Age-Oriented Scheduling of Correlated Sources in Multi-server System

Minh Hoang Ly[1], Josu Doncel[2], and Mohamad Assaad[1]
[1]*Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux & Systèmes (L2S), Gif-sur-Yvette, France*
[2]*UPV/EHU, University of the Basque Country, Leioa, Spain*
Email: hoang.ly-minh@student-cs.fr, josu.doncel@ehu.eus, mohamad.assaad@centralesupelec.fr

***Abstract*—This paper investigates a system formed by N sources, each of them sending status updates of a process of interest though a multi-server queue. Taking into consideration the mutual correlation across different sources, we propose an innovative model in which this correlation was embodied as an additional, virtual information source. Assuming that replication of packets to multiple servers is allowed, we present a policy to reduce the sum age of information (AoI) of these N sources. Our numerical results demonstrate the superiority of proposed policy when compared to the popular Last-Generate First-Serve policy.**

***Index Terms*—Age of Information, Scheduling policy, Correlated sources.**

I. Introduction

A. Introduction & Related works

The concept of Age of Information (AoI) was first introduced in 2011 in [1] to specify the freshness of knowledge obtained by a monitor about the status of a far-off source. Concretely, AoI is the time elapsed since the generation of the last successfully received message containing update information about the source of interest. When the monitor's most recently received update at time t has timestamp $N(t)$, i.e., the update is generated at time instant $N(t)$, the status update age, or simply the age, is defined as $\Delta(t) = t - N(t)$. The vast attention AoI has been enjoying is due to two factors. The first is the novelty provided by this new metric in characterizing the timeliness (or staleness) of information versus, for example, delay or throughput [2]. Second, the importance and need of characterizing the freshness of such information is paramount in a wide range of modern-day applications such as self-driving car or remote surgery. So far, the AoI has been studied and examined under a variety of systems, as a concept, a performance metric, and a tool [3].

Novel developments in communication technologies and the rise of connected devices led to the advent of various Internet of Things (IoT) systems where timely information updating is critical [4]. These systems often consist of multiple IoT devices, which need to transmit timely updates bringing information of a common underlying physical process to a shared destination [5]. Since these devices belong to a temporal and/or geographical proximity, their updates could be correlated in the sense that they are all associated with one common physical process. Recently, there has been several works investigating the AoI in systems of correlated sources/information/devices that we discuss next. In [6], authors investigate the use of correlated sources, i.e., sensing devices that transmit to a gateway periodic updates of an observed process and determine the optimal update strategy to improve data timeliness. It is shown that there is an optimal waiting time before updating such that estimation error at the gateway is lowest. Leveraging the correlation between sources, researchers in [7] consider the sensors' ability to decide to send, at each time, either actual information or differential information based on the receiver's feedback. Using this generalized scheme, the timeliness gains for some theoretical example sources are examined. Motivated by more practical systems such as wireless camera networks, the authors in [8] consider a system consisting of wireless camera nodes with overlapping fields of view. The joint optimization of processing node assignment and camera transmission scheduling is proposed to minimize the maximum peak AoI. Under some widely-used interference models, the formulated age minimization problem is proved to be NP-hard where a decomposition solution is proposed to solve the optimization problem efficiently. In [9], the authors study average AoI in a system where physical sources generate discrete-time, independent updates that are each observed by several sensors. Exploiting the common sensor information, two scheduling policies are proposed to minimize the AoI: one in which the system parameters are assumed known, the other one in which they are learned. Taking a look on a real-world IoT scenario, the authors in [4] propose an application-oriented scheduling to optimize data freshness in the presence of correlated information sources by formulating a Markov Decision Process (MDP) problem.

B. Contributions

In this work, we consider a system with N sources, each of them sending status updates of an underlying process through a transmission channel of multiple servers with packet replication capability. Considering the widely-used age metric that is sum AoI of all sources, we present a policy to further reduce this age penalty function at the monitor. Our work differs from [10] since we do not assume that the sources need to be synchronized. In the special case of single source, our proposed policy becomes the preemptive Last-Generate First-Serve with replication (from hereon, denoted as Prmp-LGFS-R) policy analyzed in [11], and when there are multiple but synchronized sources, it becomes the Maximum Age First, Last-Generate First-Serve (MAF-LGFS) policy discussed in [10]. We also consider the correlation between sources such that for each packet from a given source, there is a probability p_c that this packet also brings update information about all other sources. This model includes as a special case the case of uncorrelated sources, simply by setting the correlation probability to 0. With this in mind, we explore numerically the performance improvement of the proposed policy when compared to the well-known Last-Generate First-Serve (LGFS) policy. To the best of our knowledge, this is the first paper to examine the AoI in the broad setting of multi-source multi-server system that proposes a policy which numerically outperforms the widely-investigated LGFS.

