Exact Analytical Model of Age of Information in Multi-Source Status Update Systems With Per-Source Queueing

Ege Orkun Gamgam and Nail Akar, *Member, IEEE*

***Abstract*—We study a multisource status update system with Poisson information packet arrivals and exponentially distributed service times. The server is equipped with a waiting room holding the freshest packet from each source referred to as single buffer per-source queueing (SBPSQ). The sources are assumed to be equally important, i.e., (nonweighted) average Age of Information (AoI) or average age violation probability are used as the information freshness metrics to optimize for, and subsequently, two symmetric SBPSQ-based scheduling policies are studied in this article, namely, first source first serve (FSFS) and the earliest served first serve (ESFS) policies. By employing the theory of Markov fluid queues (MFQs), an analytical model is proposed to obtain the exact distribution of the AoI for each source when the FSFS and ESFS policies are employed at the server. Additionally, a benchmark scheduling-free scheme named single buffer with replacement (SBR), which uses a single buffer to hold the freshest packet across all sources, is also studied with a similar but less complex analytical model. We comparatively study the performance of the three policies through numerical examples in terms of the average AoI and the age violation probability averaged across all sources, in a scenario of sources possessing different traffic intensities but sharing a common service time.**

***Index Terms*—Age of Information (AoI), buffer management, Markov fluid queues (MFQs), multisource queueing model, scheduling.**

I. Introduction

TIMELY delivery of status packets has been gaining utmost importance in Internet of Things (IoT)-enabled applications [1], [2] where the information freshness of each IoT device at the destination (or monitor) is crucial, especially for applications requiring real-time control and decision making. A widely studied process for quantifying the freshness of data is the Age of Information (AoI) process, which stands for the time elapsed since the generation time of the last status update packet received at the monitor. More formally, the AoI at time t is defined as the random process $\Delta(t) = t - U(t)$ where $U(t)$ denotes the timestamp of the last status packet received at the monitor. On the other hand, the peak AoI process is obtained by sampling the AoI process just before packet reception instants. Performance metrics involving AoI processes were first introduced in [3] for a single-source M/M/1 queueing model and since then a surge of studies followed in the context of a wide range of information update systems; see the recent surveys [4] and [5] on AoI and the references therein. AoI in multisource models sharing a single server or multiple servers have also been recently studied in several works; see for example [6]–[8]. Traffic comprising status update packets, for which AoI-driven performance metrics are most crucial for performance, as opposed to conventional metrics such as delay, loss, or throughput that are used for conventional data traffic, is referred to as age-sensitive traffic [9]. Performance modeling and optimization of status update systems carrying age-sensitive traffic is the focus of this article.

In most existing studies, the focus is mainly toward the characterization of the average AoI and average peak AoI metrics since the analysis of distributional properties of AoI or peak AoI is not straightforward [1]. The average AoI is defined as the long-term time average of the AoI process $\Delta(t)$ and it has been recently investigated in various IoT settings [10]–[15]. In [10], the age–energy tradeoff for a single source IoT system is studied by deriving closed form expressions of both average AoI and average power consumption of the IoT device. Zhou and Saad [11] studied the optimal update policy that minimizes the average AoI under an energy constraint imposed on the IoT devices. Multicast transmissions with deadlines are investigated in [12] for a real-time IoT network. A dense IoT monitoring system with CSMA-type random access is considered in [13], where a closed-form expression for per-device average AoI is obtained. Timeliness in UAV-assisted IoT networks has also been recently studied in [14] and [15] by investigating the average AoI.

In addition to the time average of AoI, obtaining its steady-state distribution allows one to perform more comprehensive analysis which may be useful in IoT-related scenarios. As an example, for IoT services having stringent timeliness and reliability requirements, occurrences of rare but extremely catastrophic events (such as a complete failure due to a very large AoI) must also be thoroughly considered for which the analysis of the average AoI is not solely enough [16]. Such motivations led to various recent studies [17]–[19] focusing on distribution-related performance metrics, such as age violation

Manuscript received 2 November 2021; revised 24 February 2022 and 12 April 2022; accepted 13 May 2022. Date of publication 27 May 2022; date of current version 7 October 2022. *(Corresponding author: Nail Akar.)*

Ege Orkun Gamgam is with the Communication and Information Technologies Division, ASELSAN Inc., 06800 Ankara, Turkey, and also with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: gamgam@ee.bilkent.edu.tr).

Nail Akar is with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: akar@ee.bilkent.edu.tr).

Digital Object Identifier 10.1109/JIOT.2022.3178657

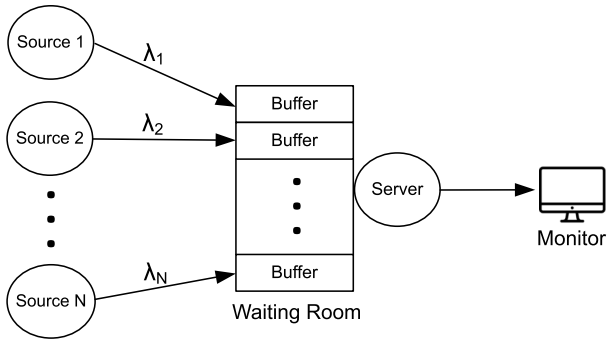

Fig. 1. Per-source buffering system where a remote monitor is updated by N information sources through a single server.

probability or higher order moments, all of which can be easily calculated once the steady-state AoI distribution is obtained.

Scheduling conventional data traffic in wireline and wireless networks has long been studied in the communications literature with real-life implementations. For example, the proportional fair network resource allocation concept proposed first by Kelly [20] that attempts to strike a balance between fairness and efficiency through the maximization of the sum of the logarithms of user throughputs, has successfully been used for scheduling in cellular wireless networks [21] and also in wireless LANs and ad hoc networks [22]. However, research on scheduling age-sensitive traffic in status update systems is relatively immature. Bedewy *et al.* [23] studied the problem of minimizing the average sum AoI in a multisource system and shows that the maximum age first (MAF) scheduling policy provides the best age performance among all scheduling policies whereas Joo and Eryilmaz [24] proposes an age-based scheduler that uses age along with the interarrival times of incoming packets to make scheduling decisions. Kadota *et al.* [7], [25] proposed nearly optimal age-aware and age-agnostic schedulers for an asymmetric (source weights are different and service times are not necessarily identically distributed) discrete-time wireless network with a base station serving multiple age-sensitive traffic streams with per-source queueing. Our focus in this article is on age-agnostic buffer management and scheduling schemes for age-sensitive traffic and the development of exact analytical models for scheduling in multisource systems for comparative performance assessment.

In this article, we consider an information update system which consists of N sources each of which asynchronously samples an independent stochastic process and subsequently sends these samples in the form of status update packets to a single remote monitor through a server as shown in Fig. 1. Information packets from each source contain sensed data along with a timestamp. The packets from source-$n, n = 1, 2, \ldots, N$, are generated according to a Poisson process with rate λ_n, which is also considered in various IoT settings, such as [13], [16], [18], and [26]. Generated packets are immediately forwarded to the server with a waiting room (queue) which can contain at most one packet (the freshest) from each source. Therefore, a packet waiting in the queue is replaced with a new fresh packet arrival from the same source. This buffer management is called single buffer per-source queueing (SBPSQ). The server is responsible for sending the information packets to the monitor through a communication network which introduces a random service time that is exponentially distributed with parameter μ_n for source-n. A new packet arrival immediately starts to receive service if the server is found idle. On the other hand, SBPSQ needs to be accompanied by a scheduling policy since the server is to choose a source packet among the waiting sources upon a service completion. As an example (also see [26]), consider a status monitoring system for a vehicle where sensors of various types (acceleration, proximity, tire pressure, etc.) continuously generate information packets to be delivered wirelessly to a remote destination such as other vehicles in the vicinity or an information fusion center for further processing. Ideally, the destination would receive packets from each sensor immediately after they are generated but this is often limited due to scarce network resources in practical settings, where the generated packets may get queued at an IoT gateway while waiting for their transmission turns. Thus, it is important to develop effective queueing/scheduling policies employed at the gateway. For maximizing the throughput, one could consider first come first serve (FCFS)-based queue management techniques holding multiple packets from the same sensor but such techniques are known to work poorly for AoI due to increased waiting time of packets at the gateway [4]. On the other hand, keeping only the most recent status packet of each sensor, i.e., SBPSQ-based buffer management, is highlighted in [27] for reducing AoI, which also limits the need for large buffers at the gateway.

In this setting, we study the following three queueing/scheduling policies employed at the server.

1) In the first source first serve (FSFS) policy for SBPSQ, also studied in [28] for the case of two sources and a focus on average AoI only, is similar to an FCFS system except that when a new packet arrival belonging to source-n replaces a staler packet in the queue, the service order of that particular source stays the same. If the source-n packet finds its source buffer empty, then its service order will follow the other sources in the waiting room as in FCFS.
2) We propose the earliest served first serve (ESFS) policy for SBPSQ for which the server selects a source (with an existing packet in the queue) that has not received service for the longest duration since the previous selection instant of that particular source. In the ESFS policy, the server locally holds an ordered list of sources based on their last selection instants for service. While choosing a source to serve, this ordered list is the only input for the ESFS policy in contrast with age-based approaches that also take into account of the timestamps of information packets in the queue or the instantaneous AoI values at the destination while making scheduling decisions.
3) For benchmarking purposes, we also consider a server with a one-packet buffer shared by all sources, which is studied as last come–first serve (LCFS) with *preemption only in waiting* (LCFS-W) policy in [26] but with emphasis only on the average AoI. In this setting, a packet waiting in the buffer is replaced with a new

packet arrival from any source. Upon a service completion, the packet held in the buffer (if it exists) starts to receive service. In this article, we refer to this scheduling-free queueing policy as the single buffer with replacement (SBR) policy.

The main contributions of this article are the following.

1) Based on the theory of Markov fluid queues (MFQs) [29], [30], we propose novel analytical models within a unifying framework to numerically obtain the exact distributions of AoI processes for multisource status update systems for FSFS, ESFS, and SBR queueing/scheduling policies. This enables us to obtain the AoI-driven performance metrics of interest such as the average AoI or average AoI violation probability.
2) Although the analytical models for multisource status update systems deploying FSFS, ESFS, and SBR are highly mathematical, they lead to stable numerical algorithms for which we have produced a MATLAB code. For researchers and practitioners working on AoI analysis and optimization, this source code is made publicly available at IEEE DataPort [31].
3) We study and compare the performance of the three buffer management and scheduling policies under several system load scenarios where the sources may have different traffic intensities but a common service time. Through numerical examples, we show that the proposed age-agnostic ESFS policy, which is quite easy to implement, outperforms the FSFS and SBR policies in terms of the average AoI and the age violation probability averaged across all sources, i.e., symmetric source AoI requirements. Therefore, SBPSQ-based buffer management accompanied by the age-agnostic ESFS scheduling offers a promising and simple-to-implement solution for traffic management involving age-sensitive traffic when one seeks to minimize the nonweighted average AoI or average age violation probability.

