Status Updates with Priorities: Lexicographic Optimality

Ali Maatouk*, Yin Sun§, Anthony Ephremides†, and Mohamad Assaad*

*TCL Chair on 5G, Laboratoire des Signaux et Systèmes, CentraleSupélec, Gif-sur-Yvette, France
§ECE Dept., Auburn University, Auburn, AL 36849
†ECE Dept., University of Maryland, College Park, MD 20742

***Abstract*—In this paper, we consider a transmission scheduling problem, in which several streams of status update packets with diverse priority levels are sent through a shared channel to their destinations. We introduce a notion of *Lexicographic age optimality*, or simply *lex-age-optimality*, to evaluate the performance of multi-class status update policies. In particular, a lex-age-optimal scheduling policy first minimizes the Age of Information (AoI) metrics for high-priority streams, and then, within the set of optimal policies for high-priority streams, achieves the minimum AoI metrics for low-priority streams. We propose a new scheduling policy named Preemptive Priority, Maximum Age First, Last-Generated, First-Served (PP-MAF-LGFS), and prove that the PP-MAF-LGFS scheduling policy is lex-age-optimal in the single exponential server settings. This result holds (i) for minimizing any time-dependent, symmetric, and non-decreasing age penalty function; (ii) for minimizing any non-decreasing functional of the stochastic process formed by the age penalty function; and (iii) for the cases where different priority classes have distinct arrival traffic patterns, age penalty functions, and age penalty functionals. For example, the PP-MAF-LGFS scheduling policy is lex-age-optimal for minimizing the mean peak age of a high-priority stream and the time-average age of a low-priority stream. Numerical results are provided to illustrate our theoretical findings.**

I. Introduction

Due to the proliferation of cheap hardware, remote monitoring has become the norm for modern technology applications. In these applications, a monitor is interested in timely updates about the status of a remote system. These status updates range from vehicles' position and velocity in autonomous driving to the temperature and humidity levels of a certain area in environmental monitoring. To capture this notion of timeliness, the Age of Information (**AoI**), which is defined as the information time lag at the monitor, has been introduced [1]. Due to its widespread application range and its ability to quantify the freshness of information, the AoI is regarded as a fundamental performance metric in communication networks. The AoI has attracted a significant surge of interest in recent years [1]–[16]. In particular, transmission scheduling of multiple update streams in both centralized and distributed settings has been explored in [17]–[24]. For example, the authors in [17] proposed both age-optimal and near age-optimal scheduling policies for the single and multi-server cases, respectively.

This work has been supported by ONR N000141812046, NSF CCF1813078, NSF CNS1551040, NSF CCF1420651, and ONR N00014-17-1-2417.

In a variety of real-life applications, information streams are assigned different priorities based on how crucial and time-sensitive their data are. A simple example is a vehicular network where data can be divided into two categories: crucial safety data and non-safety-related information. As the former is more time-sensitive than the latter, it should always be given a higher priority by the service facility [25]. Accordingly, priority-based scheduling problems have been extensively studied in the queuing theory literature for different performance measures (e.g., delay [26], throughput [27]). In [27], a notion of Lexicographic optimality, or simply, lex-optimality, was introduced for throughput maximization in multi-class scheduling scenarios. The idea of lex-throughput-optimality is to first find a class of optimal scheduling policies Π_{opt} that maximize the throughput of a high priority class, and then find the optimal scheduling policies within Π_{opt} that maximize the throughput of the low priority class. Therefore, it is clear how it elegantly provides high priority streams the best possible service by the facility, and at the same time, optimize the performance of the low priority streams.

There exist several recent studies on status updates with multiple priority classes. In [28], the authors considered multiple information streams, each with a different priority, and sharing a common service facility with no buffer space (i.e., low priority packets that are preempted by higher priority packets are discarded). Using a tool named Stochastic Hybrid Systems (**SHS**), the authors found an expression of the average age of each stream. The arrival rate of each stream was then optimized accordingly. In another work [29], the authors investigated the same settings of [28] but by letting each stream have its own buffer space. Most recently, closed-form expressions of the average Peak Age of Information (**PAoI**) were found in M/M/1/1 settings where streams are assigned different priorities [30]. As can be seen from the past works in the literature, the research efforts lay mainly in finding closed-form expressions of the average AoI/PAoI in a particular scenario, and for a specific arrival traffic model, to provide insights on the performance of the system. Accordingly, the question of what is the age-optimal scheduling policy in a multi-class priority-based scheduling scenario remains open. In our paper, we find an answer to this question. To that end,

ISBN 978-3-903176-29-4 © 2020 IFIP

we summarize in the following the key contributions of this paper:

- We introduce the notion of Lexicographic optimality for the age minimization framework, which we will refer to as the lex-age-optimality. The lex-age-optimality elegantly captures both the age-optimality and the order of time-cruciality between the streams in a general multi-class scheduling scenario. This approach guarantees that the performance of low priority streams is optimized while ensuring that high priority streams are granted the best possible service by the facility.
- In the case of a single server with i.i.d. exponential service times, we propose the *Preemptive Priority, Maximum Age First, Last-Generated, First-Served* (**PP-MAF-LGFS**) scheduling policy. Using a sample-path argument, we show that this policy is lex-age-optimal. Our lex-age-optimality results are not constrained to the traditional minimization of the average AoI and PAoI frameworks previously adopted in [28]–[30]. In fact, they hold for (i) minimizing any time-dependent, symmetric, and non-decreasing penalty function of the ages, and (ii) minimizing any non-decreasing functional of the age penalty process. Moreover, our lex-age-optimality results hold when the priority classes have distinct traffic patterns and different dissatisfaction levels of the aged information. This showcases the wide scope of our results as classes typically represent diverse applications, each with its data timeliness requirements. For example, we could be interested in minimizing the average PAoI for a class and the average AoI for another.

The rest of the paper is organized as follows: Section II is dedicated to the system model where the required definitions and the queuing model are presented. In Section III, we introduce the notion of lex-age-optimality and propose a lex-age-optimal policy in the single exponential server settings. Numerical results that corroborate these findings are laid out in Section IV while the paper is concluded in Section V. The results in this paper can be potentially used as a basis for future research directions, such as establishing lex-age-optimal policies for more general service time distributions, multi-server settings, and multi-hop networks.

II. System Model

A. Notations and Definitions

We let x and $\boldsymbol{x}$ denote deterministic scalars and vectors respectively. Similarly, we will use X and $\boldsymbol{X}$ to denote random scalars and vectors respectively. Let x^i denote the i-th element of vector $\boldsymbol{x}$, and let $x^{[i]}$ denote the i-th largest element of vector $\boldsymbol{x}$. Hence, $x^{[1]}$ and $x^{[N]}$ denote the largest and smallest elements of vector $\boldsymbol{x}$ respectively. We denote by $[\boldsymbol{x}]$ the sorted version of vector $\boldsymbol{x}$ where elements are arranged in descending order (i.e., $[\boldsymbol{x}]^i = x^{[i]}$). Vector $\boldsymbol{x} \in \mathbb{R}^N$ is said to be smaller than $\boldsymbol{y} \in \mathbb{R}^N$, denoted by $\boldsymbol{x} \leq \boldsymbol{y}$, if $x^i \leq y^i$ for $i = 1, \ldots, N$. The composition of two functions f and g is denoted by $f \circ g(\boldsymbol{x}) = f(g(\boldsymbol{x}))$. A function $p : \mathbb{R}^N \mapsto \mathbb{R}$ is said to be symmetric if $p(\boldsymbol{x}) = p([\boldsymbol{x}])$ for all $\boldsymbol{x} \in \mathbb{R}^N$. Next, we define stochastic ordering, which we will use in our subsequent age-optimality analysis.

Definition 1. *Stochastic Ordering of Random Variables* [31]: A random variable X is said to be stochastically smaller than a random variable Y, denoted by $X \leq_{st} Y$, if $\Pr(X > t) \leq \Pr(Y > t)\ \forall t \in \mathbb{R}$.

Definition 2. *Stochastic Ordering of Random Vectors* [31]: A set $\mathscr{U} \subseteq \mathbb{R}^N$ is called upper if $\boldsymbol{y} \in \mathscr{U}$ whenever $\boldsymbol{x} \leq \boldsymbol{y}$ and $\boldsymbol{x} \in \mathscr{U}$. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two n-dimensional random vectors, $\boldsymbol{X}$ is said to be stochastically smaller than $\boldsymbol{Y}$, denoted by $\boldsymbol{X} \leq_{st} \boldsymbol{Y}$, if for all upper sets $\mathscr{U} \subseteq \mathbb{R}^N$

$$\Pr(\boldsymbol{X} \in \mathscr{U}) \leq \Pr(\boldsymbol{Y} \in \mathscr{U}). \tag{1}$$

Definition 3. *Stochastic Ordering of Stochastic Processes* [31]: A stochastic process $\{X(t), t \geq 0\}$ is said to be stochastically smaller than a stochastic process $\{Y(t), t \geq 0\}$, denoted by $\{X(t), t \geq 0\} \leq_{st} \{Y(t), t \geq 0\}$, if for any sequence of time instants $t_1 < t_2 < \ldots < t_m \in \mathbb{R}^+$

$$(X(t_1), X(t_2), \ldots, X(t_m)) \leq_{st} (Y(t_1), Y(t_2), \ldots, Y(t_m)). \tag{2}$$

Let $\mathbb{V}$ be the set of Lebesgue measurable functions on $[0, \infty)$, i.e.,

$$\mathbb{V} = \{g : [0, \infty) \mapsto \mathbb{R} \text{ is Lebesgue measurable}\}. \tag{3}$$