C. Organization of the article

The rest of this paper is structured as follows. We present the system model and the considered age penalty function Section II. Section III then introduces the modified system model based on which we proposed a policy. Simulation results to demonstrate the performance of introduced policy are given in Section IV. Finally, we provide the main conclusions of the article in Section V.

II. Model Description

A. Notation

We study a system where N processes of interest must be observed by a monitor through one common transmission channel formed of 1 queue with infinite buffer and m servers. Each process is captured by an information source that sends its status updates to the common queue from time to time, which will forward these updates to the available servers. We will use the term *source* and *flow* interchangeably and denote λ_j as the arrival rate of source/flow $j, j \in \{1, 2, \ldots, N\}$. Moreover, we assume that the inter-arrival times between packets from the same sources and between different sources are *i.i.d.* distributed and $\lambda_1 = \lambda_2 = \cdots = \lambda_N = \lambda$. These information flows are not independent but assumed to be correlated. Concretely, for each packet from any source, there is a correlation probability p_c that this packet also brings update information about all other sources. This assumption also includes the case of independents sources as a special case in which $p_c = 0$. Note that under this assumption, for each flow $j, j \in \{1, 2, \ldots N\}$, the age of flow j at the monitor can be updated by packets from source j itself (with probability 1) or by packets from other sources (with probability p_c). System model is depicted in Figure 1.

The packet service times are *i.i.d.* across servers and the packets assigned to the same server, and are independent of the packet generation and arrival events. We remark that in our model packets may arrive at the buffer *out of order* of their generation times, i.e., packets generated earlier may arrive at the buffer later.

We assume that one packet can be replicated to several servers and the first replica/copy to complete service is considered as the valid execution of the packet. After that, the remaining replicas of this packet are dropped immediately to free the servers. The maximum replication degree is denoted by r $(1 \le r \le m)$ and it is the maximum number of servers that a packet can be replicated to. Moreover, we assume that preemption is allowed, i.e., any server may stop serving a packet currently under service and switch to process another.

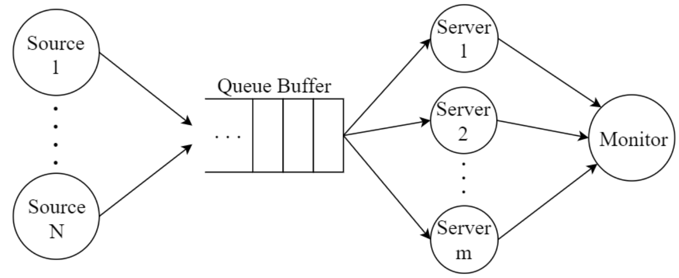

Fig. 1. Different sources send status updates to a common monitor via a shared queue. These sources are correlated with correlation probability p_c.

We denote by $\mathbf{A}(t)$ the vector tracking generation times of all the packets currently in the queue at time t and by $\mathbf{B}(t)$ the vector containing generation times of all the packets that have arrived at the monitor before time t. A *policy* $P = P(t) = P(\mathbf{A}(t), \mathbf{B}(t))$ determines which packet is served at time t, i.e.,

$$P(t) = P(\mathbf{A}(t), \mathbf{B}(t)) : \left[\mathbf{A}(t), \mathbf{B}(t)\right] \longmapsto \text{a packet } p \text{ to serve.}$$