The remainder of this article is organized as follows. In Section II, related work is given. In Section III, MFQs are briefly presented. In Section IV, we formally describe the proposed analytical method for obtaining the exact per-source distribution of the AoI process for all three studied policies. In Section V, the proposed method is verified with simulations and a comparative analysis of the policies is provided under several scenarios. Finally, we conclude in Section VI.

II. Related Work

There have been quite a few studies on queueing-theoretic AoI analysis for multisource setups when the updates have random service times. The first study on multiple sources sharing a single queue appeared in [32], where the authors derived the average AoI for an M/M/1 FCFS queue. This work is extended in [26] in which the authors studied an M/M/1 queue FCFS service as well as LCFS queues under preemptive and nonpreemptive with replacement policies using the stochastic hybrid systems (SHSs) technique. A nonpreemptive M/M/1/m queue is revisited in [33], where the authors obtained the average AoI expressions. Moltafet *et al.* [34] independently derived the average AoI for the M/M/1 FCFS model studied in [32] and also provided approximate expressions for a multisource M/G/1 queue. In [35], the peak AoI was studied for multisource M/G/1 and M/G/1/1 systems with heterogeneous service times. Najm and Telatar [36] derived closed-form expressions for the average AoI and peak AoI in a multisource M/G/1/1 queue by extending the single source age analysis in [37]. Moltafet *et al.* [38] considered three source-aware packet management policies in a two-source system for which they obtained the per-source average AoI for each policy using SHS. Arafa *et al.* [39] investigated a multisource status updating system for which the multiple threshold-based scheduling policies along with the closed-form expressions for the AoI have been derived. In another line of studies [40]–[42], the AoI analysis of multiple sources with different priorities have been considered under several packet management policies. For distributional properties, Abd-Elmagid and Dhillon [43] studied nonpreemptive and preemptive policies for which the moment generating function (MGF) of AoI is obtained using SHS framework. Moltafet *et al.* [44] considered the preemptive and blocking policies in a bufferless two-source system deriving the per-source MGF of AoI. Doğan and Akar [45] studied the distributions of both the AoI and peak AoI in a preemptive bufferless multisource M/PH/1/1 queue allowing arbitrary and probabilistic preemption among sources.

The most relevant existing studies to this article are the ones that study the analytical modeling of SBPSQ systems. The benefits of SBPSQ are shown in [27] in terms of lesser transmissions and reduced per-source AoI. Moltafet *et al.* [28] obtained the average AoI expressions using SHS techniques for a two-source M/M/1/2 queueing system in which a packet in the queue is replaced only by a newly arriving packet of same source. Moltafet *et al.* [46] derived the per-source MGF of AoI in a two-source system for the nonpreemptive and self-preemptive policies, the latter being a per-source queueing technique. Akar and Karasan [47] proposed an analytical model based on absorbing Markov chains for probabilistic scheduling in an SBPSQ-based status update system with two sources only, when the minimization of weighted average AoI is sought.

III. Markov Fluid Queues

In fluid queue models, a fluid acts as the input to and output of a buffer [48]. Although a fluid may represent different quantities depending on the system of interest, in this article, it represents the AoI in status update systems. In particular, MFQs are described by a joint Markovian process $\boldsymbol{X}(t) = (X(t), Z(t))$ with $t \geq 0$ where $X(t)$ represents the fluid level of the process and $Z(t)$ is the modulating continuous-time Markov chain (CTMC) with state space $\boldsymbol{S} = \{1, 2, \ldots, K\}$. When $Z(t) = i \in \boldsymbol{S}$, then the rate of fluid change (or drift) of the fluid level process $X(t)$ at time t is r_i. In MFQs, the fluid level cannot take negative values, i.e., when $Z(t) = i$ and $X(t) = 0$ for $r_i < 0$, the fluid level $X(t)$ is assumed to stick to the boundary at zero. In the simplest type of MFQ, referred to as single-regime MFQ (SRMFQ), the infinitesimal generator of $Z(t)$ and drift values are constant throughout the whole buffer space [48], [49]. Mathematically, two matrices completely characterize the SRMFQ, i.e., $\boldsymbol{X}(t) \sim$ SRMFQ$(\boldsymbol{Q}, \boldsymbol{R})$ where

the infinitesimal generator of $Z(t)$ is denoted by $\boldsymbol{Q}$ and the drift matrix $\boldsymbol{R} = \mathbf{diag}\{r_1, r_2, \ldots, r_K\}$. The steady-state joint cumulative density function (cdf) vector $\boldsymbol{F}(x)$ is defined in [48] as follows:

$$F_i(x) = \lim_{t\to\infty} \Pr\{X(t) \le x,\ Z(t) = i\} \tag{1}$$

$$\boldsymbol{F}(x) = \begin{bmatrix} F_1(x) & F_2(x) & \cdots & F_K(x) \end{bmatrix} \tag{2}$$

where the authors studied a data-handling system by a SRMFQ model with infinite buffer capacity. In particular, they showed that the following differential equation holds for the cdf vector $\boldsymbol{F}(x)$:

$$\frac{\mathrm{d}}{\mathrm{d}x}\boldsymbol{F}(x)\boldsymbol{R} = \boldsymbol{F}(x)\boldsymbol{Q} \tag{3}$$

along with the boundary condition

$$F_i(0) = 0, \quad r_i > 0, \quad i \in \boldsymbol{S} \tag{4}$$

implying the fluid level cannot stay at zero when the drift is positive. Using (3) and (4), they obtained the stationary solutions by employing a spectral expansion approach. This analysis is then extended in [49] to the SRMFQ with a finite buffer capacity B, where (3) and (4) similarly hold along with an additional boundary condition, given as $F_i(B) = 0$ with $r_i < 0$, implying the fluid level cannot stay at level B when the drift is negative.

SRMFQs are generalized in [50] to feedback MFQs, or also called multiregime MFQs (MRMFQs) in [30], where the buffer is partitioned into a finite number of regimes (or intervals) and the fluid behavior in each of these regimes and the boundaries separating the regimes is allowed to be different than each other. In particular, each regime and boundary has its own infinitesimal generator and drift matrix. In [50], an additional set of boundary conditions for the MRMFQ are shown to hold at the boundaries between adjacent regimes and the stationary solutions are obtained using the spectral expansion approach similar to [48]. Kankaya and Akar [30] studied numerically stable and efficient matrix-analytical methods to provide a general solution to MRMFQs while considering more general boundary behaviors than [50].

The MFQ of interest in this article is slightly different than a SRMFQ and it can be categorized as a special case of MRMFQs. In particular, when the fluid level $X(t) > 0$, it behaves according to a single generator $\boldsymbol{Q}$ similar to a SRMFQ, whereas it behaves according to another generator $\tilde{\boldsymbol{Q}}$ when $X(t) = 0$. The drift values, however, are common for both $X(t) > 0$ and $X(t) = 0$. In this article, we focus only on this generalized SRMFQ (called GMFQ in this article) for which the infinitesimal generator of $Z(t)$ is denoted by $\boldsymbol{Q}$ ($\tilde{\boldsymbol{Q}}$) for $X(t) > 0$ ($X(t) = 0$) and the common drift matrix is denoted by $\boldsymbol{R}$. The two infinitesimal generators and the drift matrix completely characterize the GMFQ, i.e., $\boldsymbol{X}(t) \sim\ GMFQ(\boldsymbol{Q}, \tilde{\boldsymbol{Q}}, \boldsymbol{R})$, where the size of these matrices, K, is the order of the GMFQ. Based on [50], the differential equation (3) also holds for the GMFQ. Next, we define the steady-state joint probability density function (pdf) vector $\boldsymbol{f}(x)$

$$f_i(x) = \lim_{t\to\infty} \frac{\mathrm{d}}{\mathrm{d}x} \Pr\{X(t) \le x,\ Z(t) = i\} \tag{5}$$

$$\boldsymbol{f}(x) = \begin{bmatrix} f_1(x) & f_2(x) & \cdots & f_K(x) \end{bmatrix}. \tag{6}$$

As shown in [30], it is also possible to write (3) in terms of $\boldsymbol{f}(x)$ as follows:

$$\frac{\mathrm{d}}{\mathrm{d}x}\boldsymbol{f}(x)\boldsymbol{R} = \boldsymbol{f}(x)\boldsymbol{Q} \tag{7}$$

which is obtained by differentiating both sides of (3) with respect to x. Furthermore, the steady-state probability mass accumulation (pma) vector at zero, defined as

$$\boldsymbol{c} = \begin{bmatrix} c_1 & c_2 & \cdots & c_K \end{bmatrix}$$

where $c_i = F_i(0)$, satisfies the following boundary condition:

$$\boldsymbol{f}(0)\boldsymbol{R} = \boldsymbol{c}\tilde{\boldsymbol{Q}} \tag{8}$$

which is shown in [30] by using the differential equations in cdf form that are originally developed in [50].

For the GMFQs of interest to the current manuscript, $r_i = 1$ when $i \le L = K-1$ and $r_i = -1$ for $i = K$. Moreover, for the GMFQs that arise in this article, when $X(t) > 0$, no transition can occur from state K to any other state, which makes the last row of the generator $\boldsymbol{Q}$ a row vector of zeros. However, a transition is possible from state K to any one of the other states when $X(t) = 0$. Hence, the characterizing matrices of $\boldsymbol{X}(t) \sim\ GMFQ(\boldsymbol{Q}, \tilde{\boldsymbol{Q}}, \boldsymbol{R})$ to be studied in this article will be in the following general form:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{h} \\ \boldsymbol{0} & 0 \end{bmatrix},\ \tilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\alpha} & -\boldsymbol{\alpha}\boldsymbol{1} \end{bmatrix},\ \boldsymbol{R} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -1 \end{bmatrix} \tag{9}$$

where the sizes of the north-west, north-east, and south-west partitions are $L \times L$, $L \times 1$, and $1 \times L$, respectively, and the notations $\boldsymbol{I}$, $\boldsymbol{1}$, and $\boldsymbol{0}$ are used to denote an identity matrix, column matrix of ones, and a matrix of zeros of appropriate sizes, respectively. We are interested in finding the steady-state joint pdf vector $\boldsymbol{f_L}(x)$ defined as

$$\boldsymbol{f_L}(x) = \begin{bmatrix} f_1(x) & f_2(x) & \cdots & f_{K-1}(x) \end{bmatrix} \tag{10}$$

that is the joint pdf vector comprising of the states with positive drift. The following theorem provides an expression for the steady-state joint pdf vector $\boldsymbol{f_L}(x)$.

Theorem 1: Consider the process

$$\boldsymbol{X}(t) \sim\ GMFQ\Big(\boldsymbol{Q}, \tilde{\boldsymbol{Q}}, \boldsymbol{R}\Big)$$

with its characterizing matrices in the form given in (9). Then, the steady-state joint pdf vector $\boldsymbol{f_L}(x)$ is given in the following matrix-exponential form (up to a constant) as follows:

$$\boldsymbol{f_L}(x) = \eta\boldsymbol{\alpha}\mathrm{e}^{\boldsymbol{W}x} \tag{11}$$

where η is a scalar constant.