A functional $\phi : \mathbb{V} \mapsto \mathbb{R}$ is said to be non-decreasing if $\phi(g_1) \leq \phi(g_2)$ holds for all $g_1, g_2 \in \mathbb{V}$ that satisfy $g_1(t) \leq g_2(t)$ for $t \in [0, \infty)$. We note that $\{X(t), t \geq 0\} \leq_{st} \{Y(t), t \geq 0\}$ if, and only if, [31]

$$\mathbb{E}[\phi(\{X(t), t \geq 0\})] \leq \mathbb{E}[\phi(\{Y(t), t \geq 0\})] \tag{4}$$

holds for every non-decreasing functional ϕ for which the expectations in (4) exist.

B. Queuing Model

Consider the status-update system illustrated in Fig. 1, where N streams of update packets are sent through a common service facility. Each update stream has a buffer space, which can be infinite or finite. The server can process at most one packet at a time. The packet service times are i.i.d. across streams and time. The information streams are divided into I priority classes, with streams of the same class i having the same priority. Each information stream is indexed by two components (i, j), where i denotes the class index and j denotes the stream index within class i. The classes are indexed in a decreasing order of priority. In other words, classes 1 and I are the highest and lowest priority classes, respectively. Let J_i be the number of steams in class i. Let $s_{i,j}$ and $d_{i,j}$ denote the source and destination nodes of stream (i, j), respectively. Different streams can have different source and/or destination nodes.

The system starts operating at time $t = 0$. The n-th update packet of stream (i, j) is generated at time $S_n^{i,j}$, arrives to the stream's buffer at time $A_n^{i,j}$, and is delivered to the

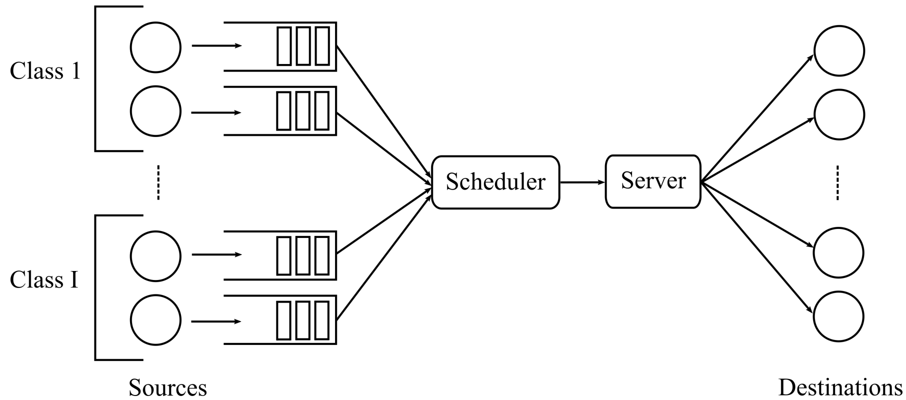

Fig. 1: System model.

destination $d_{i,j}$ at time $D_n^{i,j}$. Accordingly, we always have $0 \leq S_1^{i,j} \leq S_2^{i,j} \leq \ldots$ and $S_n^{i,j} \leq A_n^{i,j} \leq D_n^{i,j}$. We consider in our paper the following class of synchronized packet generation and arrival processes.

Definition 4. *Intra-class Synchronized Sampling and Arrivals*: The packet generation and arrival times are said to be synchronized across streams within each class, if for all classes $i = 1, \ldots, I$, there exist two sequences $\{S_1^i, S_2^i, \ldots\}$ and $\{A_1^i, A_2^i, \ldots\}$ such that for all $n = 1, 2, \ldots$ and $j = 1, \ldots, J_i$

$$S_n^{i,j} = S_n^i, \quad A_n^{i,j} = A_n^i. \tag{5}$$

Note that we let each class have its unique traffic pattern as we do not impose *inter-class* synchronization. In practice, the synchronization between streams within each class can take place when these streams are synchronized by the same clock, e.g., in monitoring and control applications [32], [33]. An example of such scenario is a vehicular network where safety-related data (e.g., position and velocity) are generated every T time units, while other data of lower priority can have different traffic pattern (e.g., updates on the traffic are generated every T' time units) [25]. Also, when $J_i = 1$ for a certain class i, the synchronization assumption within class i reduces to arbitrary packet generation and arrival processes for the aforementioned class. It is worth mentioning that our work is not restricted to any traffic arrival distribution, and can include arbitrary arrival processes where packets may arrive out of order of their generation times. In the sequel, we let

$$\mathscr{I} = \{(S_n^i, A_n^i),\ i = 1, \ldots, I, \quad n = 1, 2, \ldots\} \tag{6}$$

denote the sequence of generation/arrival times for all the classes of the system. We suppose that $\mathscr{I}$ is independent of the service times of the packets and is not altered by the choice of the scheduling policy.

