2025 36th International Teletraffic Congress (ITC 36)

Server Selection and Inference Rate Optimization in AoI-Driven Distributed Systems

Leonardo Badia*, Paolo Castagno†, Vincenzo Mancuso‡§, Matteo Sereno† and Marco Ajmone Marsan§
*Dept. Information Engineering, University of Padova, 35131 Padua, Italy
†Computer Science Department, University of Turin, 10125 Turin, Italy
‡Department of Engineering, University of Palermo, 90133 Palermo, Italy
§IMDEA Networks Institute, 28918 Leganes (Madrid), Spain
Email: leonardo.badia@unipd.it, {paolo.castagno,matteo.sereno}@unito.it,
vincenzo.mancuso@ieee.org, marco.ajmone@imdea.org

***Abstract*—Many of today's user applications are both time-critical and computationally intensive. A typical example is provided by assisted- and self-driving systems, where the data collected by onboard sensors must be fused over network computing elements, possibly using artificial intelligence (AI) tools, to accurately reconstruct a vehicle's environment in a sufficiently short time to guarantee safe operations. Our study considers this example, but also covers more general cases, and extends to any system in which independent sources generate time-critical queries for networked services. Obtaining good performance in these cases requires the careful engineering of both communication networks and computing facilities. In addition, when multiple computation facilities are available to run AI processes (in the fog, edge or cloud, or even on the device itself), users running those time-critical and computationally intensive applications experience the dilemma of which remote resource to use so as to obtain results within the limited available time budget. This does not necessarily imply the choice of the fastest servers, as they may end up getting congested by multiple requests. In this paper, we use optimization and game theory to analyze the balance of user updates among remote AI engines, as well as the choice of the intensity of user traffic, trying to optimize the age of information (AoI) that users experience on their time-critical AI-assisted processes. We show that targeting the minimization of AoI leads to non-trivial server selection and data injection policies, and that the unavoidable price of anarchy of systems that enforce a distributed AI server selection can be low, as long as autonomous adaptation of the individual injection rate of the users is properly kept under control.**

***Index Terms*—Age of information, AI, Game Theory, Distributed computing systems, Beyond 5G networks.**

I. Introduction

The digital ecosystem continues to grow in size and capabilities, permeating our societies, and transforming the way we work and live. Especially, communication networks now offer high data rates and low latency, thereby enabling the implementation of quasi-real-time applications [1], [2]. Combined with the diffusion of computing capabilities in the cloud / edge, this paves the road for the integration of artificial intelligence and machine learning (AI/ML) inferences in applications, towards more and more sophisticated services [3]–[5].

The combination of these features makes it possible to think of implementing AI-assisted latency-constrained services, like those permitting the control of complex distributed cyber-physical systems [6]. A paradigmatic example is given by self-driving systems [7], where the data collected by onboard sensors must be fused, possibly using artificial intelligence (AI) tools, to accurately reconstruct a vehicle's environment in sufficiently short time to guarantee safe operations.

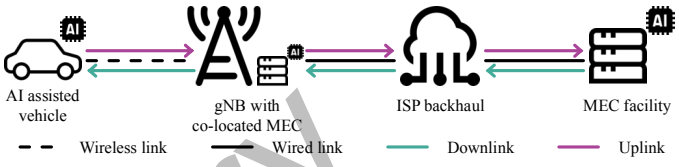

Fig. 1. AI-assisted application workflow

The network and its computing facilities are expected to succeed in coping with the time criticality of interactive queries and system information updates along the entire chain that connects users to network servers and delivers answers/responses/status updates back to interested users, as illustrated in Fig. 1. The complexity of this scenario requires careful management, jointly handling the aspects that influence system performance and user-perceived experience, possibly avoiding centralized control, which would slow down a process that is required to be timely [8], [9]. Moreover, the presence of multiple steps within the decision process (namely, the choice of the server and the frequency of update injection) may suggest a layered approach, where separate decisions are made, possibly with different approaches towards system optimization, thereby resulting in different levels of efficiency.

We consider an AI-assisted cyber-physical system in which users provide information (e.g., gathered by means of local sensors on a vehicle) to one of the available shared AI engines that interpret the sensed data and generate a response (e.g., by issuing warnings to the driver). For such a scenario, the goal of our work is to analyze the importance of (i) server selection policies [10], [11] (e.g., should a "local", low-latency less powerful nearby AI engine or a more powerful "remote" AI engine be used, incurring the associated higher latency?) and (ii) information update rates that feed the AI engines, i.e., how frequently should the inferences on the local or remote AI engines be requested? Note that this last point may possibly

978-3-948377-03-8/19/$31.00 ©2025 ITC

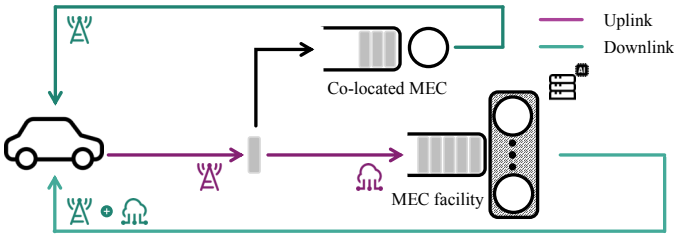

Fig. 2. Queueing model of the system

involve competition with other similar sources [12], [13].

The primary metric we consider is the age of information (AoI) [14], [15], whose original introduction was exactly for vehicular networks [16]. This metric tells how quickly newly gathered pieces of information are incorporated into the AI engine system and used to modify the control of the cyber-physical system (e.g., the assisted vehicle). More in general, we aim at studying how users can parse, with the help of AI, data about a process they observe and need to interpret and act upon, based on the freshest possible information.

Therefore, our AoI evaluation does not just include time needed to convey information updates (e.g., acquired images) to the AI engine, but also the inference time and the time needed to issue and deliver a status update from the AI engine to the user (e.g., a cruise control warning or a confirmation that there is no need to modify the cruise). There exist studies for systems with similar characteristics, typically based on FCFS M/M/1 queues, and sometimes extended to more complex queueing systems [17]–[19]. However, unlike many AoI-based studies [20], [21], our system features likely do not allow the drop of user updates at the server's queue, as different updates might either belong to distinct users and/or carry complementary information that cannot be neglected.

Our reference scenario, shown in Fig. 2, considers the optimization of after-processing-AoI, where the MEC server used for interpreting the data can be chosen as a close edge server, possibly co-located with the base station connecting the user, or a remote MEC facility with higher computational capability but also located farther in the cloud. However, this same choice is taken by multiple equivalent sources (i.e., all the vehicles sending status updates). Therefore this becomes a choice between two paths of different quality, where however the one that is in principle better is typically chosen more frequently; as a result, it may be subject to higher congestion, which may degrade its performance to make it become the worse one. Similar phenomena are known in the literature and connected to the Pigou-Knight-Downs paradox [22]

Hence, building on AoI results for queueing systems, we tackle combined server and inference rate selection strategies in AoI-driven distributed systems. Specifically, we compare selfish and Pareto-optimal choices for both *where* inference should be run and *how frequently*, in order to keep user's awareness about the AI-assisted information acquisition process as fresh as possible, so as to be able to react swiftly, when needed. Our findings reveal that while a fully centralized system achieves the lowest AoI, allowing selfish server selection introduces only modest inefficiencies. Moreover, unsupervised solutions, where server and inference rates are chosen selfishly, emerge as viable alternatives, trading some efficiency for the absence of central orchestration.