Proof: Let us express the steady-state joint pdf vector of $\boldsymbol{X}(t)$ as $\boldsymbol{f}(x) = [\boldsymbol{f_L}(x) \quad f_K(x)]$. Using (7), we obtain the following expression:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x}\boldsymbol{f}(x) &= \begin{bmatrix} \boldsymbol{f_L}(x) & f_K(x) \end{bmatrix}\boldsymbol{Q}\boldsymbol{R}^{-1} \\ &= \begin{bmatrix} \boldsymbol{f_L}(x) & f_K(x) \end{bmatrix}\begin{bmatrix} \boldsymbol{W} & -\boldsymbol{h} \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{12}$$

which gives the differential equation of interest as follows:

$$\frac{\mathrm{d}}{\mathrm{d}x}\boldsymbol{f_L}(x) = \boldsymbol{f_L}(x)\boldsymbol{W}. \tag{13}$$

Next, we specify the steady-state pma vector at zero as $\boldsymbol{c} = [\,\mathbf{0} \quad \eta\,]$ by letting $c_i = 0$ for each state i with $r_i > 0$ using (4) and $c_K = \eta$ where η is a scalar constant. The boundary condition (8) is then used to obtain

$$\begin{aligned}\big[\boldsymbol{f_L}(0) \quad f_K(0)\big] &= \big[\mathbf{0} \quad \eta\big]\tilde{\boldsymbol{Q}}\boldsymbol{R}^{-1} \\ &= \big[\,\eta\boldsymbol{\alpha} \quad \eta\boldsymbol{\alpha}\mathbf{1}\,\big]. \end{aligned} \tag{14}$$

The solution to (13) can now be written as $\boldsymbol{f_L}(x) = \boldsymbol{f_L}(0)\mathrm{e}^{\boldsymbol{W}x}$ with $\boldsymbol{f_L}(0) = \eta\boldsymbol{\alpha}$ from (14), which completes the proof. ■

In the next section, we will obtain the characterizing matrices in the form (9) of the associated MFQs for each of the three policies and describe how the distribution of AoI can be obtained from the steady-state joint pdf vectors of the underlying MFQs.

Remark 1: In [45], the scalar constant η was also explicitly obtained for similar MFQs with a more elaborate algorithm. However, we have recently observed that obtaining the quantity $\boldsymbol{f_L}(x)$ up to a scalar constant is sufficient for finding the AoI distributions of interest.

IV. Analytical Models

We consider the information update system shown in Fig. 1 consisting of N sources with packet arrivals from source-n modeled by a Poisson process with intensity λ_n. The traffic intensity vector is defined as $(\lambda_1, \lambda_2, \ldots, \lambda_N)$. The service times of source-n packets are exponentially distributed with mean $1/\mu_n$. Subsequently, the rate vector is defined as $(\mu_1, \mu_2, \ldots, \mu_N)$. Moreover, the per-source load is defined as $\rho_n = \lambda_n/\mu_n$ and the overall system load is given by $\rho = \sum_{n=1}^{N} \rho_n$. The packet management policy is as follows: A newly arriving packet immediately receives service if the server is found idle. Otherwise, the packet gets queued in the 1-packet buffer allocated to that particular source. If the buffer is not empty, the existing packet is replaced only if the arriving packet belongs to same source. Upon a service completion, if there exists only one packet in the waiting room, this packet immediately starts to receive service. On the other hand, a specific policy is applied to select a source to be served if there exist multiple packets in the waiting room. In this setting, which we refer as SBPSQ, we first study two policies, namely, the FSFS and the ESFS, for which we construct a unifying MFQ model to obtain the exact AoI distributions for each source. Subsequently, this framework is employed to study the SBR policy.

A. First Source First Serve Policy

In the FSFS policy, the source of the first packet that has arrived at the system is the first source to be served among all the backlogged sources. In other words, the service order of sources with an existing packet in the queue is solely determined by their first packet arrival times and, thus, the service order does not change under replacement events. In our modeling approach, we focus on a source, say source-1, for which we obtain the exact distribution of AoI where the distribution for any source can be obtained similarly by renumbering the sources.

As the first step, we will obtain the probability distribution of the possible system states that an arriving source-1 packet finds upon its arrival to the system which will then be used while constructing the proposed MFQ model in the second step. For this purpose, we construct a finite state-space CTMC, denoted by $Y(t)$. We enumerate each state for $Y(t)$ as a tuple $q = (i, (P_m)) \in \mathcal{Q}_Y$ where $i \in \mathcal{I}_Y = \{0, 1, \ldots, N\}$ enumerates the source tag of packet that is currently being served except the case when $i = 0$ which is used for enumerating the idle server. For ease of notation, we let $P_m = s_1, s_2, \ldots, s_m$, $1 \le m \le N$, enumerate an m-permutation of the set $\mathcal{N} = \{1, 2, \ldots, N\}$ such that any $P_m \in \Gamma_Y$ can be generated by choosing m distinct source tags s_j, $1 \le j \le m$, from the set $\mathcal{N}$ and ordering them. When the server is busy and the queue contains m packets, we define the queue status (P_m) as follows:

$$(P_m) = \begin{cases} (0), & m = 0 \\ (s_1, s_2, \ldots, s_m), & 1 \le m \le N \end{cases} \tag{15}$$

where the term $(s_1, s_2, \ldots, s_m)$ enumerates the ordering of $m \ge 1$ sources in the queue with respect to their first packet arrival times in ascending order. When there are $m \ge 1$ packets in the queue and a packet belonging to source-s_j, $1 \le j \le m$, arrives to the system, the replacement event occurs but the queue status (P_m) does not get updated. According to the FSFS policy, the packet of the leftmost source will receive service first, i.e., s_1 denotes the source tag of packet which will receive service first among those in the queue. Similarly, s_2 is the source tag of packet which will receive service after the service completion of source-s_1 and so on. Since, the s_j terms $\forall j \in \{1, 2, \ldots, m\}$ in P_m are all distinct, we also denote the set of sources with an existing packet in the queue as $\{P_m\}$ without any ambiguity. Finally, when the server is idle, we enumerate the system state as $q = (0, (0))$ since there cannot be any waiting packet in the queue when the server is idle.

Suppose that the system state at time t is $Y(t) = q$ at which moment a service completion event occurs when there are $m > 0$ packets in the queue. According to the FSFS policy, the server selects the packet of source-s_1 for service after which the system transitions into the state $q' = (s_1, (P'_m))$ where the updated queue status (P'_m) with $m-1$ packets in the queue is given as

$$(P'_m) = \begin{cases} (0), & m = 1 \\ (s_2, s_3, \ldots, s_m), & 1 < m \le N \end{cases} \tag{16}$$

that is the source-s_1 is removed from the ordered list of sources with an existing packet in the queue.

Let $\nu_{q,q'} > 0$, $q, q' \in \mathcal{Q}_Y$, denote the rate of transition from state $q = (i, (P_m))$ to state $q' \ne q$ where we list all such transitions for the FSFS policy in Table I for which the rows 1–3 (4 and 5) correspond to the transition rates for arrival (departure) events. For any other state pair q, $q' \in \mathcal{Q}_Y$, the transition rate $\nu_{q,q'}$ is zero.

Let us denote the probability that the system is in state q as time goes to infinity, i.e., $\pi_q = \lim_{t\to\infty} P(Y(t) = q)$. Following the ergodicity of Markov chain $Y(t)$, the steady-state distribution converges to a unique vector consisting of elements π_q, $q \in \mathcal{Q}_Y$, which satisfies the following set of linear

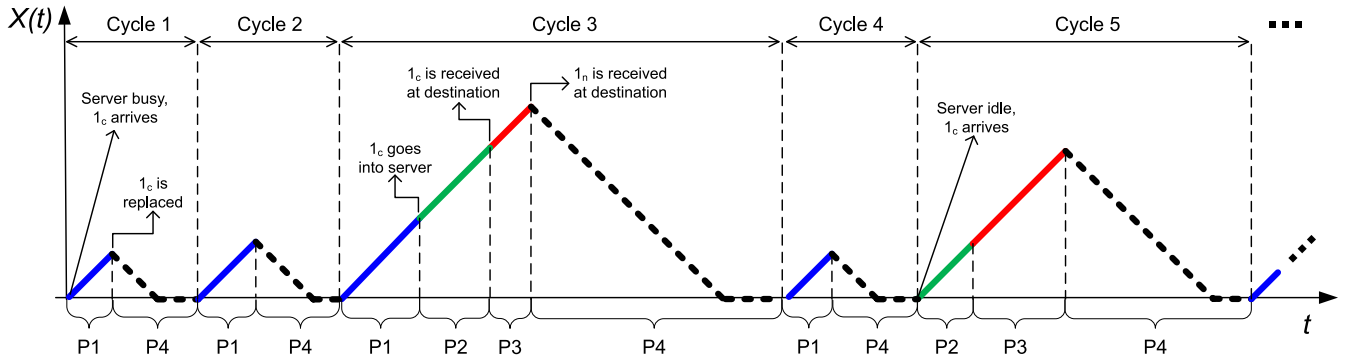

Fig. 2. Sample path of the fluid level process $X(t)$ with infinitely many independent cycles.

TABLE I
TRANSITION RATES $\nu_{q,q'}$ OF $Y(t)$ FOR THE FSFS POLICY

q	q'	$\nu_{q,q'}$	Conditions
$(0,(0))$	$(i,(0))$	λ_i	$i \in \mathcal{N}$
$(i,(0))$	$(i,(j))$	λ_j	$i,j \in \mathcal{N}$
$(i,(P_m))$	$(i,(P_m,j))$	λ_j	$i,j \in \mathcal{N}$, $j \notin \{P_m\}, P_m \in \Gamma_Y$
$(i,(0))$	$(0,(0))$	μ_i	$i \in \mathcal{N}$
$(i,(P_m))$	$(s_1,(P'_m))$	μ_i	$i \in \mathcal{N}, P_m \in \Gamma_Y$

equations:

$$\pi_q \sum_{q' \in \mathcal{Q}_Y} \nu_{q,q'} = \sum_{q' \in \mathcal{Q}_Y} \pi_{q'} \nu_{q',q} \ \forall q \in \mathcal{Q}_Y \tag{17}$$

$$\sum_{q \in \mathcal{Q}_Y} \pi_q = 1. \tag{18}$$

Since the packet arrivals are Poisson, the probability that an arriving packet finds the system in state $q \in \mathcal{Q}_Y$ is π_q as a direct consequence of the Poisson Arrivals See Time Averages (PASTA) property [51].

In the second step, we construct the proposed MFQ process $\boldsymbol{X}(t) = (X(t), Z(t))$ which describes a fluid level trajectory with infinitely many independent cycles as shown in Fig. 2 where each cycle begins with an arriving source-1 packet to the system and ends with either the reception of the next source-1 packet at the destination (cycles 3 and 5 in Fig. 2) or the possible packet replacement by another source-1 arrival (cycles 1, 2, and 4 in Fig. 2). First, we construct the state-space $\boldsymbol{S}$ of the subprocess $Z(t)$ by dividing a cycle into four phases and defining the set of states for each phase. We also define three additional tags for packets belonging to source-1 to differentiate between them in different states and phases.