Let $\mathcal{P}$ denotes the set of *causal, deterministic* policies in which the scheduling decisions are made based on the history and current states of the system. We define $\mathcal{P}_r$ as the set of all policies in $\mathcal{P}$ when the maximum replication degree is r. Hence, it follows by definition that:

$$\mathcal{P}_1 \subset \mathcal{P}_2 \subset \cdots \subset \mathcal{P}_m.$$

B. Problem Formulation

At each time t, denote $b_1(t)$ as the generation time of the packet having largest generation time (i.e., the freshest) of all the packets bringing information about the process of interest of source 1 ($b_1(t) = 0$ if none of the packets bringing information about source 1 has arrived at the Monitor by time t). At time t, the AoI of source 1 is

$$\Delta_1(t) = t - b_1(t).$$

Likewise, we define $b_2(t), \ldots, b_N(t)$ and $\Delta_2(t), \ldots, \Delta_N(t)$. Let

$$\mathbf{b}(t) = [b_1(t), b_2(t), \ldots, b_N(t)]$$

be the generation time vector of freshest packets at the monitor of N processes of interest. In this article, we are interested in reducing the instantaneous sum of AoI of all the processes of interest and, thus, we define the following age penalty function

$$\mathbf{\Delta}(t) = \sum_{i=1}^{N} \Delta_i(t)$$

Therefore, for a given replication degree r, we aim to find a scheduling policy that alleviates $\mathbf{\Delta}(t)$. Taking into account that:

$$\begin{aligned}\mathbf{\Delta}(t) &= \sum_{i=1}^{N} \Delta_i(t) \\ &= Nt - \sum_{i=1}^{N} b_i(t) \\ &= Nt - S(t),\end{aligned}$$

where

$$S(t) = \sum_{i=1}^{N} b_i(t)$$

is the sum of generation times of all N sources at the monitor, the above minimization problem is equivalent to maximize

$$S(t) = \sum_{i=1}^{N} b_i(t).$$

In the next section, we present the policy to address the problem above and in Section IV we compare its performance with another policy.

Remark 1. *We would like to remark that the results presented in this article can be generalized to an age-penalty function that consists of a linear combination of the AoI of all the processes of interest,*

$$\sum_{i=1}^{N} c_i \Delta_i(t),$$

where $\sum_i c_i = 1$.

III. Main Results

In this section, we present the main results of this article. First, we propose an adjusted model in which the correlation between sources is modelized as an additional source. Based on this altered model, we present a policy that prioritizes the packet with the maximum generation time increment and show that it is an extension of other policies that have been proven to be optimal in other settings.

A. Modified system model

In order to leverage the correlation between sources to minimize the age penalty function at the monitor, we now propose a slightly alternative system setting. Consider the system model described in section II.A with N information flows. The mentioned correlation between these sources is now modelized as an additional, 'virtual' source $N+1$ whose packets bring information about all other N sources. We therefore can assume that the other N sources now become *independent, uncorrelated*. If we denote λ_{N+1} as the arrival rate of packets from this additional source, we may have $\lambda_{N+1} = p_c\lambda$. In general, since $p_c < 1$, the arrival rate of source $N+1$ is smaller than that of the first N sources. Figure 2 illustrates the modified model. Note that under this altered

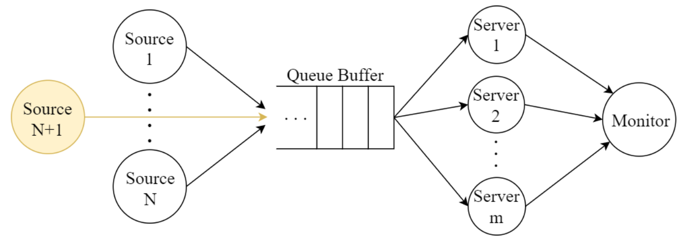

Fig. 2. Modified model where $N+1$ sources send their status updates to a common monitor via a shared queue. First N sources are independent and source $N+1$ captures information about all other sources.

model, for each flow $j, j \in \{1, 2, \dots N\}$, packets bringing information about this flow can only come from either source j itself or source $N+1$.

B. Description of the Proposed Policy

We now consider the preemptive *Maximum Generation Time Increment First (MGTIF)* policy. The intuition behind the policy is as follows: since we want to minimize the sum AoI of all sources, we will prioritize whatever packet that would cause the largest age reduction at the monitor, or equivalently, trigger the biggest generation-time increment. Consequently, this policy will not always pick the last generated packets in the queue to serve.