The rest of this paper is organized as follows. Section II discusses previous related contributions. Section III presents the system and the approach we used to study AoI. Section IV introduces the different alternatives that we considered for AoI optimization. Section V discusses the features of the system simulator we developed to validate the theoretical findings. Section VI presents and discusses numerical results. Finally, Section VII reports conclusions and ideas for future work.

II. Related Work

The concept of AoI was originally introduced and made popular by the authors of [14], [16]. Since then, in the last decade it has been often applied as a performance evaluation instrument for real-time applications [15], which can lead to devise AoI-oriented scheduling strategies or medium access protocols [9], [23]–[25]. Other directions include the extension of AoI to combine with semantic aspects to obtain, e.g., version AoI, age of context, or age of incorrect information (AoII) [3], [5], [26], [27].

However, one of the earliest lines of research for AoI, already explored in the first seminal papers, was the application to queueing systems, revisiting the traditional formulas of queueing theory through the lens of a new performance metric [18], [19]. Especially, references [13], [28], [29] consider multiple information sources in a single queue, which is one of the main foundations of our analysis.

Another aspect sometimes applied to AoI is *game theory*, seen as a way of distributed systems to achieve a stable operation point. Game theoretic approaches to AoI allow to derive medium access procedures of multiple real-time sources without a central controller [12], [30] as well as exploring security and privacy issues at multiple layers of the protocol stack [31]–[33]. However, it would be possible to apply game theory, leveraging its analytical character, also within the aforementioned line of research involving AoI in queueing systems. Hints of this idea were already contained in [13], since it was derived that the optimal allocation of the queue data injection rate from multiple sources differs from their Nash equilibrium. In [34], this was pushed further to evaluate how the resulting anarchy is lowered by possible correlation in the information content of the sources.

While all of this serves as the preliminary analytical basement of our analysis, the present paper also relates to game theoretical investigations exploring the problem of path selection in a network by distributed agents [11]. In particular, we look at the traditional line of comparison between routing and server selection [35], [36]. Made popular mostly by [37], the evaluations of the inefficiencies of *selfish routing* connect to historical problems like the Pigou–Knight–Downs and Braess paradoxes [22]. Even though selfish routing and the price of anarchy (PoA) are sometimes explored from the

standpoint of queueing theory [38]–[40], there are surprisingly few investigations using AoI as a performance metric, possibly due to the misalignment of the main research in the area, the PoA in selfish routing being popular around the 2000s, whereas AoI was started in the 2010s.

However, we argue that both lines of research can converge in the study of computing architectures based on mobile edge computing (MEC), which are strongly pushed forward by the increase in AI-driven applications. MEC has become a critical enabler for machine learning particularly in scenarios requiring low latency and real-time data processing, such as autonomous driving, smart healthcare, and industrial IoT [8], [10].

In this sense, it is particularly relevant that some recent contributions, such as [41], [42], explore MEC as related to AoI, and in particular [6] adapts the classic queueing formulas for AoI from [13] to the context of processor sharing in a MEC server. Our proposal, while taking inspiration from all these ideas, combines them in an original way, especially focusing on a game theoretic approach for server selection in MEC, as in [11], where, however, the performance metric was task completion probability within a deadline, and not AoI. Instead, in this paper we apply the queueing theoretic evaluations of [13], actually as revised by [6], [28], but on top of these we apply a game theoretic approach as in [34]. All of these contributions, however, just focus on AoI in a single queue, whereas our analysis involves two different levels of decision-making, both influenced by game theory, i.e., server selection and the management of the individual users inside the same queue. This clarifies that, while building on the existing literature, the development is entirely original and, additionally, able to obtain unforeseen conclusions in terms of guidelines for the management of MEC systems.

III. System Model and Analysis of AoI

We consider a number N of distributed sources, acting without coordination, sending status updates about their monitored process. This can represent multiple application scenarios, e.g., autonomous driving, with multiple vehicles reporting ambient information about their surroundings, the state of the traffic an so on. These sources strive to minimize their individual AoI, but the processing of their data is too heavy to be performed locally, and therefore must be offloaded at a mobile edge server, which will return an update for each received request. When multiple MEC facilities are available, the choice of which one to use depends on the overall computing power of the facility, but also on the congestion experienced there. Indeed, even a MEC with abundant computing power may not be convenient if too many other sources choose it.

Thus, we consider two different MEC facilities whose service rates are μ_1 and μ_2, respectively. The problem is interesting when $\mu_2/N < \mu_1 < \mu_2$, i.e., the second facility is faster if only one user is present, but a congested facility 2 is slower than facility 1 with just one user. From a game theory perspective, our offloading choice resembles that of a minority game [43]. To illustrate our reasoning, we will refer in the following to M/M/1 FCFS queues, which are the simplest way to allow for independent sensing nodes to select adjustable injection rates. However, in the numerical evaluations we will validate different queueing systems, showing that similar conclusions still hold true. For an M/M/1 FCFS queue with service rate μ, where a single source injects data with rate λ, the expression of the average AoI, Δ, is [14]:

$$\Delta = \frac{1}{\mu}\left(1+\frac{1}{\rho}+\frac{\rho^2}{1-\rho}\right), \tag{1}$$

where $\rho = \lambda/\mu$ is the queue load factor.

Moreover, if N_i multiple sources with data injection rates $\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,N_i}$ share an M/M/1 FCFS queue with service rate μ_i, $i \in \{1,2\}$, denoting $\rho_{i,j} = \lambda_{i,j}/\mu_i$, the resulting average AoI can be derived by extending (1), as discussed in [28]. In the following, we will adopt the equivalent expression derived in [2], which quantifies $\Delta_{i,j}$ as the average AoI of the jth source on the ith facility as

$$\Delta_{i,j} = \frac{1}{\mu_i}\left[\frac{1-S_{i,j}}{(R_i - S_{i,j}\mathcal{E}_{i,j})(1-\rho_{i,j}\mathcal{E}_{i,j})} + \frac{1}{1-R_i} + \frac{S_{i,j}}{\rho_{i,j}}\right], \tag{2}$$
$$\text{where: } \mathcal{E}_{i,j} = \frac{1+R_i-\sqrt{(1+R_i)^2-4S_{i,j}}}{2S_{i,j}},$$

with $R_i = \sum_j \rho_{i,j}$ and $S_{i,j} = R_i - \rho_{i,j}$. Remarkably, in [13], these two terms use the game theoretic notation of ρ and ρ_{-j}, respectively, since they correspond to the total rate and the total rate without source j, but we deviate from this notation since we better highlight the different facilities with index i. This expression leverages the superposition effect of Poisson arrival processes from different sources.

