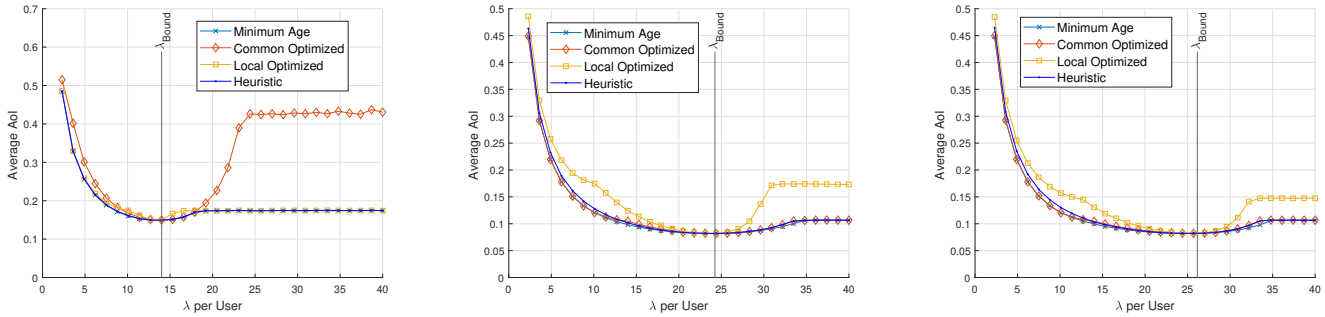
- **M/M/1 local queue and** $\mu_C = 80$ (Fig. 5b and Fig. 6b).

When the common server is made four times faster, the system behavior changes markedly. For low arrival rates, the optimal strategy is to offload all packets, because the queueing delay on the common server is now negligible; the faster server therefore yields the lowest average AoI. As λ increases, the queueing delay at the common server increases and the heuristic policy, which offloads only a fraction of the traffic, becomes preferable.

In principle, the heuristic can be extended to cover the region where full offloading is optimal, but doing so would undermine the main purpose of the method: it is meant to be an easy-to-apply rule that already provides results that are sufficiently accurate. Even in the worst case the relative error of the heuristic never exceeds 6.8%.

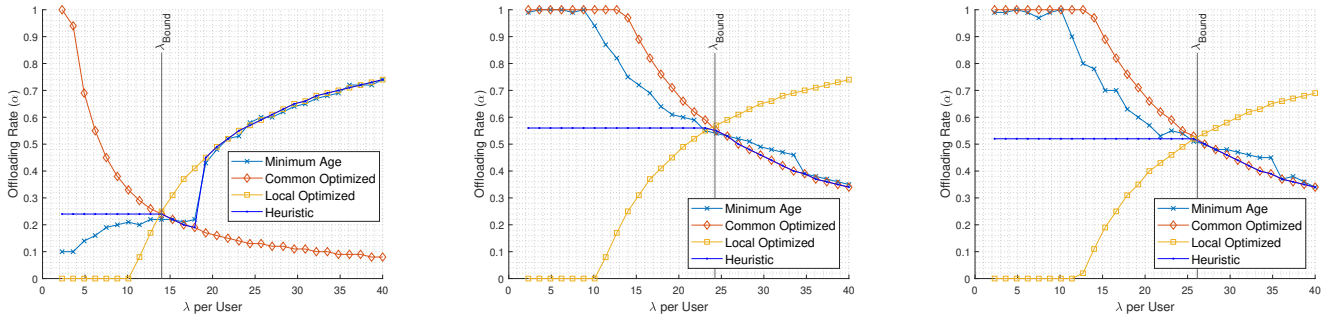
- **M/D/1 local queue with** $\mu_C = 80$ (Fig. 5c and Fig. 6c).

Altering the local queue processing statistics also changes the problem. For an M/D/1 local queue, the offloading rate for



(a) Local M/M/1 queue & $\mu_C = 20$ (b) Local M/M/1 queue & $\mu_C = 80$ (c) Local M/D/1 queue & $\mu_C = 80$

Fig. 5. Average AoI vs. arrival rate (λ) for common M/M/1 queue, $\mu_L = 20$, $N = 4$ and $D = 0$.



(a) Local M/M/1 queue & $\mu_C = 20$ (b) Local M/M/1 queue & $\mu_C = 80$ (c) Local M/D/1 queue & $\mu_C = 80$

Fig. 6. Offloading rates (α) vs. arrival rate (λ) for common M/M/1 queue, $\mu_L = 20$, $N = 4$ and $D = 0$.