1) The packet 1_c, i.e., current source-1 packet, refers to the source-1 packet that initiates each cycle with its arrival at the system.
2) When a packet 1_c arrives at the system, the server can be busy already serving another source-1 packet which is denoted by 1_p, i.e., previous source-1 packet.
3) The packet 1_n (next source-1 packet) refers to the received source-1 packet subsequent to the packet 1_c at the destination.

Each cycle consists of four phases, namely, phases 1–4 as shown in Fig. 2. If the server is busy when the packet 1_c arrives, the cycle starts from phase 1 (solid blue curve) during which the fluid level $X(t)$ increases at a unit rate and terminates with either the beginning of its service time at which instance the system transitions into phase 2 (which occurs during cycle 3 in Fig. 2) or the replacement of it by another source-1 arrival (which occurs during cycles 1, 2, and 4 in Fig. 2). In the latter case, the queue wait time of the packet 1_c needs a reset which is accomplished by transitioning directly into a final phase, defined as phase 4 (dashed black curve), that is used for resetting the fluid level by bringing it down to zero with a unit rate. For phase 1, we enumerate each state as $q = (i,(P_m)) \in \mathcal{Q}_1$ where $i \in \mathcal{I}_1 = \{1_p, 2, \ldots, N\}$ denotes the source tag of the packet under service. For any $i \in \mathcal{I}_1$ value, the element (P_m), $1 \le m \le N$, enumerates the ordering of packets in the queue similar to the previously given definition for $Y(t)$ with the exception that the packet 1_c always exists in the queue during phase 1. Thus, any $P_m \in \Gamma_1$ can be generated by ordering 1_c and another $(m-1)$ distinct source tags selected from the set $\{2, 3, \ldots, N\}$. With all these definitions, we enumerate the queue status (P_m) containing m packets for phase 1 as follows:

$$(P_m) = (s_1, s_2, \ldots, s_m),\ 1_c \in \{P_m\},\ 1 \le m \le N \tag{19}$$

which is valid for any $i \in \mathcal{I}_1$. If an arriving packet 1_c finds the system idle, a direct transition to phase 2 (solid green curve) occurs (which is shown as cycle 5 in Fig. 2). During phase 2, the fluid level continues to increase at a unit rate until the reception instant of packet 1_c at destination at which instance the system transitions into phase 3. Note that once the packet 1_c starts receiving service, it can no longer be replaced by another packet arrival. Thus, the only possible transition out of phase 2 is into 3. We enumerate each state for phase 2 as $q = (1_c,(P_m)) \in \mathcal{Q}_2$ where the queue status (P_m), $1 \le m \le N$, is similar to the previously given definitions with the exception that the packet 1_n may exist or not in the queue during phase 2. In the latter case, any $P_m \in \Gamma_2$, $1 \le m \le N-1$, can be generated by ordering m distinct source tags selected from the set $\{2, 3, \ldots, N\}$. In the former case, we impose a restriction given as $s_m = 1_n$, i.e., the last packet to be served in the queue is always 1_n. The reason is that any packet behind 1_n in the queue is irrelevant because, as shown in Fig. 2, the system always transitions into the final phase, i.e., phase 4, at the reception instant of packet 1_n at destination regardless of the queue status. Therefore, for this case, any $P_m \in \Gamma_2$,

$1 \leq m \leq N$, can be generated by selecting 1_n and another $(m-1)$ distinct source tags from set $\{2, 3, \ldots, N\}$, and ordering them while satisfying the condition $s_m = 1_n$, i.e., the last source to be served is the source-1 when there are $m \geq 1$ packets in the queue. Finally, when there is no packet in the queue, we define the queue status as $(P_m) = (0)$. With all these definitions, we enumerate the queue status (P_m) containing m packets for phase 2 as follows:

$$(P_m) = \begin{cases} (0), & m = 0 \\ (s_1, s_2, \ldots, s_m), & 1 \leq m \leq N-1, 1_n \notin \{P_m\} \\ (s_1, s_2, \ldots, s_m), & 1 \leq m \leq N, s_m = 1_n. \end{cases} \quad (20)$$

Once phase 2 is over, phase 3 (solid red curve) starts and continues until the reception of the packet 1_n, at destination. Each state for phase 3 is enumerated as $q = (i, (P_m)) \in \mathcal{Q}_3$ where $i \in \mathcal{I}_3 = \{0, 1_n, 2, \ldots, N\}$ denotes the source tag of the packet under service except the case when $i = 0$ which is used for enumerating the idle server. Similar to the arguments for phase 2, any packet behind 1_n in the system is irrelevant. Therefore, when the packet under service is 1_n, the system state is enumerated as $q = (1_n, (0))$. If the server is busy but the packet under service is not 1_n, i.e., $i \neq 1_n$, the queue status (P_m) can be defined as given in (20) similar to phase 2. In particular, if the queue contains the packet 1_n, any $P_m \in \Gamma_3$, $1 \leq m \leq N$, can be generated by ordering 1_n and another $(m-1)$ distinct elements selected from set $\{2, 3, \ldots, N\}$, satisfying the condition $s_m = 1_n$. If the queue does not contain 1_n, any $P_m \in \Gamma_3$, $1 \leq m \leq N-1$, can be generated by ordering m distinct elements selected from the set $\{2, 3, \ldots, N\}$. Finally, we enumerate the state corresponding to the idle server as $q = (0, (0))$ which may occur only in phase 3 when the packet 1_c was delivered to the destination but the next source-1 packet, i.e., packet 1_n, has not arrived at the system yet.

Once phase 3 is over, the system transitions into the final stage, i.e., phase 4, where the fluid level is brought down to 0 with a drift of minus one after which the fluid level stays at 0 for exponentially distributed time with unit rate. Thus, phase 4 consists of a single state which we enumerate as $q = (-1, (-1)) \in \mathcal{Q}_4$. After the fluid level is brought down to 0 in phase 4, the only possible transition out of phase 4 is to phase 1 or 2 both of which initiates a new cycle that is independent from all previous cycles. With all these definitions, the state-space $\boldsymbol{S}$ of $Z(t)$ can now be defined as $\boldsymbol{S} = \bigcup_{p=1}^{4} \mathcal{Q}_p$ consisting of all states defined for phases 1–4.

State transition diagram of the subprocess $Z(t)$ can be represented as a directed graph as shown in Fig. 3, where each edge represents a set of transitions between or within phases. We define the corresponding transition rates such that if the system remains in the same phase after a transition, we will refer such a transition as an intraphase transition for which the rate is denoted by $\alpha_{q,q'}, q, q' \in \mathcal{Q}_p, p = 1, 2, 3, 4$, whereas if it transitions to another phase, it will be referred as an interphase transition in which case the rate is denoted by $\beta_{q,q'}, q \in \mathcal{Q}_p, q' \notin \mathcal{Q}_p, p = 1, 2, 3, 4$. In line with the FSFS policy, all intraphase and interphase transitions are listed in Tables II and III, respectively, where the set $\mathcal{J}_p$, $p = 1, 2, 3$ is defined as the set of source tags to which any packet in the queue may belong in phase p that is $\mathcal{J}_1 = \{1_c, 2, \ldots, N\}$ and $\mathcal{J}_2 = \mathcal{J}_3 = \{1_n, 2, \ldots, N\}$. Unless explicitly stated otherwise, in the corresponding row, given transitions are defined for the condition $X(t) > 0$, which constitute the entries of the matrix $\boldsymbol{Q}$, whereas the transitions defined for $X(t) = 0$ constitute the entries of the matrix $\tilde{\boldsymbol{Q}}$. For intraphase transitions, the rows 1 and 2, 3 and 4, and 5–9 refer to the transitions for phases 1–3, respectively. Note that there is no intraphase transition for phase 4 since its state-space consists of a single state. For interphase transitions, the rows 1, 2, 3 and 4, 5, 6, and 7–9 refer to the transitions from phase 1 to 2, phase 1 to 4, phase 2 to 3, phase 3 to 4, phase 4 to 2, and phase 4 to 1, respectively. Since the transitions from phase 4 to 1 or 2

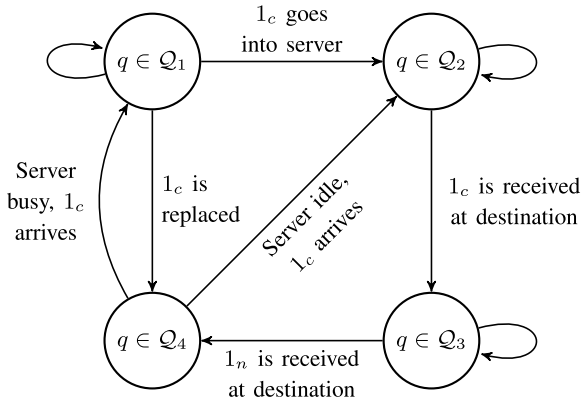

Fig. 3. State transition diagram of the subprocess $Z(t)$.

TABLE II
INTRAPHASE TRANSITION RATES $\alpha_{q,q'}$ FOR THE FSFS POLICY

q	q'	$\alpha_{q,q'}$	Conditions
$(i,(P_m))$	$(i,(P_m,j))$	λ_j	$i \in \mathcal{I}_1, j \in \mathcal{J}_1,$ $j \notin \{P_m\}, P_m \in \Gamma_1$
$(i,(P_m))$	$(s_1,(P'_m))$	μ_i	$i \in \mathcal{I}_1, s_1 \neq 1_c,$ $P_m \in \Gamma_1$
$(1_c,(0))$	$(1_c,(j))$	λ_j	$j \in \mathcal{J}_2$
$(1_c,(P_m))$	$(1_c,(P_m,j))$	λ_j	$j \in \mathcal{J}_2, j \notin \{P_m\},$ $s_m \neq 1_n, P_m \in \Gamma_2$
$(0,(0))$	$(i,(0))$	λ_i	$i \in \mathcal{I}_3 - \{0\}$
$(i,(0))$	$(i,(j))$	λ_j	$i \in \mathcal{I}_3 - \{0,1_n\}, j \in \mathcal{J}_3$
$(i,(P_m))$	$(i,(P_m,j))$	λ_j	$i \in \mathcal{I}_3 - \{0,1_n\}, j \in \mathcal{J}_3,$ $s_m \neq 1_n, j \notin \{P_m\}, P_m \in \Gamma_3$
$(i,(0))$	$(0,(0))$	μ_i	$i \in \mathcal{I}_3 - \{0,1_n\}$
$(i,(P_m))$	$(s_1,(P'_m))$	μ_i	$i \in \mathcal{I}_3 - \{0,1_n\}, P_m \in \Gamma_3$