Thus, we assume that, as the result of their independent decision of which facility to use, the N sources are split such that N_1 nodes process their information at MEC facility 1, and $N_2 = N - N_1$ nodes offload it on MEC facility 2. Symmetry considerations imply that, if any optimization process is applied, all sources sharing the same facility must have the same injection rate, therefore, $\rho_{i,j} = \rho_{i,k}$ for any $1 \le j,k \le N_i$. Hence, we will drop the dependence on the second index and consider ρ_i as a value chosen by all sources offloading on the ith facility, and further evaluate $R_i = N_i\rho_i$ and $S_{i,j} = (N_i-1)\rho_i$. If we substitute these conditions directly into (2), we get an average AoI $\Delta_i(N_i,\rho_i)$ for the N_i sources joining the ith facility, all generating inference tasks with the same individual factor ρ_i as

$$\Delta_i(N_i,\rho_i) = \frac{1}{\mu_i}\left[\frac{(1-(N_i-1)\rho_i)/(1-\rho_i\mathcal{E}_i)}{(N_i\rho_i-(N_i-1)\rho_i\mathcal{E}_i)} + N_i\frac{1+\rho_i-N_i\rho_i}{1-N_i\rho_i}\right],$$
$$\text{with } \mathcal{E}_i = \frac{1+N_i\rho_i-\sqrt{(1+N_i\rho_i)^2-4(N_i-1)\rho_i}}{2N_i\rho_i}. \tag{3}$$

As pointed out in [13], [34], if $N_i > 1$ sources share the same facility i, there are two fundamentally different ways for them to determine their injection rate λ_i and therefore ρ_i. First of all, the sources may choose the *optimal value* of the individual load factor $\rho^*(N_i)$ as the value minimizing the average AoI as reported in (3), where we highlight the dependence on N_i.

However, this does not correspond to the *Nash equilibrium* (NE) of independent selfish sources, which requires a more complex reasoning. At the NE, sources seek for a value of ρ_i where no unilateral improvement is possible [12], [30]. This corresponds to setting the value of $\rho^{(\mathrm{NE})}(N_i)$ to a local maximum of (2) in the individual values of $\rho_{i,j}$.

The difference between the two maximization approaches, i.e., optimal working point $\rho^*(N_i)$ (opt) and Nash equilibrium $\rho^{\mathrm{NE}}(N_i)$, can be described by the following equations:

$$\begin{aligned} \text{opt}: &\ \textbf{set}\ \rho_{i,j} = x \quad \forall j : 1 \le j \le N_i \qquad (4)\\ &\ \textbf{solve}\ \rho_i^* \quad = \mathrm{argmin}_x \Delta_i(x, N_i) \text{ from (2)} \\ \mathrm{NE}: &\ \textbf{set}\ \rho_{i,j} = x \text{ for } j \text{ only} \qquad (5)\\ &\ \textbf{solve}\ \rho_i^{(\mathrm{NE})} = \mathrm{argmin}_x \Delta_{i,j}(x, N_i) \text{ from (1)} \end{aligned}$$

From an operational standpoint, various methods can be used for minimization. We can use the interior point method implemented within Matlab or, thanks to the analytical character of the AoI expressions, compute the first-order derivatives and set them to 0, as the minimizing value is never on the boundaries. The difference is just in the order between computing the argmin and equating $\rho_{i,j}$ for all j. Note that the second line of the NE procedure computes the same $\rho_i^{(\mathrm{NE})}$ for all j and therefore implicitly equates all of them. Also, in the maximization, we should note that R_i actually depends on $\rho_{i,j}$, but $S_{i,j}$ does not, being indeed equal to the sum of the $\rho_{i,k}$ for $k \neq j$. In other words, the NE corresponds to the choice of $\rho_{i,j}$ that minimizes AoI for source j only, assuming that the other sources $k \neq j$ are constant (even though, in reality, all sources follow the same approach, and in the end they get the same $\rho_{i,j}$ anyway). This leads to a *tragedy of the commons* effect [44] that gives $\rho_{i,j} = \rho^{(\mathrm{NE})}(N_i) \ge \rho^*(N_i)$, with equality holding only in the case of an individual source, i.e., $N_i = 1$, in which case both approaches fall down to (1).

The last result can be interpreted as follows [13]. The best choice of ρ in (1) to minimize AoI when a source is alone is neither to never update ($\rho = 0$) nor to clog the facility ($\rho = 1$), but to a value ρ^* in between (which in an FCFS M/M/1 queue is found as 0.531 [14]). If N_i sources are sharing the facility, they cannot simply choose ρ^*/N_i, which would give them the same total offered load as before, but they need to increase from this value because they only get a fraction $1/N_i$ of the updates. Still, they stay away from congestion and choosing $R_i = 1$ as this will give infinite AoI. However, the marginal increase of $\rho_{i,j}$ depends on whether the sources take an optimal (virtually coordinated) or a selfish approach. The optimal choice corresponds to assuming that any increase in $\rho_{i,j}$ is mirrored by all other sources; thus, one ought to simply choose the optimum point in (3).

In contrast, the selfish approach takes the myopic standpoint of increasing the individual $\rho_{i,j}$ assuming that the other $\rho_{i,k}$ does not change (in other words, the congestion caused by an increase is underestimated by a factor $1/N_i$). This suboptimal approach is the only one that can be supported as a NE. Optimizing (3) does not provide an NE as each source has an incentive for a unilateral deviation since, from a selfish perspective, they can choose a higher $\rho_{i,j}$. These two approaches to choosing the individual $\rho_{i,j}$, once the number N_i of sources on the ith facility is set, ultimately reflect the problem of server selection, as argued in the following.

IV. Server and Inference Rate Selection

The average AoI achieved at a given MEC facility depends on the global load offered by all sources insisting there. Therefore, the AoI is determined by two factors: (i) how many sources connect to the facility and (ii) how frequently the sources generate inference tasks.

If homogeneous sources generate homogeneous inference requests, we can assume that N_i sources connected to the same facility i use the same inference rate (thus adopt the same ρ_i) and therefore observe the same average AoI $\Delta_i(N_i, \rho_i)$. However, sources can operate to achieve a global optimum in terms of AoI, or rather act selfishly. This means that once N_i is set, the value of Δ_i can be computed through (3) but obtaining two different values: $\Delta^*(N_i)$ as the minimal AoI achieved by N_i sources obtained from the optimal $\rho^*(N_i)$, or $\Delta^{(\mathrm{NE})}(N_i)$ as the value corresponding to N_i selfish sources that consequently choose $\rho^{(\mathrm{NE})}(N_i)$.

