

# Source-Aware Packet Management for Computation-Intensive Status Updating: MGF of the AoI

Mohammad Moltafet and Markus Leinonen  
Centre for Wireless Communications – Radio Technologies  
University of Oulu, Finland  
e-mail: {mohammad.moltafet, markus.leinonen}@oulu.fi

Marian Codreanu  
Department of Science and Technology  
Linköping University, Sweden  
e-mail: marian.codreanu@liu.se

**Abstract**—We consider a multi-source status update system consisting of two independent sources and one server. The packets of different sources are generated according to a Poisson process and served according to an exponentially distributed service time. We assume that the received status update packets need further processing before being used (hence, computation-intensive). This is mathematically modeled by introducing an additional server at the sink node. The sink server serves the packets according to an exponentially distributed service time. We derive the moment generating function (MGF) of the AoI of each source under a source-aware packet management policy. According to this policy, when a server (either the transmitter or sink server) is busy at the arrival of a packet, the packet currently under service is preempted only if the arriving packet is from the same source (hence, source-aware). Numerical results are provided to assess the results.

**Index Terms**— Age of information (AoI), stochastic hybrid systems (SHS), computation-intensive status updates.

## I. INTRODUCTION

Delivery of fresh status information of various physical processes is a key enabler for time-critical applications in future networks and cyber-physical systems. The age of information (AoI) was proposed as a destination-centric metric to measure the information freshness in status update systems [1]–[3]. A status update packet contains the measured value of a monitored process and a time stamp representing the time when the sample was generated. Due to wireless channel access, channel errors, and fading etc., communicating a status update packet through the network experiences a random delay. If at a time instant  $t$ , the most recent status update available at the sink contains the time stamp  $U(t)$ , AoI is defined as the random process  $\Delta(t) = t - U(t)$ . Thus, the AoI measures for each sensor the time elapsed since the last available status update packet was generated at the sensor.

The work in [4] introduced a powerful technique, called *stochastic hybrid systems* (SHS), to calculate the average AoI. In [5], the authors extended the SHS analysis to calculate the moment generating function (MGF) of the AoI. The SHS technique has been used to analyze the AoI in various queueing models [6]–[14].

In this paper, we consider a multi-source system with computation-intensive status updates in which the embed-

ded information in each packet is not available until being processed by the sink. An autonomous driving system in which images are status updates could be an example of the considered system model. In this system, further processing is needed at the sink to expose the embedded status information in the images. We introduce a source-aware packet management policy and derive the MGF of the AoI of each source. According to this policy, when a server (either the transmitter or sink server) is busy at the arrival of a packet, the packet currently under service is preempted only if the arriving packet is from the same source.

The most related works to this paper are [14], [15]. In [14], we considered the same system model and derived the MGF of the AoI under two packet management policies namely, i) a (source-agnostic) preemptive policy and ii) a blocking policy. As shown in [10], source-aware packet management policies can improve fairness between sources in multi-source status update systems. Surprisingly, the proposed source-aware packet management policy for computation-intensive systems provides worse fairness than that of the source-agnostic policies studied in [14]. In [15], the authors derived the average AoI for a single-source system where the computation-intensive status updates are generated according to the zero-wait policy. According to the zero-wait policy, a new packet is generated immediately after the previous one is available at the sink.

## II. SYSTEM MODEL

We consider a status update system consisting of two independent sources<sup>1</sup>, one transmitter server, one sink server, and one sink, as depicted in Fig. 1. Each source observes a random process at random time instants. The sink is interested in timely information about the status of these random processes. Status updates are transmitted as packets, containing the measured value of the monitored process and a time stamp representing the time when the sample was generated. We assume that the packets of sources 1 and 2 are

<sup>1</sup>Since we have multiple sources of Poisson arrivals, to evaluate the AoI of one source, we can consider two sources without loss of generality.

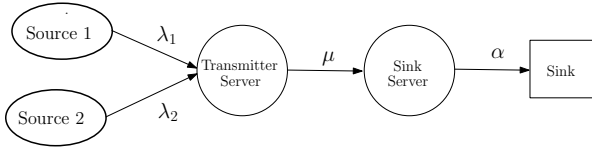


Fig. 1: The considered computation-intensive status update system.

generated according to the Poisson process with rates  $\lambda_1$  and  $\lambda_2$ , respectively, and the packets are served by the transmitter server according to the exponentially distributed service time with mean  $1/\mu$ . We assume that the embedded information in each packet is revealed to the sink only after being processed by the sink server which serves the packets according to the exponentially distributed service time with mean  $1/\alpha$ .

Let  $\rho_1 = \lambda_1/\mu$  and  $\rho_2 = \lambda_2/\mu$  be the load of source 1 and 2, respectively. The packet generation in the system follows the Poisson process with rate  $\lambda = \lambda_1 + \lambda_2$ , and the overall load at the transmitter server is  $\rho = \rho_1 + \rho_2 = \lambda/\mu$ .

#### A. Source-Aware Packet Management Policy

The system size is two packets, one at the transmitter server and one at the sink server (i.e., there are no waiting queues).

On both transmitter and sink sides, when a packet arrives, the possible packet of the same source (hence, *source-aware*) at a server is replaced by the new packet, otherwise; the arriving packet is blocked and cleared.