TABLE III
INTERPHASE TRANSITION RATES $\beta_{q,q'}$ FOR THE FSFS POLICY

q	q'	$\beta_{q,q'}$	Conditions
$(i,(P_m))$	$(1_c,(P'_m))$	μ_i	$i \in \mathcal{I}_1, s_1 = 1_c,$ $P_m \in \Gamma_1$
$(i,(P_m))$	$(-1,(-1))$	λ_1	$i \in \mathcal{I}_1, P_m \in \Gamma_1$
$(1_c,(0))$	$(0,(0))$	μ_1	
$(1_c,(P_m))$	$(s_1,(P'_m))$	μ_1	$P_m \in \Gamma_2$
$(1_n,(0))$	$(-1,(-1))$	μ_1	
$(-1,(-1))$	$(1_c,(0))$	$\pi_{(0,(0))}$	$X(t) = 0$
$(-1,(-1))$	$(i,(P_m))$	$\pi_{(i,(P_m))}$	$X(t) = 0,$ $i \in \mathcal{I}_1, s_m \neq 1_c,$ $1_c \in \{P_m\}, P_m \in \Gamma_1$
$(-1,(-1))$	$(i,(1_c))$	$\pi_{(i,(0))} +$ $\pi_{(i,(1))}$	$X(t) = 0, i \in \mathcal{I}_1$
$(-1,(-1))$	$(i,(P_m,1_c))$	$\pi_{(i,(P_m))} +$ $\pi_{(i,(P_m,1))}$	$X(t) = 0, i \in \mathcal{I}_1,$ $1_c \notin \{P_m\}, P_m \in \Gamma_1$

initiate a new cycle, their rates are proportional to the steady-state distribution of the system status that a source-1 packet finds upon its arrival to the system. By solving the steady-state distribution of the process $Y(t)$ as described in the first step, the rates of these transitions are expressed as given in the last four rows of Table III. Expressing the transition rates in terms of the steady-state probabilities of $Y(t)$ stems from the fact that the fluid level stays at 0 in phase 4 for an exponentially distributed time with unit rate, i.e., the sum of transitions out of phase 4 when $X(t) = 0$ should be exactly one which equals to the sum of steady-state probabilities $\pi_q, q \in \mathcal{Q}_Y$.

Next, we define the drift value r_q for each state $q \in \boldsymbol{S}$ which constitutes the diagonal entries of the drift matrix $\boldsymbol{R}$. Since the fluid level increases at a unit rate in each state for phases 1–3, we have $r_q = 1, \forall q \in \bigcup_{p=1}^{3} \mathcal{Q}_p$ whereas the fluid level is brought down to 0 with a drift of minus one in phase 4. Thus, we have $r_q = -1, q \in \mathcal{Q}_4$ which completes the construction of the proposed MFQ model $\boldsymbol{X}(t) \sim GMFQ(\boldsymbol{Q}, \tilde{\boldsymbol{Q}}, \boldsymbol{R})$ with the characterizing matrices being in the form (9) by ensuring that the single state with negative drift, i.e., $(-1, (-1)) \in \mathcal{Q}_4$, is numbered as the last state that is the state K in the formulation given in Section III.

By sample path arguments, we observe that one sample cycle of the AoI process coincides with the part of the sample cycle of $X(t)$ associated with phase 3 only as indicated by the red parts of the curve in Fig. 2. Therefore, the pdf of the AoI for source-1, denoted by $f_{\Delta_1}(x)$, can be written in terms of the joint pdfs of the GMFQ $\boldsymbol{X}(t)$ as follows:

$$f_{\Delta_1}(x) = \frac{\sum_{q \in \mathcal{Q}_3} f_q(x)}{\int_0^\infty \sum_{q \in \mathcal{Q}_3} f_q(x')\, dx'}, x \geq 0. \tag{21}$$

For censoring out all states with positive drift other than the ones in $\mathcal{Q}_3$, we define a column vector $\boldsymbol{\beta}$ of size L containing only zeros except for the states $q \in \mathcal{Q}_3$ for which it is set to 1. Using (11) along with this definition, we can finally obtain

$$f_{\Delta_1}(x) = \epsilon\, \boldsymbol{\alpha} \mathrm{e}^{\boldsymbol{W}x} \boldsymbol{\beta},\ x \geq 0 \tag{22}$$

where $\epsilon^{-1} = -\boldsymbol{\alpha W^{-1} \beta}$. The kth noncentral moments of Δ_1 can also be easily written as follows:

$$\mathbb{E}\Big[(\Delta_1)^k\Big] = (-1)^{k+1} k!\, \epsilon\, \boldsymbol{\alpha W^{-(k+1)} \beta}. \tag{23}$$

Similar steps are then followed for obtaining the pdf of the AoI for source-n, denoted by $f_{\Delta_n}(x), n = 2, 3, \ldots, N$, by renumbering the sources. Finally, we define the performance metrics of interest, namely, the average AoI and the average age violation probability, denoted by $\mathbb{E}[\Delta]$ and $\Theta(\gamma)$, respectively, as follows:

$$\mathbb{E}[\Delta] = \sum_{n=1}^{N} \frac{\mathbb{E}[\Delta_n]}{N}, \Theta(\gamma) = \sum_{n=1}^{N} \frac{Q_{\Delta_n}(\gamma)}{N} \tag{24}$$

where $\Delta = (1/N) \sum_{n=1}^{N} \Delta_n$, $\mathbb{E}[\Delta_n]$ is the average AoI for source-n, and $Q_{\Delta_n}(\gamma)$ is the age violation probability for source-n which is calculated as $Q_{\Delta_n}(\gamma) = Pr\{\Delta_n > \gamma\}$ where γ is a given age violation threshold.

The framework that we introduced in this section is unifying in the sense that it can be generalized to any SBPSQ policy by only redefining the following terms.

1) The state-space $\mathcal{Q}_Y$ and the corresponding transition rates $\nu_{q,q'}$ of the process $Y(t)$.
2) The state-space $\mathcal{Q}_p$ for $p = 1, 2, 3, 4$, and the corresponding intra- (inter-) phase transition rates $\alpha_{q,q'}$ ($\beta_{q,q'}$).

Since the sample path of the fluid level process shown in Fig. 2 and the state transition diagram shown in Fig. 3 are valid for any such policy. In fact, from sample path arguments, this generalization also holds for the SBR policy. Therefore, using this unifying framework, we provide the analytical models for both ESFS and SBR policies by only redefining the above-mentioned state spaces and transition rates.

B. Earliest Served First Serve Policy

Each state of $Y(t)$ is enumerated as a tuple $q = ((H), \{C_m\}) \in \mathcal{Q}_Y$ for the ESFS policy. Let $H = h_1, h_2, \ldots, h_N$ enumerate an N-permutation of set $\mathcal{N}$ such that any $H \in \mathcal{H}_Y$ can be generated by choosing N distinct source tags from set $\mathcal{N}$, i.e., all source tags, and ordering them. Accordingly, $(H) = (h_1, h_2, \ldots, h_N)$ is defined as the service status where the sources are listed in ascending order with respect to their last selection instants for service. In other words, the tag $h_1 (h_N)$ indicates the source that has not received service for the longest (shortest) duration. For any state except the idle server, the tag h_N indicates the source whose packet is currently being served. Therefore, when a packet belonging to source-i starts receiving service, the tag h_N has to be updated as i and the other terms have to be shifted accordingly. For this purpose, we define an operation $\Upsilon((H), i)$ that updates the service status (H) as follows when a packet belonging to source-i starts receiving service:

$$\Upsilon((H), i) = \big(H_i'\big) = \big(h_1, \ldots, h_{f-1}, h_{f+1}, \ldots, h_N, h_f\big) \tag{25}$$

where $h_f = i$, i.e., the tag h_f belongs to source-i. Furthermore, we let $C_m = s_1, s_2, \ldots, s_m$, $1 \leq m \leq N$, enumerate an m-combination of the set $\mathcal{N} = \{1, 2, \ldots, N\}$ such that any $C_m \in \Gamma_Y$ can be generated by choosing m distinct source tags s_j, $1 \leq j \leq m$, from the set $\mathcal{N}$. Accordingly, the queue status $\{C_m\} = \{s_1, s_2, \ldots, s_m\}$ for $m \geq 1$ is defined as the set of m sources with an existing packet in the queue where the ordering of s_j terms is irrelevant in contrast with the FSFS policy. In the ESFS policy, the server selects the packet belonging to the source that has not received service for the longest duration among those with an existing packet in the queue. Suppose that the system state at time t is $Y(t) = q$ at which moment a service completion event occurs when there are $m > 0$ packets in the queue. In line with the ESFS policy, the server selects the packet of source-i^* for service where the tag i^* is defined as

$$i^* = h_{f^*},\ f^* = \min_{\forall f \in \mathcal{N}} f, h_f \in \{C_m\} \tag{26}$$

after which the system transitions into the state $q' = ((H'_{i^*}), \{C'_m\})$ where the updated queue status $\{C'_m\}$ with $m-1$ packets in the queue is given as

$$\{C'_m\} = \begin{cases} \{0\}, & m = 1 \\ \{C_m\} - \{i^*\}, & 1 < m \leq N \end{cases} \tag{27}$$

TABLE IV
TRANSITION RATES $\nu_{q,q'}$ OF $Y(t)$ FOR THE ESFS POLICY

q	q'	$\nu_{q,q'}$	Conditions
$((H),\{-1\})$	$((H'_i),\{0\})$	λ_i	$H \in \mathcal{H}_Y, i \in \mathcal{N}$
$((H),\{0\})$	$((H),\{j\})$	λ_j	$H \in \mathcal{H}_Y, j \in \mathcal{N}$
$((H),\{C_m\})$	$((H),\{C_m,j\})$	λ_j	$H \in \mathcal{H}_Y, j \in \mathcal{N}$, $j \notin \{C_m\}, C_m \in \Gamma_Y$
$((H),\{0\})$	$((H),\{-1\})$	μ_{h_N}	$H \in \mathcal{H}_Y$
$((H),\{C_m\})$	$((H'_{i^*}),\{C'_m\})$	μ_{h_N}	$H \in \mathcal{H}_Y, C_m \in \Gamma_Y$

that is the source-i^* is removed from the list of sources with an existing packet in the queue. Next, we define the system states with empty queue as follows.

1) When the server is busy but the queue is empty, we define the system state as $q = ((H),\{0\})$ where the packet in service belongs to source-h_N.
2) When the server is idle, we define the system state as $q = ((H),\{-1\})$ since the service status has to be always preserved in the ESFS policy even if the server is idle. In this case, the source-h_N is the source whose packet has been served most recently but is not currently in service.

This concludes the state definitions for the process $Y(t)$ after which we define the transition rates $\nu_{q,q'}$ of $Y(t)$ in Table IV where the rates correspond to the arrival (departure) events are given in rows 1–3 (4 and 5).

Next, we define the states $q = ((H),\{C_m\}) \in \mathcal{Q}_p$ for each phase.