With the notation defined above, minimizing the overall average AoI implies solving the following problem:

$$\begin{aligned} \text{minimize} \quad & \frac{N_1\,\Delta_1\,(N_1,\rho_1) + N_2\,\Delta_2\,(N_2,\rho_2)}{N}, \qquad (6)\\ \text{s.t.:} \quad & N_1 + N_2 = N, \\ & 0 < \rho_j < 1, \quad j \in \{1,2\}. \end{aligned}$$

The formulated problem is, in general, non-linear and non-convex because of the expression of the AoI reported in (3). However, the separation of the variables implies that one can simply find the solution for an integer n to

$$\begin{aligned} \text{minimize} \quad & \frac{n\,\Delta_1\big(n,\rho^*(n)\big) + (N-n)\,\Delta_2\big(N-n,\rho^*(N-n)\big)}{N}, \\ \text{s.t.:} \quad & 0 \le n \le N. \end{aligned}$$

To solve the above problem, we can test all possible non-negative values of N_1 and N_2 that sum to N and consider their AoI as derived optimally once the values of N_i are set. As a side note, the value of Δ_i in the problems above depends on i because of the constant $1/\mu_i$ term within the expression, but instead $\rho^*(n)$ does not depend on it, being normalized to μ_i. We indicate this overall optimal solution as "Super-Best," to distinguish it from other approaches containing Nash equilibria and partial optimizations that we describe next.

When a centralized optimization is not enforced, sources independently select which facility to connect to, which determines N_1 and N_2. We treat this as a *dynamic game of complete information* [45], which is a good way to model distributed choices, yet with a different priority order.

In other words, we assume a game played by the sources through two subsequent stages. In *Stage 1*, either facility 1 or 2 is chosen by each source individually and unbeknownst to each other. This is a mere problem of server selection [36]. In

fact, in our physical model, sources are free to move to another facility if they find it more convenient. However, since we are looking for subgame-perfect Nash equilibria [38], we will find an allocation point where no source wants to deviate in Stage 1, that is, change facility. Subsequently, in Stage 2, the sources observe the choices made in Stage 1 and adjust the inference generation rate on their facility of choice, with full awareness of the number of sources with which they are sharing the facility. In practical cases, this number can be inferred from measurements of perceived AoI or from minimal signaling.

This kind of game can be solved through the principle of *sequential rationality* [45], dictating that rational players can anticipate the outcome of subsequent stages and optimize the choices in the earlier ones. In simpler scenarios, this boils down to applying backward induction through Bellman's optimality principle [25]. We remark that we cannot simply use this approach here, since the moves of the players are simultaneous in both stages. Thus, sequential rationality implies that instead of an optimal working point, we play a NE [45]. Starting from Stage 2, as customarily sequential rationality implies to begin from the last stage, we can infer that, once the values of N_1 and N_2 are set, the sources will choose their injection rates as $\rho^{(\mathrm{NE})}(N_i)$, $i \in \{1, 2\}$.

This outcome of Stage 2 can be anticipated in Stage 1 by rational sources, without any ambiguity due to the uniqueness of the NE. In general, finding the NE to play for the resulting game, where the NE outcomes of Stage 2 are brought up to Stage 1, is not trivial. Yet, we can leverage the principle that rational players do not want to unilaterally deviate from a NE [12]. This implies that the overall NEs of the dynamic game ought to be the choice of N_1 and N_2 that satisfies

$$\Delta_1\big(N_1, \rho^{(\mathrm{NE})}(N_1)\big) = \Delta_2\big(N_2, \rho^{(\mathrm{NE})}(N_2)\big)\,. \tag{7}$$

However, note that N_1 and N_2 must be integer numbers, thus in reality the above equality can only be approximated and we can have two configurations that approximate the NE, or, equivalently, two NEs of Stage 1 as the result of condition (7), since the values of N_1 and N_2 can be rounded up or down. Indeed, the idea behind this equation is that if one source sees that the AoI is lower on the other facility, this would give an option for a unilateral deviation that contradicts the NE. In reality, this does not hold true if a granularity of 1 source exists since it may be that

$$\begin{aligned} &\Delta_1\big(N_1, \rho^{(\mathrm{NE})}(N_1)\big) < \Delta_2\big(N_2, \rho^{(\mathrm{NE})}(N_2)\big) \\ &\text{but } \Delta_2\big(N_2, \rho^{(\mathrm{NE})}(N_2)\big) < \Delta_1\big(N_1{+}1, \rho^{(\mathrm{NE})}(N_1{+}1)\big)\,. \end{aligned} \tag{8}$$

In the following numerical evaluations of Section VI, we will assume an implicit equilibrium selection for the better case, but the simulation results may be affected by a granular error due to this. We also remark that in light of the uniqueness of the NE in Stage 2, the NE of the game must also be subgame-perfect [45]. This approach, which supports both the server selection scheme and the injection rate adaptation, being fully distributed, originates from choosing the NE two times in a row (at both stages), and therefore we will refer to it in the following as "Nash-Nash."

However, an intermediate approach is possible between fully distributed and centralized selections for both decisions. If we assume that, once joining facility i, the N_i sources connected to it are controlled to enact a data injection rate that uses $\rho^*(N_i)$ instead of $\rho^{(\mathrm{NE})}(N_i)$, we can change (7) to

$$\Delta_1\big(N_1, \rho^*(N_1)\big) = \Delta_2\big(N_2, \rho^*(N_2)\big)\,. \tag{9}$$

This results in lower AoI (though not necessarily the same $N_1{:}N_2$ split), as $\Delta_i\big(N_i, \rho^*(N_i)\big) < \Delta_i\big(N_i, \rho^{(\mathrm{NE})}(N_i)\big)$.

This approach is denoted in the following as "Nash-Best," since the server selection is still performed in a selfish way, i.e., through the indifference principle. From an implementation standpoint, this approach still requires centralized coordination, but only at the individual MEC facility level, whereas the server selection itself is fully distributed. However, the procedure may still be subject to Pigou-like paradoxes [22].

As a final remark, we mention that we derived the entire computation by leveraging the formulas for an FCFS M/M/1 queue from [28], yet this is by no means restrictive as the same conclusions, as well as the possibility of defining the three approaches of "Super-Best," "Nash-Best," or "Nash-Nash," would still hold true with different expressions for $\Delta_i\big(N_i, \rho(N_i)\big)$. In particular, an interesting element to consider would be the presence of a fixed round trip delay d_i to MEC facility i. In this case, all expressions of average AoI, such as (1) or (3), increase by d_i. The role of network latency when the facility is physically distant is indeed relevant, although often neglected in analytical investigations [46]. For this reason, in the following we will explicitly refer to a scenario where facility 1 is interpreted as a local edge server with relatively small μ_1 but $d_1 \approx 0$, while facility 2 is located in the cloud and more powerful, so that $\mu_2 > \mu_1$ but $d_2 > 0$.