### III. THE SHS TECHNIQUE TO CALCULATE MGF

In the following, we briefly present how to use the SHS technique for our MGF analysis in Section IV. We refer the readers to [4], [5] for more details.

The SHS technique models a queueing system through the states  $(q(t), \mathbf{x}(t))$ , where  $q(t) \in \mathcal{Q} = \{0, 1, \dots, m\}$  is a continuous-time finite-state Markov chain that describes the occupancy of the system and  $\mathbf{x}(t) = [x_0(t) \cdots x_n(t)] \in \mathbb{R}^{1 \times (n+1)}$  is a continuous process that describes the evolution of age-related processes in the system. Following the approach in [4], [10], we label the source of interest as source 1 and employ the continuous process  $\mathbf{x}(t)$  to track the age of source 1 updates at the sink.

The Markov chain  $q(t)$  can be presented as a graph  $(\mathcal{Q}, \mathcal{L})$  where each discrete state  $q(t) \in \mathcal{Q}$  is a node of the chain and a (directed) link  $l \in \mathcal{L}$  from node  $q_l$  to node  $q'_l$  indicates a transition from state  $q_l \in \mathcal{Q}$  to state  $q'_l \in \mathcal{Q}$ .

A transition occurs when a packet arrives or departs in the system. Since the packets are generated according to the Poisson process and the service time is exponentially distributed, transition  $l \in \mathcal{L}$  from state  $q_l$  to state  $q'_l$  occurs with the exponential rate  $\lambda^{(l)} \delta_{q_l, q(t)}$ , where the Kronecker delta function  $\delta_{q_l, q(t)}$  ensures that the transition  $l$  occurs only when the discrete state  $q(t)$  is equal to  $q_l$ . When a transition  $l$  occurs, the discrete state  $q_l$  changes to state  $q'_l$ , and the continuous state  $\mathbf{x}$  is reset to  $\mathbf{x}'$  according to a binary reset map matrix  $\mathbf{A}_l \in \mathbb{B}^{(n+1) \times (n+1)}$  as  $\mathbf{x}' = \mathbf{x} \mathbf{A}_l$ . In addition, as long as state

$q(t)$  is unchanged we have  $\dot{\mathbf{x}}(t) \triangleq \frac{\partial \mathbf{x}(t)}{\partial t} = \mathbf{1}$ , where  $\mathbf{1}$  is the row vector  $[1 \cdots 1] \in \mathbb{R}^{1 \times (n+1)}$ .

Note that unlike in a typical continuous-time Markov chain, a transition from a state to itself (i.e., a self-transition) is possible in  $q(t) \in \mathcal{Q}$ . In the case of a self-transition, a reset of the continuous state  $\mathbf{x}$  takes place, but the discrete state remains the same. In addition, for a given pair of states  $\bar{q}, \hat{q} \in \mathcal{Q}$ , there may be multiple transitions  $l$  and  $l'$  so that the discrete state changes from  $\bar{q}$  to  $\hat{q}$  but the transition reset maps  $\mathbf{A}_l$  and  $\mathbf{A}_{l'}$  are different (for more details, see [4, Section III]).

To calculate the MGF of the AoI using the SHS technique, the state probabilities of the Markov chain, the correlation vector between the discrete state  $q(t)$  and the continuous state  $\mathbf{x}(t)$ , and the correlation vector between the discrete state  $q(t)$  and the exponential function  $e^{s\mathbf{x}(t)}$ ,  $s \in \mathbb{R}$ , need to be defined. Let  $\pi_q(t)$  denote the probability of being in state  $q$  of the Markov chain. Let  $\mathbf{v}_q(t) = [v_{q0}(t) \cdots v_{qn}(t)] \in \mathbb{R}^{1 \times (n+1)}$  denote the correlation vector between the discrete state  $q(t)$  and the continuous state  $\mathbf{x}(t)$ . Let  $\mathbf{v}_q^s(t) = [v_{q0}^s(t) \cdots v_{qn}^s(t)] \in \mathbb{R}^{1 \times (n+1)}$  denote the correlation vector between the state  $q(t)$  and the exponential function  $e^{s\mathbf{x}(t)}$ . Accordingly, we have

$$\pi_q(t) = \Pr(q(t) = q) = \mathbb{E}[\delta_{q,q(t)}], \quad \forall q \in \mathcal{Q}, \quad (1)$$

$$\mathbf{v}_q(t) = [v_{q0}(t) \cdots v_{qn}(t)] = \mathbb{E}[\mathbf{x}(t) \delta_{q,q(t)}], \quad \forall q \in \mathcal{Q}, \quad (2)$$

$$\mathbf{v}_q^s(t) = [v_{q0}^s(t) \cdots v_{qn}^s(t)] = \mathbb{E}[e^{s\mathbf{x}(t)} \delta_{q,q(t)}], \quad \forall q \in \mathcal{Q}. \quad (3)$$

Let  $\mathcal{L}'_q$  denote the set of incoming transitions and  $\mathcal{L}_q$  denote the set of outgoing transitions for state  $q$ , defined as

$$\mathcal{L}'_q = \{l \in \mathcal{L} : q'_l = q\}, \quad \mathcal{L}_q = \{l \in \mathcal{L} : q_l = q\}, \quad \forall q \in \mathcal{Q}.$$