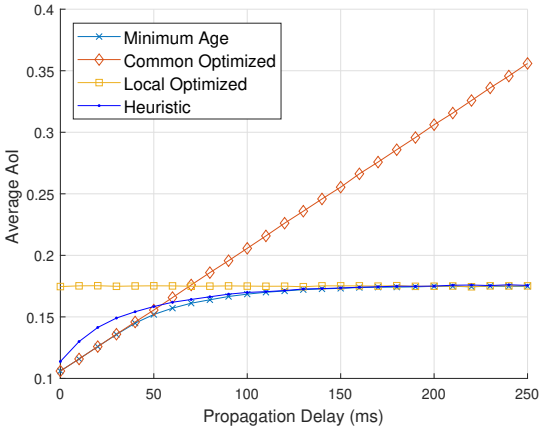
arrival rates $\lambda < \lambda_{\text{Bound}}$ is relatively smaller than in the M/M/1 case, because the optimal local utilization $\rho_{L,\text{opt}}$ is higher for M/D/1 queues. This difference is evident when comparing Figs. 6b and 6c. On the other hand, the average AoI behavior closely resembles that of a local M/M/1 queue. For small values of λ, the optimal strategy follows the common server optimized approach, and significant differences are observed in the selected α values. However, due to the infrequent arrival of tasks, the average AoI remains relatively insensitive to changes in α. As λ increases, the system behavior begins to deviate from this regime. Beyond a certain threshold, the optimal α values converge toward the heuristic solution and eventually coincide with it at λ_{Bound}.

The reported results clarify how the proposed heuristic delivers satisfactory performance in a range of scenarios where there is no propagation delay, with some performance degradation in the low-to-moderate λ regime. To complement our study, we further examine such challenging conditions in the presence of significant round trip times, which are representative of OEC environments. In such systems, the two-way propagation delay is non-negligible, ranging from 2 to 26 ms for LEO satellites, 48 to 198 ms for MEO satellites, and 240 to 272 ms for GEO satellites [23]. In these settings, propagation delay often dominates both queuing and processing latencies, significantly shaping system dynamics. To evaluate performance under low-load conditions across different delay regimes, a controlled experiment is conducted with a fixed arrival rate of $\lambda = 10$ packets per second and a common service rate of $\mu_C = 200$ packets per second. The resulting average AoI and the corresponding offloading rate are presented in Fig. 7a and Fig. 7b, respectively.

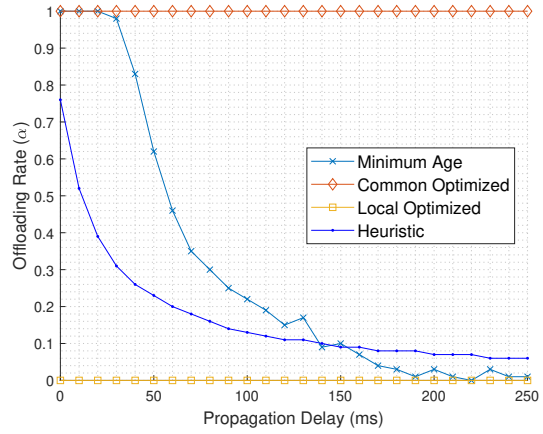
In LEO satellites, where the propagation delay is minimal, the common server strategy remains effective and the full offloading achieves near-optimal performance. The gap of the heuristic solution compared to the optimal is small at very low D, as the simplified observed service rate has minimal impact on system behavior. The gap peaks around D=15 ms, where the heuristic underestimates the true capabilities of the common server by relying on the observed service rate. This estimation leads to suboptimal offloading decisions in this intermediate delay range. In contrast, in MEO and GEO regions, where propagation delays are significantly larger, the heuristic solution becomes increasingly advantageous, consistently producing results that are nearly indistinguishable from the optimal policy. This shows that the heuristic is robust under high-latency conditions, even in the worst cases.

VI. Conclusions

We investigated AoI-aware task offloading in a dual queue architecture modeling satellite-assisted edge computing systems. The objective is to probabilistically split traffic between dedicated local processors and a shared common queue that represents a global satellite server, which is more powerful but also subject to propagation delay and resource contention.



(a) Average AoI



(b) Offloading Rate (α)

Fig. 7. Changes over propagation delay (D) at $\mu_L = 20$, $\mu_C = 200$, $\lambda = 10$, and $N = 4$.

We proposed a heuristic approach towards AoI minimization, which leverages the optimal operating points of individual queues and adapts the offloading rate through a piecewise workload allocation strategy. The resulting solution is non-iterative, scalable, and directly computable from system parameters. Numerical results demonstrated that the proposed policy closely approximates minimum-AoI performance across different service rates and delay regimes, including scenarios where propagation delay dominates queueing effects.