1) For phase 1, the term H is defined as an N-permutation of set $\mathcal{I}_1 = \{1_p, 2, \ldots, N\}$. For any $H \in \mathcal{H}_1$, the packet 1_c always exists in the queue from the definition of phase 1. Thus, any $C_m \in \Gamma_1$, $1 \leq m \leq N$, can be generated by choosing the tag 1_c and $(m-1)$ distinct tags from set $\{2, \ldots, N\}$.
2) In phase 2, the server may only serve the packet 1_c and the queue may contain the packet 1_n or not from the definition of phase 2. Therefore, the term H is defined as an N-permutation of set $\mathcal{I}_2 = \{1_c, 2, \ldots, N\}$ satisfying the condition $h_N = 1_c$ which ensures that the packet under service belongs to the source-1. For any $H \in \mathcal{H}_2$, the term $C_m \in \Gamma_2$ for $1 \leq m \leq N$ is defined as an m-combination of set $\{1_n, 2, \ldots, N\}$ whereas we use $\{C_m\} = \{0\}$ when the queue is empty.
3) For phase 3, we define the term $H \in \mathcal{H}_3$ as an N-permutation of the set $\mathcal{I}_3 = \{1_n, 2, \ldots, N\}$. When the server is idle, we define the system state as $q = ((H),\{-1\})$ similar to the previously given definition for $Y(t)$. When the server is busy, the states are defined as follows: when the tag of packet under service is 1_n, i.e., $h_N = 1_n$, the queue status is defined as $\{C_m\} = \{0\}$ since the packets behind 1_n are irrelevant in our model as discussed in the FSFS policy. When $h_N \neq 1_n$, the term $C_m \in \Gamma_3$ for $1 \leq m \leq N$ is defined as an m-combination of set $\{1_n, 2, \ldots, N\}$ whereas we use $\{C_m\} = \{0\}$ when the queue is empty.
4) For phase 4, we have a single state which we define as $q = (-1,(-1)) \in \mathcal{Q}_4$ similar to the FSFS policy.

Finally, we list all the intraphase and interphase transitions for the ESFS policy in Tables V and VI, respectively. For intraphase transitions, the rows 1 and 2, 3 and 4, and 5-9 refer to the transitions for phases 1–3, respectively. For interphase transitions, the rows 1, 2, 3 and 4, 5, 6, and 7 and 8 refer to the transitions from phase 1 to 2, phase 1 to 4, phase 2 to 3, phase 3 to 4, phase 4 to 2, and phase 4 to 1, respectively. This concludes the analytical model for the ESFS policy.

TABLE V
INTRAPHASE TRANSITION RATES $\alpha_{q,q'}$ FOR THE ESFS POLICY

q	q'	$\alpha_{q,q'}$	Conditions
$((H),\{C_m\})$	$((H),\{C_m,j\})$	λ_j	$H \in \mathcal{H}_1, j \in \mathcal{J}_1$, $j \notin \{C_m\}, C_m \in \Gamma_1$
$((H),\{C_m\})$	$((H'_{i^*}),\{C'_m\})$	μ_{h_N}	$H \in \mathcal{H}_1, i^* \neq 1_c$, $C_m \in \Gamma_1$
$((H),\{0\})$	$((H),\{j\})$	λ_j	$H \in \mathcal{H}_2, j \in \mathcal{J}_2$
$((H),\{C_m\})$	$((H),\{C_m,j\})$	λ_j	$H \in \mathcal{H}_2, j \in \mathcal{J}_2$, $j \notin \{C_m\}, C_m \in \Gamma_2$
$((H),\{-1\})$	$((H'_i),\{0\})$	λ_i	$H \in \mathcal{H}_3, i \in \mathcal{I}_3$
$((H),\{0\})$	$((H),\{j\})$	λ_j	$H \in \mathcal{H}_3, h_N \neq 1_n$, $j \in \mathcal{J}_3$
$((H),\{C_m\})$	$((H),\{C_m,j\})$	λ_j	$H \in \mathcal{H}_3, h_N \neq 1_n$, $j \in \mathcal{J}_3, j \notin \{C_m\}$, $C_m \in \Gamma_3$
$((H),\{0\})$	$((H),\{-1\})$	μ_{h_N}	$H \in \mathcal{H}_3, h_N \neq 1_n$
$((H),\{C_m\})$	$((H'_{i^*}),\{C'_m\})$	μ_{h_N}	$H \in \mathcal{H}_3, h_N \neq 1_n$, $C_m \in \Gamma_3$

TABLE VI
INTERPHASE TRANSITION RATES $\beta_{q,q'}$ FOR THE ESFS POLICY

q	q'	$\beta_{q,q'}$	Conditions
$((H),\{C_m\})$	$((H'_{i^*}),\{C'_m\})$	μ_{h_N}	$H \in \mathcal{H}_1, i^* = 1_c$, $C_m \in \Gamma_1$
$((H),\{C_m\})$	$(-1,(-1))$	λ_1	$H \in \mathcal{H}_1, C_m \in \Gamma_1$
$((H),\{0\})$	$((H),\{-1\})$	μ_1	$H \in \mathcal{H}_2$
$((H),\{C_m\})$	$((H'_{i^*}),\{C'_m\})$	μ_1	$H \in \mathcal{H}_2, C_m \in \Gamma_2$
$((H),\{0\})$	$(-1,(-1))$	μ_1	$H \in \mathcal{H}_3, h_N = 1_n$
$(-1,(-1))$	$((H'_i),\{0\})$	$\sum \pi_{((H),\{-1\})}$	$X(t) = 0, H'_i \in \mathcal{H}_2$
$(-1,(-1))$	$((H),\{1_c\})$	$\pi_{((H),\{0\})} + \pi_{((H),\{1\})}$	$X(t) = 0, H \in \mathcal{H}_1$
$(-1,(-1))$	$((H),\{C_m,1_c\})$	$\pi_{((H),\{C_m\})} + \pi_{((H),\{C_m,1\})}$	$X(t) = 0, H \in \mathcal{H}_1$, $1_c \notin \{C_m\}, C_m \in \Gamma_1$

C. Single Buffer With Replacement Policy

For the SBR policy, each state for $Y(t)$ is enumerated as a tuple $q = (i,(j)) \in \mathcal{Q}_Y$ where $i \in \mathcal{I}_Y = \{0, 1, \ldots, N\}$ enumerates the source tag of packet that is currently being served, i.e., the server status, except the case when $i = 0$ which is used for enumerating the idle server. For any $i > 0$, element (j) enumerates the buffer status such that $j \in \mathcal{B}_Y = \{0, 1, \ldots, N\}$ indicates the source tag of packet waiting in the buffer except the case when $j = 0$ which is used for enumerating the empty buffer status. When the server is idle, i.e., $i = 0$, the only possible buffer status is $(j) = (0)$ since the server can be idle only when the buffer is empty. Thus, we enumerate the state corresponding to the idle server as $q = (0,(0))$ which completes the state definitions of $Y(t)$ for the SBR policy. Next, we provide the transitions rates $\nu_{q,q'}$ of $Y(t)$ in Table VII where rows 1–3 and 4 and 5 correspond to the arrival and departure events, respectively. Since the buffer is shared by all sources in the SBR policy, an arrival from any source replaces the existing packet in the buffer (in contrast with the FSFS and ESFS policies) as defined in row 3.

Next, we define the states $q = (i,(j)) \in \mathcal{Q}_p$ for each phase.

TABLE VII
TRANSITION RATES $\nu_{q,q'}$ OF $Y(t)$ FOR THE SBR POLICY

q	q'	$\nu_{q,q'}$	Conditions
$(0,(0))$	$(i,(0))$	λ_i	$i \in \mathcal{N}$
$(i,(0))$	$(i,(j))$	λ_j	$i,j \in \mathcal{N}$
$(i,(j))$	$(i,(k))$	λ_k	$i,j,k \in \mathcal{N}, k \neq j$
$(i,(0))$	$(0,(0))$	μ_i	$i \in \mathcal{N}$
$(i,(j))$	$(j,(0))$	μ_i	$i,j \in \mathcal{N}$

TABLE VIII
INTRAPHASE TRANSITION RATES $\alpha_{q,q'}$ FOR THE SBR POLICY

q	q'	$\alpha_{q,q'}$	Conditions
$(1_c,(0))$	$(1_c,(j))$	λ_j	$j \in \mathcal{J}_2$
$(1_c,(j))$	$(1_c,(k))$	λ_k	$j,k \in \mathcal{J}_2, k \neq j$
$(0,(0))$	$(i,(0))$	λ_i	$i \in \mathcal{I}_3 - \{0\}$
$(i,(0))$	$(i,(j))$	λ_j	$i \in \mathcal{I}_3 - \{0,1_n\}, j \in \mathcal{J}_3$
$(i,(j))$	$(i,(k))$	λ_j	$i \in \mathcal{I}_3 - \{0,1_n\}$, $j,k \in \mathcal{J}_3, k \neq j$
$(i,(0))$	$(0,(0))$	μ_i	$i \in \mathcal{I}_3 - \{0,1_n\}$
$(i,(j))$	$(j,(0))$	μ_i	$i \in \mathcal{I}_3 - \{0,1_n\}, j \in \mathcal{J}_3$

TABLE IX
INTERPHASE TRANSITION RATES $\beta_{q,q'}$ FOR THE SBR POLICY

q	q'	$\beta_{q,q'}$	Conditions
$(i,(1_c))$	$(1_c,(0))$	μ_i	$i \in \mathcal{I}_1$
$(i,(1_c))$	$(-1,(-1))$	$\sum_{j \in \mathcal{J}_1} \lambda_j$	$i \in \mathcal{I}_1$
$(1_c,(0))$	$(0,(0))$	μ_1	
$(1_c,(j))$	$(j,(0))$	μ_1	$j \in \mathcal{J}_2$
$(1_n,(0))$	$(-1,(-1))$	μ_1	
$(-1,(-1))$	$(1_c,(0))$	$\pi_{(0,(0))}$	$X(t)=0$
$(-1,(-1))$	$(i,(1_c))$	$\sum_{j \in \mathcal{B}_Y} \pi_{(i,(j))}$	$X(t)=0, i \in \mathcal{I}_1$

1) For phase 1, the server status is defined as $i \in \mathcal{I}_1 = \{1_p, 2, \ldots, N\}$ similar to the FSFS policy. For any i value, the buffer status can be only $(j) = (1_c)$ since the buffer always contains the packet 1_c in phase 1.
2) In phase 2, the server may only serve the packet 1_c, i.e., $i = 1_c$, and the buffer may contain the packet 1_n or not from the definition of phase 2. Thus, we define the buffer status as (j), $j \in \{0, 1_n, 2, \ldots, N\}$, in phase 2.
3) For phase 3, the server status is defined as $i \in \mathcal{I}_3 = \{0, 1_n, 2, \ldots, N\}$ similar to the FSFS policy. When $i = 0$, the only possible buffer status is $(j) = (0)$, i.e., $q = (0, (0))$, for which we have the idle server. When $i = 1_n$, the only possible buffer status is also $(j) = (0)$ since any to-be-served packet after the packet 1_n is irrelevant in our model as discussed for the FSFS policy. When $i = 2, 3, \ldots, N$, the buffer may be empty or hold a packet from any source for which we define the buffer status as (j), $j \in \{0, 1_n, \ldots, N\}$, similar to phase 2.
4) For phase 4, we have a single state which we define as $q = (-1, (-1)) \in \mathcal{Q}_4$ similar to the FSFS policy.