Following the ergodicity assumption of the Markov chain  $q(t)$  in the AoI analysis [4], [5], the state probability vector  $\boldsymbol{\pi}(t) = [\pi_0(t) \cdots \pi_m(t)]$  converges uniquely to the stationary vector  $\bar{\boldsymbol{\pi}} = [\bar{\pi}_0 \cdots \bar{\pi}_m]$  satisfying

$$\bar{\pi}_q \sum_{l \in \mathcal{L}_q} \lambda^{(l)} = \sum_{l \in \mathcal{L}'_q} \lambda^{(l)} \bar{\pi}_{q_l}, \quad \forall q \in \mathcal{Q}, \quad (4)$$

$$\sum_{q \in \mathcal{Q}} \bar{\pi}_q = 1. \quad (5)$$

Further, it has been shown in [5, Theorem 1] that under the ergodicity assumption of the Markov chain  $q(t)$ , if we can find a non-negative limit  $\bar{\mathbf{v}}_q = [\bar{v}_{q0} \cdots \bar{v}_{qn}]$ ,  $\forall q \in \mathcal{Q}$ , for the correlation vector  $\mathbf{v}_q(t)$  satisfying

$$\bar{\mathbf{v}}_q \sum_{l \in \mathcal{L}_q} \lambda^{(l)} = \bar{\pi}_q \mathbf{1} + \sum_{l \in \mathcal{L}'_q} \lambda^{(l)} \bar{\mathbf{v}}_{q_l} \mathbf{A}_l, \quad \forall q \in \mathcal{Q}, \quad (6)$$

there exists  $s_0 > 0$  such that for all  $s < s_0$ ,  $\mathbf{v}_q^s(t)$ ,  $\forall q \in \mathcal{Q}$ , converges to  $\bar{\mathbf{v}}_q^s$  that satisfies

$$\bar{\mathbf{v}}_q^s \sum_{l \in \mathcal{L}_q} \lambda^{(l)} = s \bar{\mathbf{v}}_q^s + \sum_{l \in \mathcal{L}'_q} \lambda^{(l)} [\bar{\mathbf{v}}_{q_l}^s \mathbf{A}_l + \bar{\pi}_{q_l} \mathbf{1} \hat{\mathbf{A}}_l], \quad \forall q \in \mathcal{Q}, \quad (7)$$

where  $\hat{\mathbf{A}}_l \in \mathbb{B}^{(n+1) \times (n+1)}$  is a binary matrix whose  $k, j$ th

element,  $\hat{\mathbf{A}}_l(k, j)$ , is given as

$$\hat{\mathbf{A}}_l(k, j) = \begin{cases} 1, & k=j, \text{ and } j\text{th column of } \mathbf{A}_l \text{ is a zero vector,} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Finally, the MGF of the state  $\mathbf{x}(t)$ , which is calculated by  $\mathbb{E}[e^{s\mathbf{x}(t)}]$ , converges to the stationary vector [5, Theorem 1]

$$\mathbb{E}[e^{s\mathbf{x}}] = \sum_{q \in \mathcal{Q}} \bar{\mathbf{v}}_q^s. \quad (9)$$

As (9) implies, if we set the first element of continuous state  $\mathbf{x}(t)$  to represent the AoI of source 1 at the sink, the MGF of the AoI of source 1 at the sink converges to

$$M_{\Delta_1}(s) = \sum_{q \in \mathcal{Q}} \bar{v}_{q0}^s. \quad (10)$$

Thus, the main challenge in calculating the MGF of the AoI of source 1 using the SHS technique reduces to deriving the first elements of correlation vectors  $\bar{\mathbf{v}}_q^s, \forall q \in \mathcal{Q}$ .

#### IV. AOI ANALYSIS USING THE SHS TECHNIQUE

In this section, we use the SHS technique to calculate the MGF in (10) of each source under the considered packet management policy described in Section II-A.

The discrete states are  $0 = 00$ ,  $1 = 10$ ,  $2 = 20$ ,  $3 = 01$ ,  $4 = 11$ ,  $5 = 21$ ,  $6 = 02$ ,  $7 = 12$ ,  $8 = 22$ . State  $q = a_1 a_2$  indicates that a packet of source  $a_1$  is in the transmitter server and a packet of source  $a_2$  is in the sink server. Note that  $a_1 = 0$  (resp.  $a_2 = 0$ ) indicates that the transmitter server (resp. the sink server) is idle.

The continuous process is  $\mathbf{x}(t) = [x_0(t) \ x_1(t) \ x_2(t)]$ , where  $x_0(t)$  is the current AoI of source 1 at time instant  $t$ ,  $\Delta_1(t)$ ,  $x_1(t)$  encodes what  $\Delta_1(t)$  would become if the packet that is in the sink server is delivered to the sink at time instant  $t$ , and  $x_2(t)$  encodes what  $\Delta_1(t)$  would become if the packet that is in the transmitter server is delivered to the sink at time instant  $t$ .

Recall that to calculate the MGF of the AoI of source 1 in (10), we need to find  $\bar{v}_{q0}^s, \forall q \in \mathcal{Q}$ , which are the solution of the system of linear equations in (7) with variables  $\bar{\mathbf{v}}_q^s, \forall q \in \mathcal{Q}$ . To form the system of linear equations (6) and (7), for each state  $q \in \mathcal{Q}$ , we need to determine  $\bar{\pi}_q$  and  $\mathbf{A}_l$  and  $\hat{\mathbf{A}}_l$  for each incoming transition  $l \in \mathcal{L}'_q$ , carried out next.