For each packet p currently in the queue waiting to be served, we define by t_p the generation time of that packet. Hence, we denote by $u(p)$ the *utility function* of packet p and we define it as follows:

$$u(p) = \begin{cases} S(t) + \max\{t_p, b_j(t)\} - b_j(t) & \text{if } p \text{ is from source } j,\ j \le N \\ \sum_{i=1}^{N} \max\{t_p, b_i(t)\} & \text{if } p \text{ is from source } N+1 \end{cases} \quad (1)$$

We highlight that, if packet p is from source $j \le N$, the term $\max\{t_p, b_j(t)\} - b_j(t)$ denotes the generation time increment about source j at the monitor when this packet p is delivered. Consequently, $S(t) + \max\{t_p, b_j(t)\} - b_j(t)$ denotes the updated sum of generation times of all information flows (at the monitor) right after packet p finishes its service and arrives at the monitor. If packet p is from source $N+1$, the corresponding expression is

$$\sum_{i=1}^{N} \max\{t_p, b_i(t)\}.$$

Note that because of the out-of-order arrivals, the packets arrived in the queue later may have smaller generation time as compared to other packets, and thus can have smaller utility function.

The MGTIF policy consists of picking, at each time t, the packet with the maximum utility. Formally,

$$\pi(\mathbf{A}(t), \mathbf{B}(t)) = \pi(t) : \quad [\mathbf{A}(t), \mathbf{B}(t)] \longmapsto \underset{\{p | t_p \in \mathbf{A}(t)\}}{\text{argmax}} \ u(p),$$

with ties broken arbitrarily.

One can easily see that when there is only 1 source of information, there is no mutual correlation between different sources, and the term

$$\max\{t_p, b_j(t)\} - b_j(t)$$

will be maximized by the packet that was most lastly-generated. Therefore the MGTIF reduces to the Last-Generate First-Serve (LGFS) discussed in [11]. Moreover, when there are multiple but synchronized sources, update packets from all sources are synchronized and arrive to the queue in batches. Therefore, within each synchronized batch, the synchronized packets have the same $S(t)$ and the same generation time t_p. Determining the packet with maximum utility $u(p)$ now becomes finding the minimum $b_j(t)$, i.e., finding the most outdated flow (flow with largest age). In this case, the proposed policy becomes the Maximum Age First, Last-Generate First-Serve (MAF-LGFS) policy proposed in [10].

We now present the preemptive Maximum Generation Time Increment First with replication (from hereon, abbreviated as Prmp-MGTIF-R), which is an extension of MGTIF in which replication of packet with degree $r \geq 1$ is allowed. In Algorithm 1, we present the Prmp-MGTIF-R policy. In the remainder of this article, we denote by π this policy. Now, we describe how it works. We replicate the best packet found by policy π in the queue on r servers, and this packet is then discarded from the waiting queue. Next, we replicate the best packet found by π (among the remaining packets) in the waiting queue on the remaining idle servers such that its number of replicas does not exceed r, and so on. Since m may not be an integer multiple of r, packets under service may not be evenly distributed among the servers (degree of replication between packets may differs). If this is the case, we prioritize the $k = \left\lfloor \frac{m}{r} \right\rfloor$ packets for the first k packets found by policy π and each of them is replicated on exactly r servers. The next best packet found by π is replicated on the remaining idle servers (whose number is less than r). Otherwise, if $m = ar, a \in \mathbb{Z}_+$, then all packets under service are evenly distributed, each one of them is replicated on r servers. The implementation details when $r \geq 1$ are depicted in Algorithm 1. Concretely:

- Packet arrival event: If a new packet p_i arrives, we need to check whether or not this packet preempts any other packet that is being served in Steps 6-17. Then, if packet p_i is to be served, determine the number of copies need to be replicated for packet p_i in Steps 19-27. If packet p_i is to be served, either 1 of 3 following cases must happen:
  1) **Case 1:** $u(p_i) \geq \alpha$, α being the smallest utility function of all packets in the set Q. In this scenario, we need to replicate packet p_i on r idle servers. Hence if $I \geq r$, we replicate packet p_i on r servers. Steps 19-20 depicts these procedures.
  2) **Case 2:** Same scenario as Case 1 but the number of available servers (I) is less than r, we preempt $(r - I)$ replicas of the packet with smallest utility function in the set Q and replicate packet p_i on $(r - I) + I = r$ servers. Steps 21-23 illustrates these procedures.
  3) **Case 3:** $u(p_i) = \alpha$. In this case, packet p_i is replicated on the available idle servers such that the total number of replicas of packet p_i does not exceed r, as described in Steps 24-25.
- Packet departure event: If a packet p_j is delivered, we rescind all the remaining replicas of packet p_j to free the servers. Then pick the packet with largest utility function in the queue (found by π) and replicate it on the available idle servers such that the total number of replicas of this packet does not exceed r. Steps 29-39 depicts these procedures.