Let π represent a scheduling policy that determines the packets being sent over time. Let Π denote the set of all *causal* scheduling policies, i.e., where the decisions are taken without any knowledge of the future. A policy is said to be work-conserving if the service facility is kept busy whenever there exist one or more unserved packet in the queues. We let Π_{wc} denote the set of work-conserving causal policies. A policy is said to be preemptive if it allows the service facility to switch to transmitting another packet at any time.

C. Age Penalty Functions and Functionals

We define the instantaneous age of information of stream (i, j) at time instant t as:

$$\Delta^{i,j}(t) = t - \max\{S_n^{i,j} : D_n^{i,j} \leq t,\ n = 1, 2, \ldots\}, \tag{7}$$

which is the difference between the current time t and the generation time of the freshest packet that has been delivered to the destination $d_{i,j}$. We let $\boldsymbol{\Delta}^i(t) = (\Delta^{i,1}(t), \ldots, \Delta^{i,J_i}(t))$ denote the age vector at time t of all streams belonging to class i. Additionally, we let $\boldsymbol{\Delta}(t) = (\boldsymbol{\Delta}^1(t), \ldots, \boldsymbol{\Delta}^I(t))$ denote the age vector of all streams at time t.

We introduce an age penalty function $p_t \circ \boldsymbol{\Delta}^i(t)$ that represents the level of dissatisfaction with the aged information at time t for class i, where $p_t : \mathbb{R}^{J_i} \mapsto \mathbb{R}$ is a non-decreasing function of $\boldsymbol{\Delta}^i(t)$. Some commonly used age penalty functions are listed below.

- The sum age of the J_i streams:

$$p_{\text{sum}} \circ \boldsymbol{\Delta}^i(t) = \sum_{j=1}^{J_i} \Delta^{i,j}(t). \tag{8}$$

- The maximum age of the J_i streams:

$$p_{\text{max}} \circ \boldsymbol{\Delta}^i(t) = \max_{j=1,\ldots,J_i} \Delta^{i,j}(t). \tag{9}$$

- The average age threshold violation of the J_i streams:

$$p_{\text{exceed}-\alpha} \circ \boldsymbol{\Delta}^i(t) = \frac{1}{J_i} \sum_{j=1}^{J_i} \mathbb{1}_{\{\Delta^{i,j}(t) > \alpha\}}. \tag{10}$$

  where $\mathbb{1}_{\{.\}}$ is the indicator function, and α is a fixed age threshold that should not be violated.
- The sum age penalty function of the J_i streams:

$$p_{\text{pen}} \circ \boldsymbol{\Delta}^i(t) = \sum_{j=1}^{J_i} g(\Delta^{i,j}(t)), \tag{11}$$

  where $g : \mathbb{R}^+ \mapsto \mathbb{R}$ is a non-decreasing function. For instance, an exponential function $g(\Delta^{i,j}) = \exp(a\Delta^{i,j})$ with $a > 0$ can be used for control applications where the system is vulnerable to outdated information and the need for fresh information grows quickly with respect to the age [11].

We focus in our paper on the family of symmetric and non-decreasing penalty functions:

$$\mathscr{P}_{\text{sym}} = \{p : [0,\infty)^N \mapsto \mathbb{R} \text{ is symmetric and non-decreasing}\}.$$

This class of penalty functions $\mathscr{P}_{\text{sym}}$ is fairly large, and include the provided age penalty functions (8)-(11). Furthermore, we point out that p_t can change over time, which represents the time-variant importance of the information streams. This highlights the generality of our considered penalty functions.

In addition to age penalty functions, we use non-decreasing functionals $\phi(\{p_t \circ \boldsymbol{\Delta}^i(t), t \geq 0\})$ of the age penalty process $\{p_t \circ \boldsymbol{\Delta}^i(t), t \geq 0\}$ to represent the level of dissatisfaction with the aged information of class i, which we dub as the age penalty functionals. Some examples of these functionals are listed below.

- The time-average age penalty:

$$\phi_{\text{avg}}(\{p_t \circ \boldsymbol{\Delta}^i(t), t \geq 0\}) = \frac{1}{T}\int_0^T p_t \circ \boldsymbol{\Delta}^i(t)dt. \quad (12)$$

- The average peak age penalty:

$$\phi_{\text{peak}}(\{p_t \circ \boldsymbol{\Delta}^i(t), t \geq 0\}) = \frac{1}{K}\sum_{k=1}^{K} A_k, \quad (13)$$

  where A_k denotes the k-th peak value of $p_t \circ \boldsymbol{\Delta}^i(t)$ since time $t = 0$. In particular, when class i has only one stream and $p_t \circ \Delta^i(t) = \Delta^i(t)$, (13) reduces to the widely used average peak age metric [10], [30].