1) *Calculation of  $\bar{\pi}_q \forall q \in \mathcal{Q}$* : The Markov chain for the discrete state  $q(t)$  is shown in Fig. 2. Using (4), (5), and the transition rates among the different states illustrated in Fig. 2, it can be shown that the stationary probability vector  $\bar{\pi}$  satisfies the following equations:

$$\begin{aligned} \lambda \bar{\pi}_0 &= \alpha \bar{\pi}_3 + \alpha \bar{\pi}_6, \\ (\lambda_1 + \mu) \bar{\pi}_1 &= \lambda_1 \bar{\pi}_0 + \lambda_1 \bar{\pi}_1 + \alpha \bar{\pi}_4 + \alpha \bar{\pi}_7, \\ (\lambda_1 + \mu) \bar{\pi}_2 &= \lambda_2 \bar{\pi}_0 + \lambda_1 \bar{\pi}_2 + \alpha \bar{\pi}_5 + \alpha \bar{\pi}_8, \\ (\lambda + \alpha) \bar{\pi}_3 &= \mu \bar{\pi}_1 + \mu \bar{\pi}_4 + \mu \bar{\pi}_5, \\ (\lambda_1 + \mu + \alpha) \bar{\pi}_4 &= \lambda_1 \bar{\pi}_3 + \lambda_1 \bar{\pi}_4, \\ (\lambda_1 + \mu + \alpha) \bar{\pi}_5 &= \lambda_2 \bar{\pi}_3 + \lambda_1 \bar{\pi}_5, \end{aligned} \quad (11)$$

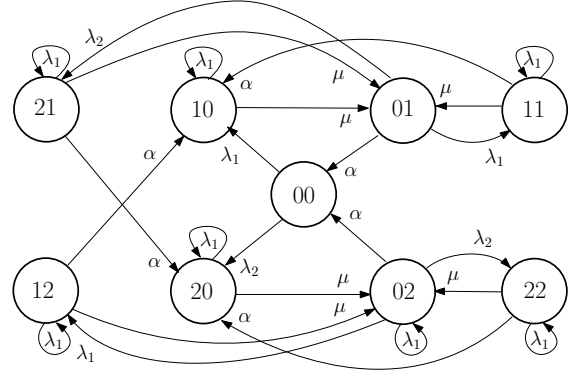


Fig. 2: The SHS Markov chain for the considered policy.

$$\begin{aligned} (\lambda + \alpha) \bar{\pi}_6 &= \mu \bar{\pi}_2 + \mu \bar{\pi}_7 + \mu \bar{\pi}_8, \\ (\lambda_1 + \mu + \alpha) \bar{\pi}_7 &= \lambda_1 \bar{\pi}_6 + \lambda_1 \bar{\pi}_7, \\ (\lambda_1 + \mu + \alpha) \bar{\pi}_8 &= \lambda_2 \bar{\pi}_6 + \lambda_1 \bar{\pi}_8, \\ \sum_{q \in \mathcal{Q}} \bar{\pi}_q &= 1. \end{aligned}$$

Solving (11), the stationary probabilities are given as

$$\begin{aligned} \bar{\pi}_0 &= \frac{\alpha \mu}{\Psi}, \quad \bar{\pi}_1 = \frac{\alpha \lambda_1 (\alpha + \mu + \lambda)}{(\mu + \alpha) \Psi}, \quad \bar{\pi}_2 = \frac{\alpha \lambda_2 (\alpha + \mu + \lambda)}{(\mu + \alpha) \Psi}, \\ \bar{\pi}_3 &= \frac{\alpha \mu}{\Psi}, \quad \bar{\pi}_4 = \frac{\lambda_1^2 \mu}{(\mu + \alpha) \Psi}, \quad \bar{\pi}_5 = \frac{\lambda_1 \lambda_2 \mu}{(\mu + \alpha) \Psi}, \\ \bar{\pi}_6 &= \frac{\lambda_2 \mu}{\Psi}, \quad \bar{\pi}_7 = \frac{\lambda_1 \lambda_2 \mu}{(\mu + \alpha) \Psi}, \quad \bar{\pi}_8 = \frac{\lambda_2^2 \mu}{(\mu + \alpha) \Psi}, \end{aligned} \quad (12)$$

where  $\Psi = (\lambda + \alpha)(\lambda + \mu)$ .

2) *Transition reset map matrices  $\mathbf{A}_l$  and  $\hat{\mathbf{A}}_l$* : The transitions and their effects on the continuous state  $\mathbf{x}(t)$  are summarized in Table I. In the following, we explain the transitions in detail.