Algorithm 1: Prmp-MGTIF-R policy.

1 $Q := \emptyset$; // Q is the set of distinct packets that are under service
2 $\alpha := 0$; // $\alpha := \min\{u(p) : p \in Q\}$
3 $I := m$; // I is the number of idle servers
4 **while** *the system is ON* **do**
5 **if** *a new packet p_i arrives* **then**
6 **if** $I = 0$ **then**
// All servers are busy
7 **if** $u(p_i) \leq \alpha$ **then**
// packet p_i won't help reduce age penalty function
8 Store packet p_i in the queue;
9 **else**
10 Find $p_j \in Q$ whose utility function equals α;
11 Preempt all replicas of packet p_j;
12 Packet p_j is stored back to the queue $Q := Q \cup \{p_i\} - \{p_j\}$;
13 Update I;
14 **end**
15 **else** // At least one of the servers is idle
16 $Q := Q \cup \{p_i\}$;
17 **end**
18 $\alpha := \min\{u(p) : p \in Q\}$;
19 **if** $p_i \in Q$ ***and*** $u(p_i) > \alpha$ ***and*** $I \geq r$ **then**
20 Replicate packet p_i on r idle servers;
21 **else if** $p_i \in Q$ ***and*** $u(p_i) > \alpha$ ***and*** $I < r$ **then**
22 Preempt $(r - I)$ replicas of the packet whose utility function equals α;
23 Replicate packet p_i on r idle servers;
24 **else**
25 Replicate packet p_i on $\min\{r, I\}$ idle servers;
26 **end**
27 Update I;
28 **end**
29 **if** *a packet p_l is delivered* **then**
30 Cancel the remaining replicas of packet p_l;
31 $Q := Q - \{p_l\}$;
32 **if** *the queue is not empty* **then**
33 Pick the packet with the largest utility function in the queue $p_h : p_h = \text{argmax}_{p \in Q} u(p)$;
34 $Q := Q \cup \{p_h\}$;
35 Replicate packet p_h on at most $\min\{r, I\}$ servers;
36 Update I;
37 **end**
38 $\alpha := \min\{u(p) : p \in Q\}$;
39 **end**
40 **end**

IV. Numerical Results

In this section we present our numerical work in which we compare the performance of the proposed Prmp-MGTIF-R policy with performance of the Prmp-LGFS-R policy [11]. We consider that the arrival rates satisfy that

$$\lambda_1 = \lambda_2 \cdots = \lambda_N = \lambda$$

and

$$\lambda_{N+1} = p_c \lambda.$$

We assume that the packet service times are exponentially distributed with mean $1/\mu = 1$ and the inter-generation times are *i.i.d.* Erlang-2 distributed with mean $1/\lambda$. The number of servers is m and number of independent sources is $N = 5$. Hence, the traffic intensity is $\rho = \lambda/m\mu$. The queue size is infinity. We also assume that the time difference between packet generation and arrival to the queue is zero, i.e., the updates arrive in the queue instantly after their generation.

Figure 3 illustrates the time-average sum age versus traffic intensity ρ for a system of $N = 5$ independent sources, $m = 4$ servers. We can observe that the proposed Prmp-MGTIF-R policy outperforms the Prmp-LGFS-R significantly. Moreover, the age gap between the two policies becomes larger as the traffic intensity ρ increases. Furthermore, we notice that $\mathbf{\Delta}(t)$ under Prmp-LGFS-R stops reducing when $\rho > 0.5$ while $\mathbf{\Delta}(t)$ under Prmp-MGTIF-R always decreases with ρ. Another important property we derive from this simulation is that the age gap between the two policies also grows with correlation probability p_c. In other words, the Prmp-LGFS-R was not able to fully leverage the correlation between sources to reduce age in the high traffic intensity regime. The reason is that new packets arrive more frequently when traffic intensity is high, and since the LGFS policy always prioritizes fresher packets, the packets from additional source (source $N + 1$) (i.e., packets bringing much information about other sources) are likely to be preempted during its service time by another packet. Another rationale is that since we are minimizing the sum age of all sources, it is more urgent to update the outdated flows (i.e., flows currently having high ages) with their new packets, rather than to transmit updates from flows that are already newly-updated.