We consider in our paper that different priority classes can have distinct age penalty functions and functionals. This is because each priority class typically represents a different application, each with its data timeliness requirements. For example, in a vehicular network, time-crucial safety data related to vehicle position should be delivered promptly. Typically, the system performance is affected by the peak of the maximum age of the delivered updates. Accordingly, we can choose the maximum age penalty function p_{max} and the peak age penalty ϕ_{peak} for this class of traffic. On the other hand, updates on gas tanks levels require an average timely delivery and, consequently, we can choose the penalty function p_{sum} and the time-average age penalty functional ϕ_{avg} for this class.

In the sequel, we use $\{\boldsymbol{\Delta}^i_\pi(t), t \geq 0\}$ and $\{p_t \circ \boldsymbol{\Delta}^i_\pi(t), t \geq 0\}$ to represent the stochastic age process and penalty process of class i respectively when policy π is adopted. We assume that the initial age $\boldsymbol{\Delta}_\pi(0^-)$ at time $t = 0^-$ is the same for all $\pi \in \Pi$.

III. Multi-class Multi-stream Scheduling

A. *Lexicographic Optimality for Age Minimization*

In the sequel, we will introduce the notion of Lexicographic optimality for the age minimization framework, which we will refer to as the lex-age-optimality. As it has been previously detailed in the introduction section, the recent efforts on age analysis in multi-class environments focused mainly on finding closed-form expressions of the average AoI/PAoI of the multi-class system for a particular policy [28]–[30]. Note that, in the multi-class system case, the minimization of the total average AoI/PAoI falls short in capturing the differences in time-cruciality between the classes. In fact, this approach lets each stream contribute equally to the penalty of the system regardless of its class. In another line of work, the weighted sum AoI has been optimized where the weight factors reflect the priority of each stream [18]. However, this approach does not enforce a hard service priority between classes since low priority streams can still be granted the channel even when higher priority packets are present in the system. As will be seen in the following, the lex-age-optimality fully captures this aspect and provides a new direction of age analysis in multi-class scheduling environments.

Definition 5. *Lex-age-optimality*: A scheduling policy $P \in \Pi$ is said to be level 1 lex-age-optimal within Π if for all $\mathscr{I}$, $p_t \in \mathscr{P}_{\text{sym}}$ and $\pi \in \Pi$

$$[\{p_t \circ \boldsymbol{\Delta}^1_P(t), t \geq 0\}|\mathscr{I}] \leq_{st} [\{p_t \circ \boldsymbol{\Delta}^1_\pi(t), t \geq 0\}|\mathscr{I}]. \quad (14)$$

We let $\Pi^1_{\text{lex-opt}} \subseteq \Pi$ denote the set of scheduling policies that are level 1 lex-age-optimal. In addition, P is said to be level k lex-age-optimal for $k = 2, \dots, I$ if it is level $k-1$ lex-age-optimal, and for all $\mathscr{I}$, $p_t \in \mathscr{P}_{\text{sym}}$ and $\pi \in \Pi^{k-1}_{\text{lex-opt}}$

$$[\{p_t \circ \boldsymbol{\Delta}^k_P(t), t \geq 0\}|\mathscr{I}] \leq_{st} [\{p_t \circ \boldsymbol{\Delta}^k_\pi(t), t \geq 0\}|\mathscr{I}], \quad (15)$$

where $\Pi^{k-1}_{\text{lex-opt}}$ is the set of scheduling policies that are level $k-1$ lex-age-optimal. If policy P is level k lex-age-optimal simultaneously for all $k = 1, \dots, I$, it is said to be lex-age-optimal.

According to (4), (14) can be equivalently expressed as

$$\begin{aligned}&\mathbb{E}[\phi(\{p_t \circ \boldsymbol{\Delta}^1_P(t), t \geq 0\})|\mathscr{I}] \\ &= \min_{\pi \in \Pi} \mathbb{E}[\phi(\{p_t \circ \boldsymbol{\Delta}^1_\pi(t), t \geq 0\})|\mathscr{I}],\end{aligned} \quad (16)$$

for all $\mathscr{I}$, $p_t \in \mathscr{P}_{\text{sym}}$, and non-decreasing functional $\phi : \mathbb{V} \mapsto \mathbb{R}$, provided that the expectations in (16) exist. Similarly, an equivalent formulation of the level k lex-age-optimality (15) of a policy $P \in \Pi^{k-1}_{\text{lex-opt}}$ is

$$\begin{aligned}&\mathbb{E}[\phi(\{p_t \circ \boldsymbol{\Delta}^k_P(t), t \geq 0\})|\mathscr{I}] \\ &= \min_{\pi \in \Pi^{k-1}_{\text{lex-opt}}} \mathbb{E}[\phi(\{p_t \circ \boldsymbol{\Delta}^k_\pi(t), t \geq 0\})|\mathscr{I}],\end{aligned} \quad (17)$$

for all $\mathscr{I}$, $p_t \in \mathscr{P}_{\text{sym}}$, and non-decreasing functional $\phi : \mathbb{V} \mapsto \mathbb{R}$, provided that the expectations in (17) exist.