- $l=1$ : A source 1 packet arrives at the transmitter server. With this transition the AoI of source 1 does not change, i.e.,  $x'_0 = x_0$ . This is because the arrival of source 1 packet does not yield an age reduction until it is delivered to the sink. Since the sink server is empty,  $x_1$  becomes irrelevant to the AoI of source 1 and thus, it does not matter what is assigned to  $x'_1$ . In this paper, when the system moves into a new state where  $x_i$  is irrelevant, we set  $x'_i = x_i, i \in \{1, 2\}$ . Thus, for the transition  $l = 1$ , we have  $x'_1 = x_1$ . Since the arriving source 1 packet is fresh and its age is zero, we have  $x'_2 = 0$ .
- $l=2$ : A source 2 packet arrives at the transmitter server. Similarly as for transition  $l = 1$ , we have  $x'_0 = x_0$  and  $x'_1 = x_1$ . Since the arriving packet is a source 2 packet, its delivery does not change the AoI of source 1, and thus, we have  $x'_2 = x_1$ . The reset maps of transition  $l = 11$  can be derived similarly.
- $l=3$ : A source 1 packet is in the transmitter server, the sink server is idle, and a source 1 packet arrives. According to the source-aware policy, the source 1 packet that is under service is preempted by the arriving packet.

TABLE I: Table of transitions for the Markov chain in Fig. 2

$l$	$q_l \rightarrow q'_l$	$\lambda^{(l)}$	$\mathbf{x}\mathbf{A}_l$	$\mathbf{A}_l$	$\hat{\mathbf{A}}_l$
1	00 $\rightarrow$ 10	$\lambda_1$	$[x_0 \ x_1 \ 0]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
2	00 $\rightarrow$ 20	$\lambda_2$	$[x_0 \ x_1 \ x_1]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
3	10 $\rightarrow$ 10	$\lambda_1$	$[x_0 \ x_1 \ 0]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
4	10 $\rightarrow$ 01	$\mu$	$[x_0 \ x_2 \ x_2]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
5	20 $\rightarrow$ 20	$\lambda_1$	$[x_0 \ x_1 \ x_1]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
6	20 $\rightarrow$ 02	$\mu$	$[x_0 \ x_0 \ x_2]$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
7	01 $\rightarrow$ 11	$\lambda_1$	$[x_0 \ x_1 \ 0]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
8	01 $\rightarrow$ 21	$\lambda_2$	$[x_0 \ x_1 \ x_1]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
9	01 $\rightarrow$ 00	$\alpha$	$[x_1 \ x_1 \ x_2]$	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
10	02 $\rightarrow$ 12	$\lambda_1$	$[x_0 \ x_0 \ 0]$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
11	02 $\rightarrow$ 22	$\lambda_2$	$[x_0 \ x_0 \ x_0]$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
12	02 $\rightarrow$ 00	$\alpha$	$[x_0 \ x_1 \ x_2]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
13	11 $\rightarrow$ 11	$\lambda_1$	$[x_0 \ x_1 \ 0]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
14	11 $\rightarrow$ 01	$\mu$	$[x_0 \ x_2 \ x_2]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
15	11 $\rightarrow$ 10	$\alpha$	$[x_1 \ x_1 \ x_2]$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
16	22 $\rightarrow$ 22	$\lambda_1$	$[x_0 \ x_0 \ x_0]$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
17	22 $\rightarrow$ 02	$\mu$	$[x_0 \ x_0 \ x_2]$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
18	22 $\rightarrow$ 20	$\alpha$	$[x_0 \ x_1 \ x_1]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
19	21 $\rightarrow$ 21	$\lambda_1$	$[x_0 \ x_1 \ x_1]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
20	21 $\rightarrow$ 01	$\mu$	$[x_0 \ x_1 \ x_2]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
21	21 $\rightarrow$ 20	$\alpha$	$[x_1 \ x_1 \ x_1]$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
22	12 $\rightarrow$ 12	$\lambda_1$	$[x_0 \ x_0 \ 0]$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
23	12 $\rightarrow$ 10	$\alpha$	$[x_0 \ x_1 \ x_2]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
24	12 $\rightarrow$ 02	$\mu$	$[x_0 \ x_0 \ x_2]$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Similarly as for transition  $l = 1$ , we have  $x'_0 = x_0$  and  $x'_1 = x_1$ . Since the arriving source 1 packet is fresh and its age is zero, we have  $x'_2 = 0$ . The reset maps of transitions  $l = 13$  and  $l = 22$  can be derived similarly.