In Figure 4, we depict the time-average Peak sum age (denoted by peak $\mathbf{\Delta}(t)$ in the plot) of the system with same parameters as in Figure 3 but with replication degree $r = 4$ and we observe that the improvement of our policy satisfies the same properties as in Figure 3. From this plot, it is seen that the Average Peak $\mathbf{\Delta}(t)$ in Figure 4 where $r = 4$ has almost the same (or even smaller) values as the Average $\mathbf{\Delta}(t)$ in Figure 3 where $r = 2$. In other words, increasing r helps reduce the age appreciably. Besides, boosting correlation probability p_c also helps reduce age, which can be simply explained by the fact that when p_c is high, more information could be encoded within one packet, which will lead to a larger age reduction at the monitor.

Finally, Figure 5 considers the above performance metrics in a system with $N = 5$ independent sources, correlation probability $p_c = 0.2$, traffic intensity $\rho = 0.6$ and replication degree $m/2$. The aim is to study the benefit of increasing number of servers m on reducing age. It is shown that the Prmp-MGTIF-R policy improves substantially the performance compared to the Prmp-LGFS-R, especially with small values of m. One can observe that the age gap between the 2 policies diminishes when m increases. We conclude that the disadvantage of Prmp-LGFS-R has been partly compensated by the huge number of available servers.

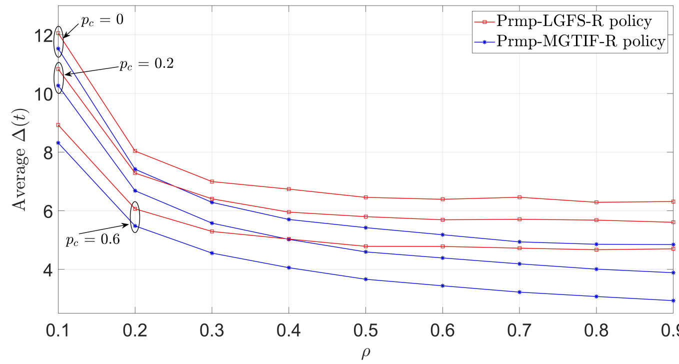

Fig. 3. Average sum age under 2 policies: Prmp-LGFS-R (red) and Prmp-MGTIF-R (blue) with various values of correlation probability p_c, number of servers $m = 4$, replication degree $r = 2$.

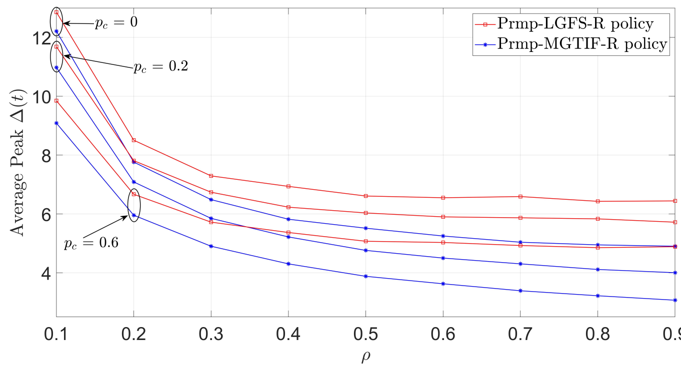

Fig. 4. Average Peak Sum age (peak $\mathbf{\Delta}(t)$) under Prmp-LGFS-R (red) and Prmp-MGTIF-R (blue) with various values of correlation probability p_c, number of servers $m = 4$, replication degree $r = 4$.

V. Conclusion

In this paper, we considered a system with N processes of interest in which each process sends its packets to a far-off monitor to update its status. These packets are sent through a communication channel consists of a queue with infinite buffer and m servers. We also took into account the mutual correlation between these processes, which was then modelized as an additional, virtual source $N + 1$ in our modified model. Moreover, preemption of packets is allowed, which means that a packet in service can be stopped. Furthermore, we assume that replication of packets to several servers is possible. In this context, we presented a policy that determines which packet to pick in order to reduce, as much as possible, the sum of instantaneous AoI of the N processes of interest.