The goal of the lex-age-optimality is to guarantee the age-optimality of high priority classes, and optimize the age performance of the low priority classes accordingly. To see how this is achieved, we recall from (16) that a level 1 lex-age-optimal policy P achieves the smallest possible expected value of any non-decreasing functional ϕ of the stochastic age penalty process $[\{p_t \circ \boldsymbol{\Delta}^1(t), t \geq 0\})|\mathscr{I}]$ among all causal policies. Next, to maintain the age-optimality of the highest priority class, our attention is restricted to the set of scheduling policies that are level 1 lex-age-optimal. We have denoted this set by $\Pi^1_{\text{lex-opt}}$. To that end, and as seen in (17), a policy P is level 2 lex-age-optimal if it achieves the smallest possible expected value of any non-decreasing functional ϕ of the stochastic age penalty process $[\{p_t \circ \boldsymbol{\Delta}^2(t), t \geq 0\})|\mathscr{I}]$ among all level 1 lex-age-optimal policies. This showcases how the lex-age-optimality captures the time-cruciality of streams since, by definition, lex-age-optimal policies grant high priority streams the best possible performance without being influenced by low priority streams. Then, while ensuring the age-optimality of the high priority streams, the performance of the low priority streams is optimized.

B. Lex-Age-Optimal Policy for Exponential Service Time

We consider the case where the service time of each packet is exponentially distributed with service rate μ. To address this multi-stream online scheduling problem, we first lay out the notion of informative packets.

Definition 6. *Informative and Non-informative Packets*: Consider a packet of stream (i,j) that is generated at time $S_n^{i,j} \leq t$. The packet is said to be informative (non-informative) at time t if $t - S_n^{i,j} < \Delta^{i,j}(t)$ ($t - S_n^{i,j} \geq \Delta^{i,j}(t)$), i.e., the age of the packet is (not) smaller than $\Delta^{i,j}(t)$.

Equipped with the above definition, we consider in the following several scheduling disciplines that are based on informative packets.

Definition 7. *Preemptive Priority (PP) policy based on Informative Packets*: Among the streams with informative packets, the class of streams with the highest priority are served first.

Definition 8. *Maximum Age First (MAF) policy*: Among the streams from a priority class, the stream with the maximum age is served first, with ties broken arbitrarily.

Definition 9. *Last-Generated, First-Served (LGFS) policy*: Among the informative packets from a stream, the last generated informative packet is served first, with ties broken arbitrarily.

By combining the above three service disciplines, we propose a new scheduling policy called Preemptive Priority, Maximum Age First, Last-Generated, First-Served (**PP-MAF-LGFS**), which is defined as follows.

Definition 10. *Preemptive Priority, Maximum Age First, Last-Generated, First-Served*: This policy is preemptive, work-conserving and obeys the following set of scheduling rules:

- If there exist informative packets, the system will serve an informative packet that is selected as follows
  - among all streams with informative packets, pick the class of streams with the highest priority;
  - among the streams from the selected priority class, pick the stream with the maximum age, with ties broken arbitrarily;
  - among the informative packets from the selected stream, pick the last generated informative packet, with ties broken arbitrarily;
- if there exists no informative packet, the system can serve any non-informative packet.

Note that our proposed policy does not drop non-informative packets as it was previously proposed in the literature (e.g., [10]). Although these packets are not necessary for reducing the age, but in many applications, they may still be needed at the monitor (e.g., social updates). In the case of a single priority class (i.e., $I = 1$), the proposed policy coincides with the Maximum Age First, Last-Generated, First-Served (**MAF-LGFS**) policy proposed in [17].

By definition, our policy ensures that the service of high priority informative packets is not interrupted nor influenced by any lower priority packets. This grants crucial timely packets the best possible service by the facility. Note that informative packets play a key role in our policy. In particular, the preemptive priority discipline is a dynamic priority rule based on the existence of informative packets: If a stream from class 1 has informative packets, the stream has the highest priority; otherwise, if the stream does not have any informative packets, the stream has the lowest priority, even lower than the streams in class I that have informative packets. This non-trivial aspect of our policy ensures that low priority classes are provided with the best possible opportunity for transmission while not affecting the age of the high priority streams. On another note, our policy guarantees that the highest possible reduction in age from the selected priority class takes place at each packet delivery. These key observations are crucial and will be used to establish the lex-age-optimality of the PP-MAF-LGFS policy.