- $l=4$ : The sink server is idle, and the source 1 packet in the transmitter server completes service and moves to the sink server. With this transition, we have  $x'_0 = x_0$  because this transition does not change the AoI at the sink. Since the source 1 packet moves to the sink server and its delivery would reduce the AoI to  $x_2$ , we have  $x'_1 = x_2$ . Since the transmitter server is empty,  $x_2$  becomes irrelevant and thus, we have  $x'_2 = x_2$ .
- $l=5$ : A source 2 packet is in the transmitter server, the sink server is idle, and a source 1 packet arrives. According to the source-aware policy, the arriving source 1 packet is blocked and cleared. Same as transition  $l = 1$ , we have  $x'_0 = x_0$  and  $x'_1 = x_1$ . Since the packet in the transmitter server is a source 2 packet, its delivery does not change the AoI of source 1, and thus we have  $x'_2 = x_1$ . The reset maps of transitions  $l = 16$  and  $l = 19$  can be derived similarly.
- $l=6$ : The sink server is idle and the source 2 packet in the transmitter server completes service and moves to the sink server. With this transition, we have  $x'_0 = x_0$ , and since the source 2 packet delivery does not change the AoI of source 1, we have  $x'_1 = x_0$ . Since the transmitter server is empty,  $x_2$  becomes irrelevant and thus, we have  $x'_2 = x_2$ .
- $l=7$ : The transmitter server is idle, a source 1 packet is in the sink server, and a source 1 packet arrives. With this transition, we have  $x'_0 = x_0$ . The delivery of the packet in the sink server would reduce the AoI to  $x_1$  thus, we have  $x'_1 = x_1$ . Since the arriving source 1 packet is fresh and its age is zero, we have  $x'_2 = 0$ .
- $l=8$ : The transmitter server is idle, a source 1 packet is in the sink server, and a source 2 packet arrives. Similarly as for transition  $l = 7$ , we have  $x'_0 = x_0$  and  $x'_1 = x_1$ . Since the arriving packet is a source 2 packet, its delivery does not change the AoI of source 1 and thus, we have  $x'_2 = x_1$ .
- $l=9$ : The transmitter server is idle and the source 1 packet in the sink server completes service and is delivered to the sink. With this transition, we have  $x'_0 = x_1$ . Since the servers become idle we have  $x'_1 = x_1$  and  $x'_2 = x_2$ . The reset maps of transition  $l = 12$  can be derived similarly.
- $l=10$ : The transmitter server is idle, a source 2 packet is in the sink server, and a source 1 packet arrives. With this transition, we have  $x'_0 = x_0$  because this transition does not change the AoI at the sink. The delivery of the source 2 packet in the sink server does not change the AoI of source 1 and thus  $x'_1 = x_0$ . Since the arriving source 1 packet is fresh and its age is zero, we have  $x'_2 = 0$ .
- $l=14$ : A source 1 packet is in the sink server and the source 1 packet in the transmitter server completes service and moves to the sink server. According to the source-aware policy, the source 1 packet that is in the sink

server is preempted by the arriving source 1 packet. With this transition, we have  $x'_0 = x_0$  because this transition does not change the AoI at the sink. Since the source 1 packet moves to the sink server, we have  $x'_1 = x_2$ . Since the transmitter server is empty we have  $x'_2 = x_2$ . The reset maps of transition  $l = 17$  can be derived similarly.

- $l=15$ : A source 1 packet is in the transmitter server and the source 1 packet in the sink server completes service and is delivered to the sink. With this transition, we have  $x'_0 = x_1$ . Since the sink server becomes idle we have  $x'_1 = x_1$ . The delivery of the source 1 packet in the transmitter server reduces the AoI to  $x_2$  and thus,  $x'_2 = x_2$ . The reset maps of transitions  $l \in \{18, 21, 23\}$  can be derived similarly.
- $l=20$ : A source 1 packet is in the sink server and the source 2 packet in the transmitter server completes service and moves to the sink server. According to the source-aware policy, the arriving source 2 packet is blocked and cleared. With this transition, we have  $x'_0 = x_0$ . The delivery of the packet in the sink server reduces the AoI to  $x_1$  and thus,  $x'_1 = x_1$ . Since the transmitter server is empty we have  $x'_2 = x_2$ . The reset maps of transition  $l = 24$  can be derived similarly.

3) *Calculation of the MGF of the AoI*: Recall from Section III that to calculate MGF of the AoI, we need to first ensure whether we can find non-negative vectors  $\bar{v}_q = [\bar{v}_{q0} \cdots \bar{v}_{qn}]$ ,  $\forall q \in \mathcal{Q}$ , that satisfy (6). Having derived the matrices  $\mathbf{A}_l$  for all transitions (see Table I), we can form the system of linear equations in (6). It can be shown that the system of linear equations in (6) has a non-negative solution. Consequently, using the derived  $\mathbf{A}_l$  and  $\hat{\mathbf{A}}_l$  for the transitions in Table I, we can form the system of linear equations in (7). By solving the system of linear equations we can find the values of  $\bar{v}_{q0}^s$ ,  $\forall q \in \mathcal{Q}$ . Substituting  $\bar{v}_{q0}^s$ ,  $\forall q \in \mathcal{Q}$ , into (10) results the MGF of the AoI of source 1 under the proposed policy, given by the following theorem.

**Theorem 1.** *The MGF of the AoI of source 1 under the proposed source-aware packet management policy is given by (13), as shown on the top of the next page, where  $\bar{s} = s/\mu$ ,*

$$\bar{\alpha} = \alpha/\mu, \bar{N} = \frac{\rho_1(\bar{\pi}_3 + \bar{\pi}_4)}{1 + \rho_1 + \bar{\alpha} - \bar{s}},$$

$$\bar{M} = \frac{\rho_1(\bar{\pi}_0 + \bar{\pi}_1)(1 + \rho_1 + \bar{\alpha} - \bar{s}) + \bar{\alpha}(\bar{\pi}_3 + \bar{\pi}_4 + \bar{\pi}_6 + \bar{\pi}_7)}{(1 + \rho_1 + \bar{\alpha} - \bar{s})(1 + \rho_1 - \bar{s})},$$