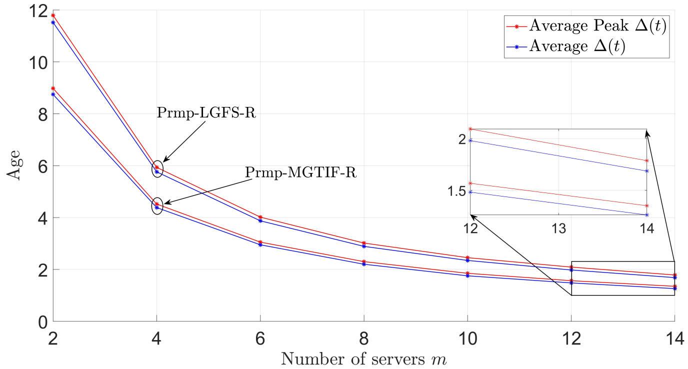

Fig. 5. Average Sum age and Average Peak Sum age (peak $\boldsymbol{\Delta}(t)$) under Prmp-LGFS-R and Prmp-MGTIF-R with different numbers of server m, replication degree $r = m/2$, $N = 5$, correlation probability $p_c = 0.2$, traffic intensity $\rho = 0.6$.

We also showed that this policy is an extension of the well-known LGFS and MAF-LGFS policies. Finally, we studied numerically the performance of our proposed policy when compared with the LGFS. We have seen that, in the considered cases, it outperforms the LGFS policy. For instance, in the absence of correlation, i.e., when $p_c = 0$, the policy we propose performs clearly better than LGFS (note that LGFS policy is shown to be optimal for systems of single source and multiple but synchronized sources [10, 11], but not for the system model under study here).

For future work, we are interested in developing a mathematical and rigorous analysis of the proposed policy and to provide a proof of the optimal policy for this system model. Besides, we would also like to extend the current model in several directions. Potential extensions include performance analysis when considering a more general age-penalty function, packet transmissions with errors, and partially correlated information sources, i.e., correlation exists within an exclusive subset of sources only.

VI. Acknowledgements

The work of Josu Doncel has been supported by the Department of Education of the Basque Government through the Consolidated Research Group MATHMODE (IT1294-19), by the Marie Sklodowska-Curie grant agreement No 777778 and by the Spanish Ministry of Science and Innovation with reference PID2019-108111RB-I00 (FEDER/AEI).

References

[1] Kaul, S. *et al.*, "Minimizing age of information in vehicular networks", *8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 350–358.

[2] M. A. Abd-Elmagid, N. Pappas and H. S. Dhillon, "On the Role of Age of Information in the Internet of Things," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 72-77, December 2019, doi: 10.1109/MCOM.001.1900041.

[3] Antzela Kosta, Nikolaos Pappas and Vangelis Angelakis (2017), "Age of Information: A New Concept, Metric, and Tool," *Foundations and Trends in Networking*, vol. 12, no. 3, pp 162–259, doi: 10.1561/1300000060.

[4] B. Yin, S. Zhang and Y. Cheng, "Application-Oriented Scheduling for Optimizing the Age of Correlated Information: A Deep-Reinforcement-Learning-Based Approach," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8748-8759, Sept. 2020.

[5] Bo Zhou and Walid Saad, "On the Age of Information in Internet of Things Systems with Correlated Devices," *arXiv preprint arXiv:2001.11162v3*, 2020.

[6] J. Hribar, M. Costa, N. Kaminski and L. A. DaSilva, "Updating Strategies in the Internet of Things by Taking Advantage of Correlated Sources," *2017 IEEE Global Communications Conference*, Singapore, 2017, pp. 1-6.

[7] S. Poojary, S. Bhambay and P. Parag, "Real-time status updates for correlated source," *2017 IEEE Information Theory Workshop (ITW)*, Kaohsiung, 2017, pp. 274-278.

[8] Q. He, G. Dán and V. Fodor, "Joint Assignment and Scheduling for Minimizing Age of Correlated Information," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 1887-1900, Oct. 2019.

[9] A. E. Kalør and P. Popovski, "Minimizing the Age of Information From Sensors With Common Observations," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1390-1393, Oct. 2019, doi: 10.1109/LWC.2019.2919528.

[10] Y. Sun, E. Uysal-Biyikoglu and S. Kompella, "Age-Optimal Updates of Multiple Information Flows," *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Honolulu, HI, 2018, pp. 136-141.

[11] A. M. Bedewy, Y. Sun and N. B. Shroff, "Minimizing the Age of Information Through Queues," *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 5215-5232, Aug. 2019, doi: 10.1109/TIT.2019.2912159.