Theorem 1 (Lex-age-optimality of PP-MAF-LGFS)**.** *If (i) the packet generation and arrival times are synchronized across streams within each class, and (ii) the packet service times are exponentially distributed and i.i.d. across streams and time, then the policy* PP-MAF-LGFS *is lex-age-optimal.*

Proof: This theorem is proven using an inductive sample-path comparison. Specifically, we show by induction that the set of scheduling rules that the PP-MAF-LGFS policy satisfies are sufficient and necessary for level k lex-age-optimality for $k = 1, \dots, I$. Contrary to previous sample-path proofs in the literature (e.g., Theorem 1 in [17]), showing these scheduling rules are sufficient for optimality is not enough in our case. Specifically, at each induction step, a characterization of the exact behavior of each policy $\pi \in \Pi^k_{\text{lex-opt}}$ for $k = 1, \dots, I$ is required. This poses several technical difficulties compared to Theorem 1 in [17], which we solve in our sample-path proof by showing the necessity of the scheduling rules for level k lex-age-optimality for $k = 1, \dots, I$. The details can be found in Appendix A of the technical report [34]. ∎

Note that when each priority class has only one stream, the intra-class synchronization assumption is always satisfied and

Theorem 1 holds for *arbitrarily given* packet generation and arrival times. This special case is of particular interest.

To the best of our knowledge, this is the first lex-age-optimality results for multi-class status updates. Our results are strong as our optimality is established in terms of stochastic ordering of stochastic processes for all symmetric non-decreasing penalty functions, and for all non-decreasing age penalty functionals. In addition, the priority classes can have different traffic patterns, age penalty functions, and age penalty functionals. As it was previously explained in Section II-C, priority classes typically represent different applications, each with their own traffic arrivals and data timeliness requirements. For example, in a certain scenario, we can be interested in minimizing the peak max-age for class 1, the time-average sum-age for class 2, and the peak sum-age for class 3. Theorem 1 guarantees that our proposed policy achieves the required data timeliness goal for any of these cases, despite the differences in age penalty functions and functionals between the classes.

IV. Numerical Results

We consider a vehicle in a V2X (Vehicle-To-Everything) network that sends packets to either nearby vehicles or road-side units (see [25], [35] for two surveys). In the aforementioned surveys, a list of possible packets use cases are presented, each of which having different priorities in the network. We consider 3 data categories in our simulations:

1) **Road Safety Data**: These are the data primarily employed to reduce the number of traffic accidents. These packets are generated periodically with a minimum frequency of $10\,\mathrm{Hz}$. We assume in our settings that the packets' generation frequency is set to $10\,\mathrm{Hz}$. This class of streams has the highest priority among all data types. We consider in our simulations that two streams belong to this class (e.g., the vehicle's position and speed).
2) **Traffic Management Data**: The goal of these data is to optimize the traffic stream and reduce the travel time in the network. We consider in our simulations that one stream belongs to this class (e.g., updates concerning the destination of the vehicle). The generation frequency of these packets is set to $1\,\mathrm{Hz}$. The priority of this class is second to the road safety class.
3) **Convenience and Entertainment Data**: The data in this class are considered to be the least crucial as their aim is to provide entertainment and convenience solely for improving the quality of travel. We consider in our simulations that two streams belong to this class and we suppose that the generation frequency of their packets is $5\,\mathrm{Hz}$.

Based on the above, we can conclude that the arrival rate to our considered system is $\lambda_{tot} = 31$ packets per second. The service facility of the vehicle is supposed to be constituted of 1 server with the transmission times being i.i.d. across streams and time. Moreover, the transmission times are considered to be exponentially distributed with service rate μ.

We compare our proposed policy to the preemptive MAF-LGFS[1] policy proposed in [17]. The preemptive MAF-LGFS policy schedules the packet of the stream with the highest age, regardless of the class it belongs to. As for the age penalty function and functional for each class, we choose $p_{\text{exceed}-\alpha}$ and ϕ_{avg} as the age penalty function and functional for class 1 respectively, where α is set to $250\,\mathrm{ms}$. By doing so, we get

$$\begin{aligned}&\mathbb{E}[\phi_{\text{avg}}(\{p_{\text{exceed}-\alpha} \circ \mathbf{\Delta}^1(t), t \geq 0\})]\\ =&\frac{1}{2}\sum_{j=1}^{2}\frac{1}{T}\int_0^T \Pr(\Delta^{1,j}(t) > \alpha)dt, \end{aligned} \tag{18}$$

where $\Pr(\Delta^{1,j}(t) > \alpha)$ is the probability of violation of the maximum tolerated age $250\,\mathrm{ms}$ by stream $(1, j)$ at time t. The interest in this time-average age penalty function is that in vehicular networks, small age for the velocity and position data can be tolerated but, after a certain value, the performance of the system starts deteriorating due to this aging. For class 2, we choose p_{sum} and ϕ_{peak} as the age penalty function and functional, respectively. In other words, we are interested in minimizing the average peak-age of class 2. Lastly, we choose p_{sum} and ϕ_{avg} for class 3. We iterate over a range of the service rate μ and we run the simulations for $10^5\,\mathrm{s}$. We report in Fig. 2-4 the simulations results that showcase the performance of each policy. We can conclude from these results the following:

- As seen in Fig. 2, our proposed policy always outperforms the preemptive MAF-LGFS policy for class 1 at any service rate. Specifically, the probability of the age threshold violation by the preemptive MAF-LGFS policy is 3 times higher than the one achieved by our policy. This is a consequence of our proposed policy's goal as it gives priority to minimizing the time-average age penalty of class 1 regardless of the other remaining classes.