V. Simulator

We developed a C++ event-based simulator to investigate multiple configurations. Simulations aim to evaluate the overall AoI achievable with the three approaches described in Section IV. We also want to expand the study beyond the plain M/M/1 queue by also considering deterministic service times and/or multiple servers in the same MEC facility.

The most relevant events that characterize the system behavior and that correspondingly determine the simulation runs evolution are few: the generation of a query for a facility, its arrival to the facility's queue, the beginning of the query processing, and its end. Events and their sequence in time are handled through an event list, where events happening in the near future are at the head of the list, while others follow farther in the list as the event time increases.

An experiment requires simulating a single queue with an infinite waiting room, a predefined number of processors and service rate, and one or multiple vehicles sending inference requests. The MEC facilities can be simulated in isolation, since once the number of sources N_i is determined and their request rate is known, they work independently. In particular, the simulation results can be used to replace the expression for $\Delta_i(N_i, \rho_i)$ in (3).

TABLE I
NUMERICAL EVALUATION PARAMETERS

Parameter	Value
Number of vehicles	30
Processors (MEC facility 1) — p_1	1
Processing rate (MEC facility 1) — μ_1 [req/ms]	1
Processors (MEC facility 2) — p_2	$[1, 2, 10]$
Processing rate (MEC facility 2) — μ_2 [req/ms]	$[5, \dots 20]$
RTT (Vehicle to MEC facility 1) — d_1 [ms]	$\simeq 0$
RTT (Vehicle to MEC facility 2) — d_2 [ms]	15

Since in all cases we consider either M/M/ or M/D/ queues, we take AI queries as independently generated at the sources at the same rate λ, with Poisson distributed inter-arrival times. Then, based on the generation time, AI queries join the queue, where they are immediately served if any idle processor is available or otherwise appended to the waiting line. When jobs enter service, the processing time is sampled from the probability density function of a random variable characterizing the service distribution, either exponential or deterministic. When multiple processors are available, the serving processor is chosen randomly. After processing a query, the AoI is updated, cumulating the time between the generation time of the previous request and the current time.

Each experiment requires multiple runs of the same configuration until the confidence interval on the mean of the target measure falls within the specified thresholds. Likewise, each simulation runs until the standard deviation on the collected measure falls within the predefined thresholds. For the experiments reported in the next section, we set the run's threshold to 0.5% of the reference measure average, and the one for the overall simulation to 1% of the same metric.

Knowing that the source population is finite and the server selection implies exclusive access to either MEC facility makes it simple to combine measures from different simulations to obtain measures of the overall system. Indeed, the overall average AoI is just weighted over the two different MEC facilities with the numbers N_1 and N_2.

VI. NUMERICAL EVALUATION

We consider a setting in which sources of queries are located on 30 vehicles that move, acquire images, and send them to one of two MEC facilities, asking the AI agent in the MEC to interpret the content of the images. The AI agents return their interpretation and vehicles act accordingly.

We consider that one MEC facility is close to the users (e.g., running as a MEC application in 3GPP network elements in proximity, at round-trip distance $d_1 \simeq 0$), whereas the other facility is reachable in a few milliseconds (e.g., farther apart in the edge of the cellular network, at $d_2 = 15$ ms). In what follows, we will refer to "local" and "remote" facilities to refer to them, and use indexes 1 and 2, respectively.

For the local facility, we only consider a single processor ($p_1 = 1$) with service rate $\mu_1 = 1$ req/ms. The remote facility is more powerful and devotes $p_2 \geq 1$ processors to serve incoming requests. The service rate at the remote MEC facility is $\mu_2 \geq \mu_1$ and is equally divided among all available processors, i.e., the service rate of individual processors is μ_2/p_2. Both local and remote facilities have infinite queueing capacity. Queries sent to AI engines by vehichles are issued according to a Poisson process with rate λ. Table I summarizes the parameters used in our numerical evaluations.

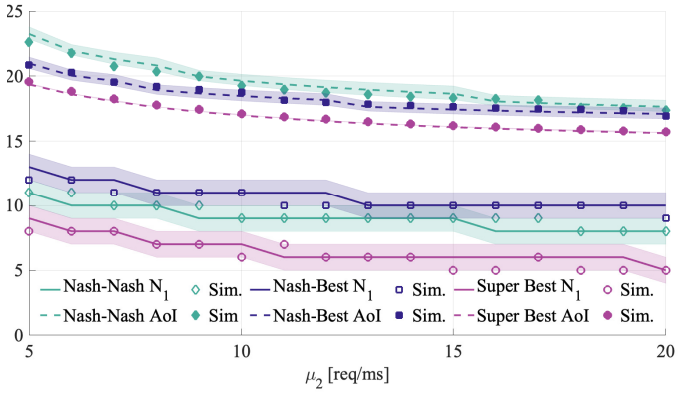

Fig. 3. Number of users at the local server (N_1, bottom three curves, with solid line) and AoI in milliseconds (upper three curves, dashed lines) versus the processing rate at the remote MEC facility μ_2— analysis (lines) and simulation (markers) with three alternative policies. Shaded areas represent the effect of adding/subtracting one user to/from N_1.

Figs. 3 to 7 were obtained for $p_2 = 1$ at the remote server. The figures show numerical results obtained by evaluating Super-Best (optimal), Nash-Best and Nash-Nash solutions computed numerically. They also display simulation points, and, in the case of numerical evaluations of NEs, a shaded area indicates how performance would change by moving one source from a facility to another, due to the granularity of the discrete number of sources, which possibly amplifies oscillations due to simulation noise. Thus, the shaded area allows us to highlight the possible outcomes considering rounding effects when moving from solutions in the continuum—the model—to discrete solutions—the simulation.

Fig. 3 shows the average AoI and the number N_1 of query sources (vehicles) that process AI queries in the local facility, as a function of the remote facility service rate μ_2. It is interesting to see that Nash-Best, which leaves the server selection to the distributed choice of the individual sources, but optimizes the inference update rate, leads to more devices processing information in the local facility, with respect to the Nash-Nash case. Instead, Nash-Nash leads to more devices issuing AI queries to the remote facility, using resources inefficiently, and obtaining the worst average AoI. Among the three approaches, the number of vehicles choosing to process information in the local facility decreases with the increase of the remote facility capacity. By comparing analytical and simulation results, we notice a very good match, with simulation results always within the shaded area (i.e., within the granularity of the rounding). Specifically, the match between the two sets of results is particularly good considering N_1, while slightly more noisy when looking at the average AoI. This effect is easier to identify for Nash-Nash because

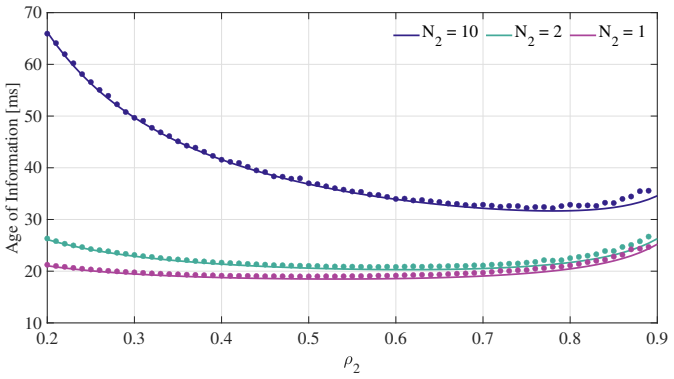

Fig. 4. Mean AoI at the remote facility (N_2= [1, 2, 10]) vs. offered load ρ_2.