$$\xi_1 = \rho_1^3(3\bar{s} - 2\bar{s}^2 - 1) + \rho_1^2\rho_2(7\bar{s} - 6\bar{s}^2 - 1) + \rho_1^2(8\bar{s} - 15\bar{s}^2 + 8\bar{s}^3 - 1) + \rho_1\rho_2^2\bar{s}(5 - 6\bar{s}) + \rho_1\rho_2\bar{s}(8 - 24\bar{s} + 16\bar{s}^2) + \rho_1\bar{s}(3 - 15\bar{s} + 22\bar{s}^2 - 10\bar{s}^3) + \rho_2^3\bar{s}(1 - 2\bar{s}) + \rho_2^2\bar{s}(2 - 9\bar{s} + 8\bar{s}^2) + \rho_2\bar{s}(1 - 9\bar{s} + 18\bar{s}^2 - 10\bar{s}^3) - \bar{s}^2(1 - 8\bar{s} + 10\bar{s}^2 - \bar{s}^3),$$

$$\xi_2 = \rho_1^3(\bar{s} - 1) + \rho_1^2\rho_2(3\bar{s} - 2) + \rho_1^2(10\bar{s} - 7\bar{s}^2 - 3) + \rho_1\rho_2^2(1 - 3\bar{s}) + \rho_1\rho_2(14\bar{s} - 14\bar{s}^2 - 2) + \rho_2^3\bar{s} + \rho_2^2(4\bar{s} - 7\bar{s}^2) + \rho_1(10\bar{s} - 21\bar{s}^2 + 12\bar{s}^3 - 1) + \rho_2(4\bar{s} - 14\bar{s}^2 + 12\bar{s}^3) + \bar{s}(1 - 7\bar{s} + 12\bar{s}^2 - 6\bar{s}^3),$$

$$\xi_3 = 2\rho\rho_1(\bar{s} - 1) + 2\rho\bar{s}(\rho_2 + 2) + 2\rho_1(2\bar{s} - 1) - 6\rho\bar{s}^2 + 2\bar{s}(1 - 3\bar{s} + 2\bar{s}^2), \text{ and}$$

$$\xi_4 = \bar{s}^2 - \bar{s}(1 + \rho) - \rho_1.$$

## V. VALIDATIONS AND NUMERICAL RESULTS

By using the derived MGF, the  $m$ th moment of the AoI of source 1,  $\Delta_1^{(m)}$ , is given by  $\Delta_1^{(m)} = \frac{d^m(M_{\Delta_1}(s))}{ds^m} \Big|_{s=0}$ .

Fig. 3 illustrates the average AoI of source 1 and its standard deviation ( $\sigma$ ) as a function of  $\lambda_1$  for the proposed source-aware policy in comparison with the (source-agnostic) preemptive policy and the blocking policy studied in [14] for  $\mu = 1$ ,  $\alpha = 1$ , and  $\lambda = 5$ . As it can be seen, the preemptive policy outperforms the source-aware and blocking policies. The gap between the preemptive and the source-aware policies is (surprisingly) small, whereas the blocking policy underperforms by a large margin. In addition, when  $\lambda_1 \rightarrow \lambda$  (i.e.,  $\lambda_2 \rightarrow 0$ , and the system becomes single-source), the average AoI of the preemptive policy and the source-aware policy converge together, as expected.

Fig. 4 illustrates the Jain's fairness index [10] for the average AoI of sources 1 and 2 as a function of  $\lambda_1$  under the different packet management policies. As it can be seen, the blocking policy provides the fairest situation and the source-aware provides the lowest fairness in the system.

We would like to point out that based on our experiments (not included herein due to space limitation), the same trend for the curves of the average AoI and fairness among the three considered policies appears to hold true for any choice of the parameters.

## VI. CONCLUSIONS

We considered a multi-source system with computation-intensive status updates and derived the MGF of the AoI under a source-aware packet management policy using the SHS technique. Using the derived MGF, we evaluated the average AoI, its standard deviation, and fairness. The results showed that the two source-agnostic policies introduced in [14] provide better fairness than that of the policy introduced herein. Moreover, the preemptive policy in [14] provides the lowest average AoI amongst the three policies.

## ACKNOWLEDGEMENTS

This research has been financially supported by the Infotech Oulu, the Academy of Finland (grant 323698), and Academy of Finland 6Genesis Flagship (grant 318927). The work of M. Leinonen has also been financially supported in part by the Academy of Finland (grant 319485). M. Moltafet would like to acknowledge the support of Nokia, Riitta ja Jorma J. Takanen, and Tauno Tönning Foundations.

## REFERENCES

- [1] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. Commun. Society. Conf. on Sensor, Mesh and Ad Hoc Commun. and Net.*, Salt Lake City, UT, USA, Jun. 27–30, 2011, pp. 350–358.

$$M_{\Delta_1}(s) = \frac{-\bar{\alpha}(\bar{N} + \bar{M})(\bar{\alpha} + 1 - \bar{s})(\bar{\alpha} + \rho - \bar{s})(\rho + 1 - \bar{s})(\bar{\alpha} + \rho + 1 - \bar{s})}{\sum_{i=1}^4 \bar{\alpha}^i \xi_i - \bar{s}(1 - \bar{s})(\rho - \bar{s}) \left( (\bar{s} - 1)(\rho \rho_1 + 2\rho_2 \bar{s}) + \rho_2 \rho \bar{s} + \rho_1(3\bar{s} - 2\bar{s}^2 - 1) + \bar{s}(\bar{s} - 1)^2 \right)} \quad (13)$$