Finally, we list all the intraphase and interphase transitions for the SBR policy in Tables VIII and IX, respectively. For intraphase transitions, rows 1 and 2 and 3–7 refer to the transitions for phases 2 and 3, respectively. In contrast with the FSFS and ESFS policies, there is no intraphase transition defined for phase 1 since the first packet arrival from any source replaces the packet 1_c in the buffer which results in a direct transition to phase 4 for the SBR policy. For interphase transitions, the rows 1, 2, 3 and 4, 5, 6, and 7 refer to the transitions from phase 1 to 2, phase 1 to 4, phase 2 to 3, phase 3 to 4, phase 4 to 2, and phase 4 to 1, respectively. In Table IX, the second row corresponds to the event where the packet 1_c waiting in the buffer is replaced by the first packet arrival from any source in phase 1. The last row corresponds to the case where the packet 1_c finds the server busy upon its arrival to the system in which case it replaces the packet in the buffer irrespective of its source as opposed to FSFS and ESFS policies. Thus, out of phase 4, the system transitions into the state $q = (i, (1_c))$ with rate $\sum_{j \in \mathcal{B}_Y} \pi_{(i,(j))}$ that is the sum of steady-state probabilities of all states in $Y(t)$ where the source-i packet is being served. This concludes the analytical model for the SBR policy.

TABLE X
VALUES OF L_{SBR}, L_{FSFS}, AND L_{ESFS} WHEN THE NUMBER OF SOURCES N RANGES BETWEEN 2 AND 5

	$N=2$	$N=3$	$N=4$	$N=5$
L_{SBR}	10	17	26	37
L_{FSFS}	16	65	326	1957
L_{ESFS}	15	80	606	5904

D. Computational Considerations for the Analytical Models

In this section, a comparison of the computational cost of the MFQ-based analytical model for each policy is provided. Note that this is different than the complexity of implementing the actual policies on the server. For this comparison, we report the size of the square matrix $\boldsymbol{W}_{L\times L}$ whose inversion is required for obtaining the average AoI for each source in (23) or the matrix-exponential function of $\boldsymbol{W}$ is needed to obtain the age violation probabilities in (22). Clearly, we have $L = |\mathcal{Q}_1| + |\mathcal{Q}_2| + |\mathcal{Q}_3|$ which corresponds to the number of states in $\mathcal{S}$ with positive drift where the notation $|\cdot|$ is used to denote the cardinality of the argument set. The values of L for each policy, denoted by L_{SBR}, L_{FSFS}, and L_{ESFS}, respectively, are listed in Table X when the number of sources ranges between 2 and 5. We observe that as the number of sources increases, the size of the matrix $\boldsymbol{W}$ grows significantly faster with the ESFS and FSFS policies than the SBR policy, which subsequently limits the number of sources that can be analyzed with the MFQ technique when the computational resources are limited. In fact, we have been able to produce numerical results using the MATLAB implementation of the MFQ-based analytical method presented in Section IV when the number of sources, N, is less than or equal to 5. When the number of sources increases further, more computational capabilities might be needed. However, we note that the proposed technique is computationally stable.

V. NUMERICAL EXAMPLES

In this section, the proposed analytical models are first verified with simulations for each policy. Subsequently, the analytical models are used to compare the three studied policies under several scenarios.

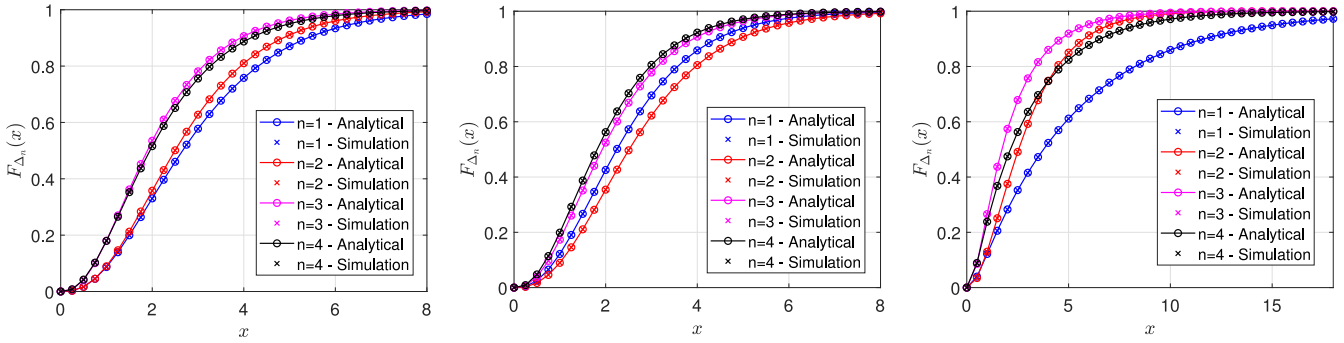

Fig. 4. Cdf $F_{\Delta_n}(x)$ of each source for (a) FSFS, (b) ESFS, and (c) SBR policies when the number of sources $N = 4$, and the arrival and the service rate vectors are (1, 2, 3, 2) and (3, 1, 2, 4) packets/sec, respectively.

A. Model Validation

We consider a scenario where $N = 4$ for which the arrival and service rate vectors are assumed to be (1, 2, 3, 2) and (3, 1, 2, 4), respectively. The cdf of the AoI for each source-n, denoted by $F_{\Delta_n}(x)$, is shown in Fig. 4 for each policy using both the analytical model and simulations. We observe that the analytical results are perfectly in line with the simulation results. Therefore, for the rest of this article, we will only use the proposed analytical models for evaluating the policies.

B. Comparative Assessment of the Scheduling Policies

In this section, the performance of the studied policies is evaluated with respect to the average AoI and average age violation probability metrics under several scenarios where the sources may have identical or different traffic intensities, referred to as balanced and unbalanced load scenarios, respectively. We assume the service rate is common and equal to one for all sources, i.e., $\mu_n = 1, n = 1, 2, \ldots, N$, for all the numerical examples.

1) Balanced Load: In this section, we consider a scenario where the load is balanced among all sources such that the arrival rate for each source is given as $\lambda_n = \rho/N, n = 1, 2, \ldots, N$. We sweep the number of sources from 3 to 5 for which we obtain sum AoI, defined as $\sum_{n=1}^{N} \mathbb{E}[\Delta_n] = N\mathbb{E}[\Delta]$, for each policy with respect to the system load ρ as shown in Fig. 5. We observe that the ESFS policy consistently outperforms the other two policies in moderate loads with FSFS being slightly worse for all the three cases. Moreover, the performance gaps between the policies grow as the number of sources increases. This shows the effectiveness of selecting the source that is not served for the longest duration as opposed to considering first packet arrival times of FSFS. Finally, as the system load increases toward infinity, the sum AoI for the ESFS and FSFS policies become identical as expected since both policies behave the same, i.e., round-robin service, when there is always a packet (in the waiting room) for each source upon a service completion.

In the next example, we evaluate the studied policies with respect to the average age violation probability metric under two system loads. Specifically, the low and moderate load scenarios are considered where the parameter ρ for each case is assumed to be 0.5 and 4, respectively. For both scenarios, the average age violation probability with respect to the age threshold parameter γ is depicted in Fig. 6 for all three policies. We observe that when the system load is low, FSFS and ESFS policies perform quite close to each other with a slightly better performance than SBR policy whereas the performance

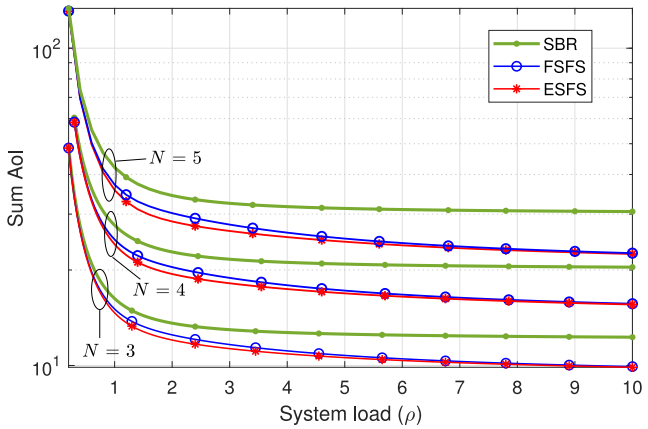

Fig. 5. Sum AoI, $N\mathbb{E}[\Delta]$, obtained with the SBR, FSFS, and ESFS policies as a function of the system load ρ when the sources have identical traffic intensities.

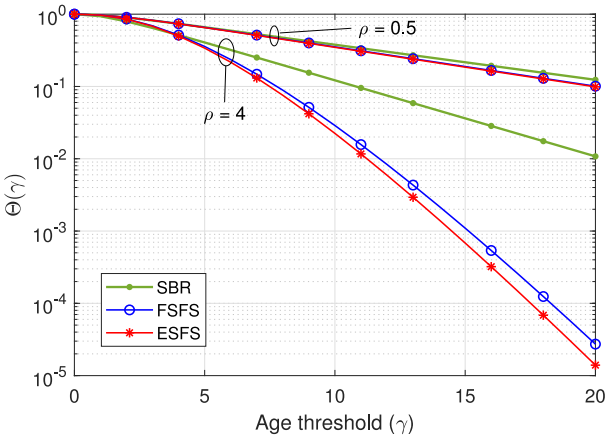

Fig. 6. Average age violation probability, $\Theta(\gamma)$, for the SBR, FSFS, and ESFS policies as a function of the age threshold parameter γ when there are $N = 4$ sources with identical traffic intensities.

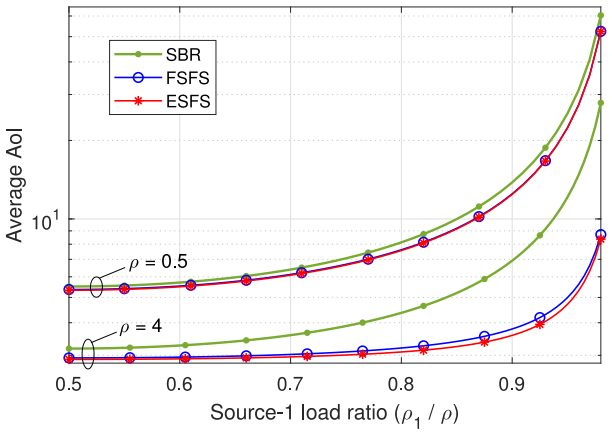

Fig. 7. Average AoI, $\mathbb{E}[\Delta]$, obtained with the SBR, FSFS, and ESFS policies as a function of the source-1 load ratio (ρ_1/ρ) when the number of sources $N = 2$.

gap grows in the moderate load regime. Moreover, the ESFS policy outperforms the other two policies in both scenarios. Finally, as the system load increases, we observe that the average age violation probability can be reduced significantly faster with respect to the violation parameter γ with SBPSQ-based FSFS and ESFS policies than it can be achieved with the SBR policy.