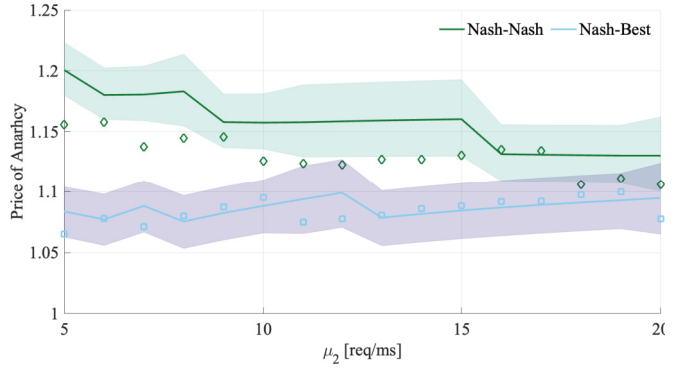

Fig. 5. PoA for the Nash-Best and Nash-Nash policies versus the processing rate at the remote MEC facility μ_2. The solid lines are the numerical evaluation, and the points are the simulation results.

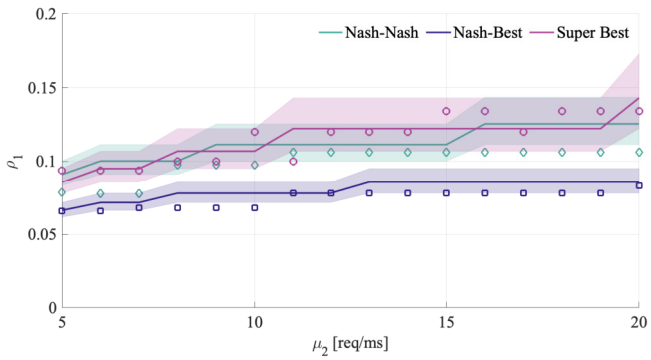

Fig. 6. Offered load by one source at the local facility (1) vs. processing rate at the remote MEC facility μ_2. Solid lines are analytical, shaded areas depict the granularity due to ± 1 source and markers are simulation results.

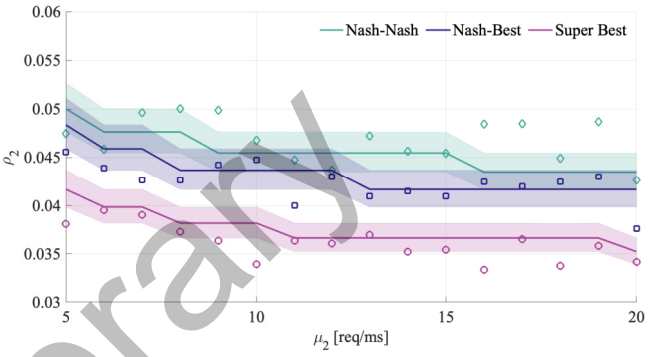

Fig. 7. Offered load by one source at the remote facility (2) vs. processing rate at the remote MEC facility μ_2. Solid lines are analytical, shaded areas gives the granularity due to ± 1 source and the markers are simulation results.

choosing between servers is much easier than finding the exact load that leads to a Nash equilibrium point. Indeed, stochastic fluctuations in service time and offered load at the facilities may lead vehicles to send fewer updates, because of incurring higher delays, than what expected from the theory.

Fig. 4 shows the average AoI that vehicles connected to the remote facility experience as the load increases. The figure reports three curves for 1, 2, and 10 vehicles connected to the remote server. The curves are computed analytically, considering an M/M/1 queue and the random latency to go back and forth from the facility, whereas points next to or on top of the curves represent simulations. The latter fit well the AoI model presented in Section III, as expected. In the figure, it is interesting to see that, for any selected aggregate load of the server, the AoI performance degrades with an increasing number of users. Moreover, the curves shown in the figure are practically flat over large areas, which explains why numerical tools might be prone to errors in the search for NEs and optimal configurations.

Fig. 5 shows the consequent PoA, i.e., the ratio between the average AoI of Nash-Nash or Nash-Best and the optimal allocation (Super-Best) in Fig. 3. Simulations of Nash-Nash show a lower PoA, since the selected issuing rate is slightly lower than the one of the analysis, leading to a lower average number of requests at the processors, either local or remote, and shorter queues. Although PoA is not high for Nash-Nash, as it ranges between 15% and 20%, introducing some control on inference rate generation with Nash-Best helps to decrease AoI substantially. Increasing the remote MEC capacity reduces the impact of server selection on AoI, because more and more vehicles will select the remote facility and the difference between $\rho^{(NE)}(N_i)$ and the optimal $\rho^*(N_i)$ becomes smaller. When the advantage of offloading to the remote facility becomes more evident, the query rate of vehicles becomes less important, since the population of vehicles is constant and also the average AoI, from a certain point onward, deteriorates with the query rate. This explains why the gap between Nash-Best and Nash-Nash diminishes with the capacity of the remote facility. For both Nash-Nash and Nash-Best, a PoA asymptotically converging at 1.1 can be interpreted as the inherent AoI inefficiency of a Pigou-like server selection [22].

Figs. 6 and 7 show the load offered by single sources to the two facilities as the capacity of the remote MEC varies, for the three approaches discussed in this paper. From the two plots, it is evident that the Nash-Nash solution with which a source issues requests at higher pace on all available resources, leading to higher inefficiency. Optimizing inference rate and letting vehicles choose their MEC leads to higher exploitation of the remote facility and lower load of local processing, per each vehicle. Eventually, Super-Best optimizes the offered traffic, preventing the remote facility from congestion, through both an optimal selection of the injection rate and also sending

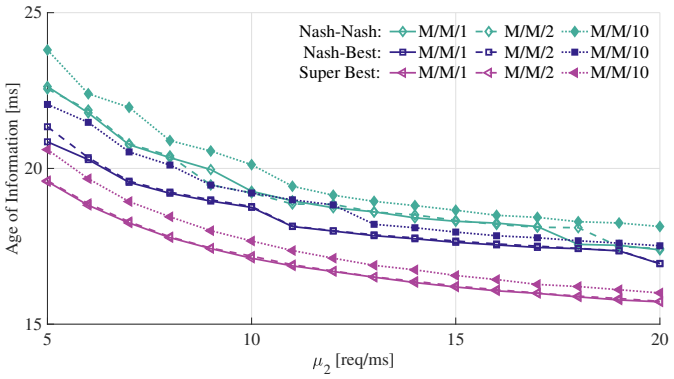

Fig. 8. Age of Information of single and multiple server queues (1, 2 or 10 servers) versus the processing rate at the remote MEC facility μ_2, with exponential service rates.