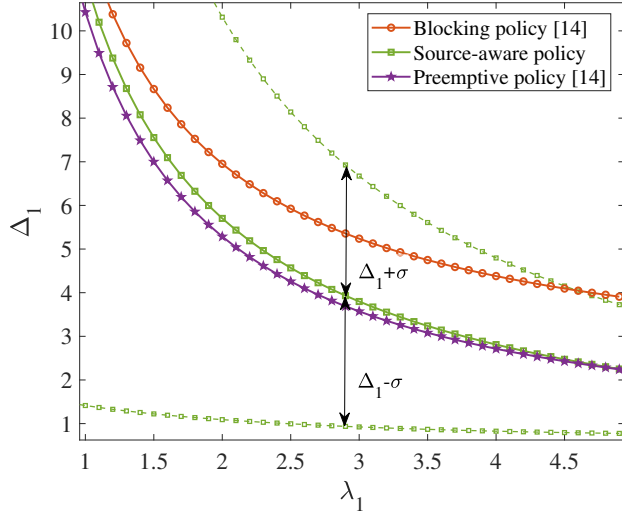


Fig. 3: Average AoI as a function of arrival rate of source 1 ( $\lambda_1$ ) with  $\mu = 1$ ,  $\alpha = 1$ , and  $\lambda = 5$ . The dashed lines visualize the standard deviation of the AoI as  $\Delta_1 + \sigma$  and  $\Delta_1 - \sigma$ .

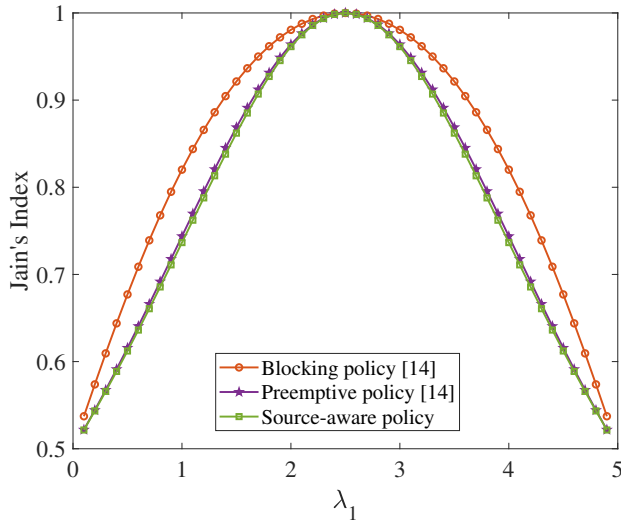


Fig. 4: The Jain's fairness index for the average AoI of sources 1 and 2 as a function of  $\lambda_1$  with  $\lambda = 5$ .

- [2] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE Int. Conf. on Computer Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 25–30, 2012, pp. 2731–2735.
- [3] S. K. Kaul, R. D. Yates, and M. Gruteser, "Status updates through queues," in *Proc. Conf. Inform. Sciences Syst. (CISS)*, Princeton, NJ, USA, Mar. 21–23, 2012, pp. 1–6.
- [4] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. Inform. Theory*, vol. 65, no. 3, pp. 1807–1827, Mar. 2019.
- [5] R. D. Yates, "The age of information in networks: Moments, distributions, and sampling," *IEEE Trans. Inform. Theory*, vol. 66, no. 9, pp. 5712–5728, May 2020.
- [6] S. K. Kaul and R. D. Yates, "Age of information: Updates with priority," in *Proc. IEEE Int. Symp. Inform. Theory*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2644–2648.
- [7] R. D. Yates, "Age of information in a network of preemptive servers," in *Proc. IEEE Int. Conf. on Computer Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 15–19 2018, pp. 118–123.
- [8] A. Javani, M. Zorghi, and Z. Wang, "Age of information in multiple sensing," in *Proc. IEEE Global Telecommun. Conf.*, Waikoloa, HI, USA, Dec. 9–13, 2019.
- [9] S. Farazi, A. G. Klein, and D. Richard Brown, "Average age of information in multi-source self-preemptive status update systems with packet delivery errors," in *Proc. Annual Asilomar Conf. Signals, Syst., Comp.*, Pacific Grove, CA, USA, 2019.
- [10] 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.
- [11] 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.
- [12] M. Moltafet, M. Leinonen, and M. Codreanu, "Average age of information in a multi-source M/M/1 queueing model with LCFS prioritized packet management," in *Proc. IEEE Int. Conf. on Computer Commun. (INFOCOM) AoI Workshop*, Toronto, Canada, Jul. 6–9, 2020, pp. 303–308.
- [13] —, "Average age of information for a multi-source M/M/1 queueing model with packet management," in *Proc. IEEE Int. Symp. Inform. Theory*, Los Angeles, CA, USA, Jun. 21–26, 2020, pp. 1765–1769.
- [14] M. Moltafet, M. Leinonen, and M. Codreanu, "Moment generating function of the AoI in multi-source systems with computation-intensive status updates," [Online] <https://arxiv.org/pdf/2102.01126.pdf>, 2021.
- [15] Q. Kuang, J. Gong, X. Chen, and X. Ma, "Analysis on computation-intensive status update in mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4353–4366, Feb. 2020.