The proposed framework provides practical design guidelines for freshness-aware workload allocation in orbital edge computing systems. Future work may extend the model to probing discrepancies, heterogeneous users, non-cooperative settings, and dynamic arrival processes with time-varying system conditions.

REFERENCES

[1] Z. Yin, C. Wu, C. Guo, Y. Li, M. Xu, W. Gao, and C. Chi, "A comprehensive survey of orbital edge computing: Systems, applications, and algorithms," *Chinese J. Aeronaut.*, vol. 38, no. 7, p. 103316, 2025.

[2] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, 2021.

[3] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, 2012, pp. 2731–2735.

[4] A. Bhati, S. R. B. Pillai, and R. Vaze, "On the age of information of a queuing system with heterogeneous servers," in *Proc. Nat. Conf. Commun. (NCC)*, 2021.

[5] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[6] A. Munari and L. Badia, "What's my age of information again? The role of feedback in AoI optimization under limited transmission opportunities," *IEEE Trans. Commun.*, vol. 73, no. 11, pp. 10 495–10 508, 2025.

[7] A. Buratto, A. Munari, and L. Badia, "Strategic backoff of slotted ALOHA for minimal age of information," *IEEE Commun. Lett.*, vol. 29, no. 1, pp. 155–159, 2025.

[8] C. Caini, R. Firrincieli, and D. Lacamera, "Comparative performance evaluation of TCP variants on satellite environments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2009.

[9] G. Cocco, T. De Cola, M. Angelone, Z. Katona, and S. Erl, "Radio resource management optimization of flexible satellite payloads for DVB-S2 systems," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 266–280, 2017.

[10] 3GPP, "Study on new radio (NR) to support non-terrestrial networks," *3rd Generation Partnership Project (3GPP), Technical Report (TR), TR 38.811*, 2020.

[11] Y. Lin, W. Feng, Y. Wang, Y. Chen, Y. Zhu, X. Zhang, N. Ge, and Y. Gao, "Satellite-MEC integration for 6G Internet of things: Minimal structures, advances, and prospects," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 3886–3903, 2024.

[12] Q. Tang, Z. Fei, B. Li, H. Yu, Q. Cui, J. Zhang, and Z. Han, "Stochastic computation offloading for LEO satellite edge computing networks: A learning-based approach," *IEEE Internet Things J.*, vol. 11, no. 4, pp. 5638–5652, 2024.

[13] Y. Hao, Z. Song, Z. Zheng, Q. Zhang, and Z. Miao, "Joint communication, computing, and caching resource allocation in leo satellite mec networks," *IEEE Access*, vol. 11, pp. 6708–6716, 2023.

[14] B. Wang, X. Li, D. Huang, and J. Xie, "A profit maximization strategy of MEC resource provider in the satellite-terrestrial double edge computing system," in *Proc. IEEE Int. Conf. Commun. Tech. (ICCT)*, 2021, pp. 906–912.

[15] L. Zhang, H. Zhang, C. Guo, H. Xu, L. Song, and Z. Han, "Satellite-aerial integrated computing in disasters: User association and offloading decision," in *Proc. IEEE ICC*, 2020, pp. 554–559.

[16] D. Li, B. Yin, D. Yan, J. Liu, G. Xing, and Z. Li, "SAFVIN: Edge intelligence for satellite and autonomous farm vehicle integrated networks," *IEEE Trans. Mobile Comput.*, 2026.

[17] K. Li, J. Jiao, J. Huang, Z. Xu, Q. Sun, X. Xu, Y. Wang, and Q. Zhang, "Age-critical joint communication and computation offloading for satellite-integrated internet," *IEEE Trans. Cognitive Commun. Netw.*, vol. 12, pp. 4387–4403, 2026.

[18] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, 2003.

[19] Ç. Arı, Ö. T. Kartal, A. Munari, L. Badia, E. Uysal, and O. Kaya, "Optimizing peak age under intermittent satellite connectivity and store-and-forward," in *Proc. Asilomar Conf.*, 2025.

[20] L. Badia and A. Munari, "Satellite intermittent connectivity and its impact on age of information for finite horizon scheduling," in *Proc. Adv. Satellite Multimedia Syst. Conf. Signal Process. Space Commun. Wkshp (ASMS/SPSC)*, 2025.

[21] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, "A general formula for the stationary distribution of the age of information and its application to single-server queues," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, p. 8305–8324, 2019.

[22] A. Jevremovic, Z. Kostic, and D. Perakovic, "Energy-efficient edge intelligence: A comparative analysis of AIoT technologies," *Mob. Netw. Applic.*, vol. 29, pp. 147–155, 2023.

[23] 3GPP, "Service requirements for the 5G system," 3rd Generation Partnership Project (3GPP), Technical Specification (TS), TS 22.261, 2025.