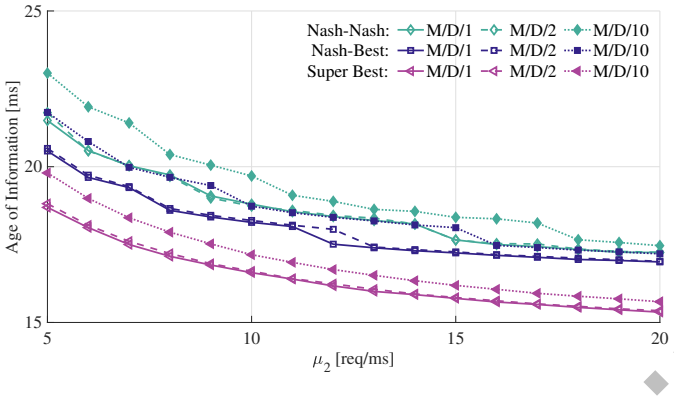

Fig. 9. Average AoI of single and multiple server queues (1, 2 or 10 servers) versus the processing rate at the remote MEC facility μ_2, with deterministic service rates.

requests to the local server. This behavior is possible because sources enforce query rate adaptation, so that when, e.g., the number of sources on a facility increases, the load per source tends to diminish. Indeed, although not shown in the figures, we have observed that the aggregate load observed at the two servers slowly grows at the local server and decreases at the remote server as its capacity increases, with total utilization values quite flat, of the order of 0.7-0.8 for Super-Best and Nash-Best, and 0.9-0.95 for Nash-Nash.

In Fig. 8, we plot the average AoI obtained with different settings at the remote facility versus the processing rate at the remote MEC, μ_2. For a given remote facility's capacity μ_2, we increase the number of available processors, decreasing their individual capacity accordingly. Overall, the trends remain the same, with the Nash-Nash solution providing higher AoI and Super-Best providing vehicles with fresher information. However, it is worth noticing that the AoI increases as the number of processors increases. This phenomenon becomes more relevant as the number of processors increases; indeed, a higher number of processors allows the reduction of the waiting times of the first p_2 arriving jobs, but the subsequent service times will be substantially longer. In other words, one single fast processor is more efficient than many parallel slow processors also in terms of AoI.

In Fig. 9, we plot curves analogous to those in 8 but in this case we use deterministic service times rather than exponential. This reduces the variability of service latency, which yields lower average sojourn times for queries in the server queue. Hence, the average AoI decreases as well if compared to results in Fig. 8.

Overall, the following takeaway messages can be drawn from the results. When server selection must consider facilities of comparable capacity (μ_1 being no more than one order of magnitude lower than μ_2), the main effect differentiating the three approaches is actually the injection rate of inference requests, and a Nash-Best performance ends up in being relatively close to the optimum, with a PoA below 1.1 indicating an increase in average AoI of about 10%. This seems to suggest that a distributed AI server selection is overall feasible, as long as the individual injection rate of the sources joining the same MEC facility is locally controlled to avoid tragedy of the commons phenomena, where sources overload the system. The latter can be achieved with some local coordination without resorting to an overall centralized control.

Instead, when the more powerful facility (MEC 2, the remote server) has significantly higher capacity, i.e., $\mu_2 \gg \mu_1$, most sources would like to join facility 2, which actually worsens the effect of anarchy even for a Nash-Best approach. In this case, indeed, the main contribution to PoA is the Pigou-like selection, possibly subject to paradoxes when the choice of one facility tends to become dominant over the other. However, even in this case, where the Nash-Nash and the Nash-Best approaches tend to converge, the PoA is still limited and below 1.1, which once again confirms that a distributed server selection may be acceptable in terms of average AoI.

VII. Conclusions

We analyzed server selection and inference rate adaptation for users connected to AI-based networked services provided by in-network computing facilities. We investigated the role of resource orchestration on the achievable AoI, comparing centralized, distributed, and mixed approaches, considering the possible use case of assisted driving services. Our results show that the system behavior is complex, but users may selfishly choose their preferred processing location, as long as they cooperate to use shared resources optimally. The analysis of the PoA shows that a fully distributed approach incurs some inefficiency, but the overall cost is limited and decreases with the amount of available computing resources.

The results presented shed light on the design and implementation of controls of cyber-physical systems through networked computing facilities. They enable service providers to understand costs and benefits of coordinated and uncoordinated access to shared computational resources.

Acknowledgments

This work has been supported by the Italian National Recovery and Resilience Plan (NRRP), partnership on "Telecommunications of the Future" (PE0000001 - program "RESTART", under subprograms Net4Future, (Cascade project REFERENCES), and S2 SUPER – Programmable Networks, Cascade

project PRISM - CUP: C79J24000190004.
This work has been also partially supported by project TUCAN6-CM (TEC-2024/COM-460), funded by CM, the Region of Madrid, Spain (ORDEN 5696/2024).