2) Unbalanced Load: In this section, we study a scenario where the sources may have different traffic intensities for a given system load ρ. We assume $N = 2$ for which the average AoI with respect to the source-1 load ratio, defined as ρ_1/ρ, is given in Fig. 7 for the low and moderate load scenarios (where we sweep ρ_1 from $\rho/2$ to ρ due to symmetry). In the low load scenario, we observe that all the three policies perform close to each other with the SBR policy being slightly worse. In the moderate system load, the average AoI worsens with remarkably slower rate with respect to the source-1 load ratio for the SBPSQ-based FSFS and ESFS policies than the SBR policy as the load asymmetry between the sources increases. Moreover, we observe that the ESFS policy consistently outperforms the FSFS and SBR policies for any ρ_1 value under both system loads, which shows the effectiveness of the ESFS policy also under scenarios with different traffic mixes.

VI. Conclusion

In this article, we studied a single-hop multisource information status update system. Under the assumption of Poisson packet arrivals and exponentially distributed heterogeneous service times for each source, we proposed and validated an analytical model to obtain the exact steady-state distributions of the per-source AoI processes in matrix-exponential form when FSFS, ESFS, and SBR queueing/scheduling policies are employed. In the numerical examples, we evaluated the studied policies in terms of average AoI and average age violation probability for several scenarios under a common service time distribution, varying system loads, and varying traffic mixes. We showed that the proposed ESFS policy consistently outperforms the other two studied policies FSFS and SBR, where the degree of out-performance with respect to FSFS being relatively modest. Furthermore, when SBPSQ-based FSFS and ESFS policies are employed at the server, performance improvement with these two policies over SBR increases with higher loads and also when the load asymmetry among the sources increases. We therefore concluded that the age-agnostic ESFS policy, which is very simple to implement, is a viable scheduling method for SBPSQ-based systems and we believe that it can be used in practical IoT networks for status update applications when sources are equally important. Future work will consist of practical scheduling policies for nonsymmetric networks when the minimization of weighted average AoI is sought. Another line of future research is the extension of the proposed analytical models to generally distributed service times, and multiserver and multihop scenarios.

References

[1] M. A. Abd-Elmagid, N. Pappas, and H. S. Dhillon, "On the role of age of information in the Internet of Things," *IEEE Commun. Mag.*, vol. 57, no. 12, pp. 72–77, Dec. 2019.

[2] B. Zhou and W. Saad, "Minimizing age of information in the Internet of Things with non-uniform status packet sizes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[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. Kosta, N. Pappas, and V. Angelakis, "Age of information: A new concept, metric, and tool," *Found. Trends Netw.*, vol. 12, no. 3, pp. 162–259, 2017.

[5] 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, May 2021.

[6] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, "Decentralized status update for age-of-information optimization in wireless multiaccess channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 2276–2280.

[7] 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, Dec. 2018.

[8] R. D. Yates and S. K. Kaul, "Status updates over unreliable multiaccess channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 331–335.

[9] T. Shreedhar, S. K. Kaul, and R. D. Yates, "Coexistence of age sensitive traffic and high throughput flows: Does prioritization help?" 2022, *arXiv:2203.00647*.

[10] Y. Gu, H. Chen, Y. Zhou, Y. Li, and B. Vucetic, "Timely status update in Internet of Things monitoring systems: An age-energy tradeoff," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5324–5335, Jun. 2019.

[11] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the Internet of Things," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7468–7482, Nov. 2019.

[12] J. Li, Y. Zhou, and H. Chen, "Age of information for multicast transmission with fixed and random deadlines in IoT systems," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8178–8191, Sep. 2020.

[13] B. Zhou and W. Saad, "Performance analysis of age of information in ultra-dense Internet of Things (IoT) systems with noisy channels," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3493–3507, May 2022.

[14] R. Han, J. Wang, L. Bai, J. Liu, and J. Choi, "Age of information and performance analysis for UAV-aided IoT systems," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14447–14457, Oct. 2021.

[15] C. M. W. Basnayaka, D. N. K. Jayakody, and Z. Chang, "Age of information based URLLC-enabled UAV wireless communications system," *IEEE Internet Things J.*, early access, Nov. 22, 2021, doi: 10.1109/JIOT.2021.3123431.

[16] F. Chiariotti, O. Vikhrova, B. Soret, and P. Popovski, "Peak age of information distribution for edge computing with wireless links," *IEEE Trans. Commun.*, vol. 69, no. 5, pp. 3176–3191, May 2021.

[17] L. Hu *et al.*, "Optimal status update in IoT systems: An age of information violation probability perspective," in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC-Fall)*, 2020, pp. 1–5.

[18] L. Hu, Z. Chen, Y. Dong, Y. Jia, L. Liang, and M. Wang, "Status update in IoT networks: Age-of-information violation probability and optimal update rate," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11329–11344, Jul. 2021.
[19] J. P. Champati, H. Al-Zubaidy, and J. Gross, "Statistical guarantee optimization for AoI in single-hop and two-hop FCFS systems with periodic arrivals," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 365–381, Jan. 2021.
[20] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33–37, 1997. [Online]. Available: https://doi.org/10.1002/ett.4460080106
[21] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. IEEE 51st Veh. Technol. Conf.*, vol. 3. Tokyo, Japan, 2000, pp. 1854–1858.
[22] L. B. Jiang and S. C. Liew, "Proportional fairness in wireless lans and ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, vol. 3, 2005, pp. 1551–1556.
[23] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Optimal sampling and scheduling for timely status updates in multi-source networks," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4019–4034, Jun. 2021.
[24] C. Joo and A. Eryilmaz, "Wireless scheduling for information freshness and synchrony: Drift-based design and heavy-traffic analysis," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2556–2568, Dec. 2018.
[25] I. Kadota and E. Modiano, "Minimizing the age of information in wireless networks with stochastic arrivals," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 1173–1185, Mar. 2021.
[26] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1807–1827, Mar. 2019.
[27] N. Pappas, J. Gunnarsson, L. Kratz, M. Kountouris, and V. Angelakis, "Age of information of multiple sources with queue management," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2015, pp. 5935–5940.
[28] M. Moltafet, M. Leinonen, and M. Codreanu, "Average age of information for a multi-source M/M/1 queueing model with packet management," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2020, pp. 1765–1769.
[29] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data handling system with multiple sources," *Bell Syst. Techn. J.*, vol. 61, no. 8, pp. 1871–1894, Oct. 1982.
[30] E. Kankaya and N. Akar, "Solving multi-regime feedback fluid queues," *Stochastic Models*, vol. 24, pp. 425–450, Aug. 2008.
[31] E. O. Gamgam and N. Akar, 2022, "MATLAB Code for Age of Information Modeling in Multi-Source Status Update Systems," IEEE DataPort. [Online]. Available: https://dx.doi.org/10.21227/13x2-8815
[32] R. D. Yates and S. Kaul, "Real-time status updating: Multiple sources," in *Proc. IEEE Int. Symp. Inf. Theory*, 2012, pp. 2666–2670.
[33] S. K. Kaul and R. D. Yates, "Timely updates by multiple sources: The M/M/1 queue revisited," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, 2020, pp. 1–6.
[34] 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, Aug. 2020.
[35] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2015, pp. 1681–1685.
[36] E. Najm and E. Telatar, "Status updates in a multi-stream M/G/1/1 preemptive queue," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops*, 2018, pp. 124–129.
[37] E. Najm, R. Yates, and E. Soljanin, "Status updates through M/G/1/1 queues with HARQ," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 131–135.
[38] M. Moltafet, M. Leinonen, and M. Codreanu, "Average AoI in multi-source systems with source-aware packet management," *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 1121–1133, Feb. 2021.
[39] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Timely status updating over erasure channels using an energy harvesting sensor: Single and multiple sources," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 6–19, Mar. 2022.
[40] E. Najm, R. Nasser, and E. Telatar, "Content based status updates," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 2266–2270.
[41] S. K. Kaul and R. D. Yates, "Age of information: Updates with priority," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 2644–2648.
[42] A. Maatouk, M. Assaad, and A. Ephremides, "Age of information with prioritized streams: When to buffer preempted packets?" in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 325–329.
[43] M. A. Abd-Elmagid and H. S. Dhillon, "Closed-form characterization of the MGF of AoI in energy harvesting status update systems," *IEEE Trans. Inf. Theory*, vol. 68, no. 6, pp. 3896–3919, Jun. 2022.
[44] M. Moltafet, M. Leinonen, and M. Codreanu, "Moment generating function of the AoI in multi-source systems with computation-intensive status updates," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2021, pp. 1–6.
[45] O. Doğan and N. Akar, "The multi-source probabilistically preemptive M/PH/1/1 queue with packet errors," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7297–7308, Nov. 2021.
[46] M. Moltafet, M. Leinonen, and M. Codreanu, "Moment generating function of the AoI in a two-source system with packet management," *IEEE Wireless Commun. Lett.*, vol. 10, no. 4, pp. 882–886, Apr. 2021.
[47] N. Akar and E. Karasan, "Scheduling algorithms for age of information differentiation with random arrivals," 2021, *arXiv 2110.10992*.
[48] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *Bell Syst. Techn. J.*, vol. 61, no. 8, pp. 1871–1894, Oct. 1982.
[49] V. G. Kulkarni, *Fluid Models for Single Buffer Systems*. Boca Raton, FL, USA: CRC Press, 1998, pp. 321–338.
[50] M. Mandjes, D. Mitra, and W. Scheinhardt, "Models of network access using feedback fluid queues," *Queueing Syst.*, vol. 44, no. 4, pp. 365–398, 2003.
[51] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, 2nd ed. New York, NY, USA: Wiley, 1985.

Ege Orkun Gamgam received the B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree.

He is a Senior Design Engineer with ASELSAN Inc., Ankara. His current research interests include the design and analysis of wireless communication systems and performance modeling of wireless networks.

Nail Akar (Member, IEEE) received the B.S. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 1987, and the M.S. and Ph.D. degrees in electrical and electronics engineering from Bilkent University, Ankara, in 1989 and 1994, respectively.

From 1994 to 1996, he was a Visiting Scholar and a Visiting Assistant Professor of Computer Science Telecommunications program with the University of Missouri–Kansas City, Kansas, MO, USA. He joined the Technology Planning and Integration Group, Long Distance Division, Sprint, Overland Park, KS, USA, in 1996, where he held a Senior Member of Technical Staff position from 1999 to 2000. Since 2000, he has been with Bilkent University as a Professor with the Electrical and Electronics Engineering Department. He visited the School of Computing, University of Missouri–Kansas City, as a Fulbright Scholar in 2010 for a period of six months. His research interests are on performance modeling of computer and communication systems and networks, wireless networks, Internet of Things, queueing theory, and optimization.