REFERENCES

[1] T. Hao, K. Hwang, J. Zhan, Y. Li, and Y. Cao, "Scenario-based AI benchmark evaluation of distributed cloud/edge computing systems," *IEEE Trans. Comp.*, vol. 72, no. 3, pp. 719–731, 2022.
[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. on Selected Areas in Comm.*, vol. 39, no. 5, pp. 1183–1210, 2021.
[3] E. Uysal *et al.*, "Semantic communications in networked systems: A data significance perspective," *IEEE Netw.*, vol. 36, no. 4, 2022.
[4] L. Das, B. M. Sahoo, A. Rana, K. Dadhich, S. Sharma, and S. A. Yadav, "Application of AI & ML in 5G communication," in *Paradigms of Smart and Intelligent Communication, 5G and Beyond*. Springer, 2023, pp. 149–170.
[5] M. Xu, D. Niyato, H. Zhang, J. Kang, Z. Xiong, S. Mao, and Z. Han, "Joint foundation model caching and inference of generative AI services for edge intelligence," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2023, pp. 3548–3553.
[6] Z. Tang, Z. Sun, N. Yang, and X. Zhou, "Age of information of multi-user mobile edge computing systems," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1600–1614, 2023.
[7] D. Li, J. Zhang, and G. Liu, "Autonomous driving decision algorithm for complex multi-vehicle interactions: An efficient approach based on global sorting and local gaming," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6927–6937, 2024.
[8] P. Han, B. Liu, Y. Liu, and L. Guo, "Cell-less offloading of distributed learning tasks in multi-access edge computing," *IEEE Trans. Mob. Comp.*, vol. 23, no. 12, pp. 14 377–14 395, 2024.
[9] O. T. Yavascan and E. Uysal, "Analysis of slotted ALOHA with an age threshold," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1456–1470, 2021.
[10] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective," *Comp. Netw.*, vol. 182, p. 107496, 2020.
[11] V. Mancuso, L. Badia, P. Castagno, M. Sereno, and M. Ajmone Marsan, "Effectiveness of distributed stateless network server selection under strict latency constraints," *Comp. Netw.*, p. 110558, 2024.
[12] L. Badia, "Age of information from two strategic sources analyzed via game theory," in *Proc. IEEE Int. Worksh. Comp. Aided Model. Design Commun. Links Netw. (CAMAD)*, 2021, pp. 1–6.
[13] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. Inf. Th.*, vol. 65, no. 3, pp. 1807–1827, 2019.
[14] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, 2012, pp. 2731–2735.
[15] A. Kosta, N. Pappas, V. Angelakis *et al.*, "Age of information: A new concept, metric, and tool," *Found. Trends Netw.*, vol. 12, no. 3, pp. 162–259, 2017.
[16] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. IEEE SECON*, 2011, pp. 350–358.
[17] 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. Th.*, vol. 65, no. 12, pp. 8305–8324, 2019.
[18] J. P. Champati, R. R. Avula, T. J. Oechtering, and J. Gross, "Minimum achievable peak age of information under service preemptions and request delay," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1365–1379, 2021.
[19] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "The age of information in a discrete time queue: Stationary distribution and non-linear age mean analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1352–1364, 2021.
[20] A. Arafa, R. D. Yates, and H. V. Poor, "Timely cloud computing: Preemption and waiting," in *Proc. IEEE Allerton Conf. Commun. Contr. Comput.*, 2019, pp. 528–535.
[21] E. Najm and E. Telatar, "Status updates in a multi-stream M/G/1/1 preemptive queue," in *Proc. IEEE Conf. Comp. Commun. Worksh. (INFOCOM Wkshps)*, 2018, pp. 124–129.
[22] J. Morgan, H. Orzen, and M. Sefton, "Network architecture and traffic flows: Experiments on the Pigou–Knight–Downs and Braess paradoxes," *Games Econ. Behav.*, vol. 66, no. 1, pp. 348–372, 2009.
[23] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Trans. Inf. Th.*, vol. 62, no. 4, pp. 1897–1910, 2016.
[24] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2637–2650, 2018.
[25] A. Munari and L. Badia, "The role of feedback in AoI optimization under limited transmission opportunities," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 1972–1977.
[26] B. Buyukates, M. Bastopcu, and S. Ulukus, "Version age of information in clustered gossip networks," *IEEE J. Sel. Areas Inf. Th.*, vol. 3, no. 1, pp. 85–97, 2022.
[27] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2215–2228, 2020.
[28] M. Moltafet, M. Leinonen, and M. Codreanu, "On the age of information in multi-source queueing models," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5003–5017, 2020.
[29] N. Akar and O. Dogan, "Discrete-time queueing model of age of information with multiple information sources," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14 531–14 542, 2021.
[30] K. Saurav and R. Vaze, "Game of ages in a distributed network," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1240–1249, 2021.
[31] G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier, and A. Ephremides, "Information freshness over an interference channel: A game theoretic view," in *Proc. IEEE INFOCOM*, 2018, pp. 908–916.
[32] Y. Yang, X. Wei, R. Xu, L. Peng, and L. Liu, "Game-based channel access for AoI-oriented data transmission under dynamic attack," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8820–8837, 2021.
[33] V. Bonagura, S. Panzieri, F. Pascucci, and L. Badia, "Strategic interaction over age of incorrect information for false data injection in cyber-physical systems," *IEEE Trans. Contr. Netw. Syst.*, vol. 12, no. 1, pp. 872–881, Mar. 2025.
[34] L. Badia and L. Crosara, "Correlation of multiple strategic sources decreases their age of information anarchy," *IEEE Trans. Circ. Syst. II: Expr. Briefs*, vol. 71, no. 7, pp. 3403–3407, 2024.
[35] C. E. Bell and S. Stidham Jr, "Individual versus social optimization in the allocation of customers to alternative servers," *Manag. Sc.*, vol. 29, no. 7, pp. 831–839, 1983.
[36] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On selfish routing in Internet-like environments," in *Proc. ACM SIGCOMM*, 2003, pp. 151–162.
[37] T. Roughgarden and É. Tardos, "How bad is selfish routing?" *J. ACM*, vol. 49, no. 2, pp. 236–259, 2002.
[38] M. Haviv and T. Roughgarden, "The price of anarchy in an exponential multi-server," *Op. Res. Lett.*, vol. 35, no. 4, pp. 421–426, 2007.
[39] J. Anselmi and B. Gaujal, "Optimal routing in parallel, non-observable queues and the price of anarchy revisited," in *Proc. IEEE Int. Teletraffic Cong. (ITC 22)*, 2010, pp. 1–8.
[40] G. Gilboa-Freedman, R. Hassin, and Y. Kerner, "The price of anarchy in the Markovian single server queue," *IEEE Trans. Autom. Contr.*, vol. 59, no. 2, pp. 455–459, 2013.
[41] H. Li, J. Zhang, H. Zhao, Y. Ni, J. Xiong, and J. Wei, "Joint optimization on trajectory, computation and communication resources in information freshness sensitive MEC system," *IEEE Trans. Veh. Tech.*, vol. 73, no. 3, pp. 4162 – 4177, 2024.
[42] Y. Dong, H. Xiao, H. Hu, J. Zhang, Q. Chen, and J. Zhang, "Mean age of information in partial offloading mobile edge computing networks," *arXiv preprint arXiv:2409.16115*, 2024.
[43] S. Ranadheera, S. Maghsudi, and E. Hossain, "Computation offloading and activation of mobile edge computing servers: A minority game," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 5, pp. 688–691, 2018.
[44] L. Prospero, R. Costa, and L. Badia, "Resource sharing in the Internet of things and selfish behaviors of the agents," *IEEE Trans. Circ. Syst. II: Expr. Briefs*, vol. 68, no. 12, pp. 3488–3492, 2021.
[45] M. J. Osborne, *A course in game theory*. MIT Press, 1994.
[46] V. Mancuso, P. Castagno, L. Badia, M. Sereno, and M. Ajmone Marsan, "Optimal allocation of tasks to networked computing facilities," in *Proc. Int. Conf. Anal. Stoch. Model. Techn. Appl. (ASMTA)*, 2024, pp. 33–50.