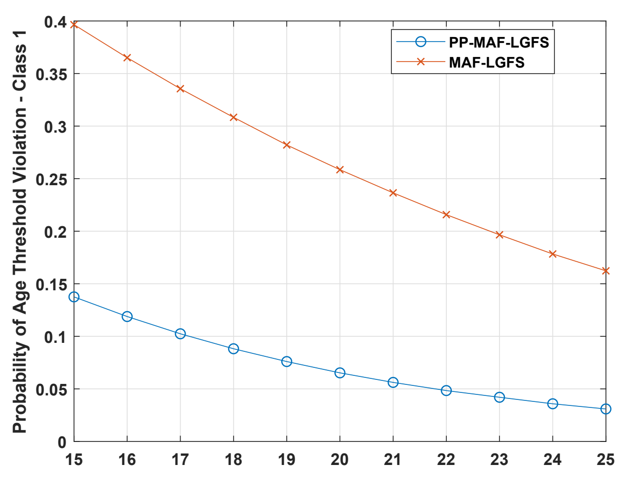

Fig. 2: Probability of age violation for class 1.

- On the other hand, we can see in Fig. 3-4 that the preemptive MAF-LGFS policy outperforms our proposed

[1]First-Come-First-Served (FCFS) policies are omitted from our simulations as they will always be outperformed by LGFS policies since queuing will lead to unnecessary staleness of the packets.

policy for classes 2 and 3. In fact, in our policy, giving the priority to class 1 leads to a penalty for the remainder of the classes. However, we recall that the probability of violation of the age threshold in class 1 for our policy is 3 times less than the preemptive MAF-LGFS. Accordingly, the penalty incurred by the remaining classes is justified. Moreover, we can see that as μ increases, the gap between the two curves in both figures shrinks. The reason behind this is that class 1's packets finish transmission much faster the higher μ is. Consequently, in our proposed policy, the server will be able to finish serving class 1 fast enough that it can start serving the other classes before new packets for class 1 arrive to the system. This reduces the incurred penalty by the low priority classes due to the presence of the high priority streams.

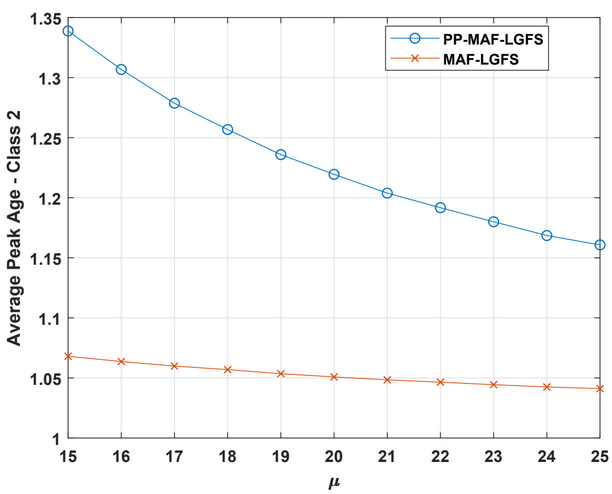

Fig. 3: Average peak age of class 2.

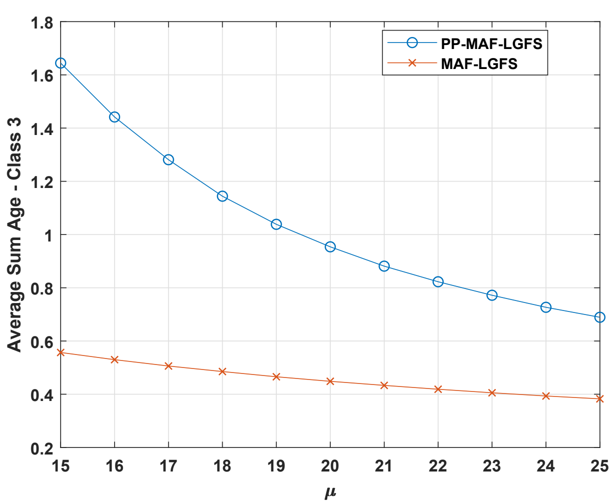

Fig. 4: Average sum age of class 3.

The above results highlight the performance of our proposed lex-age-optimal policy, and provide a new direction on age analysis in multi-class scheduling scenarios.

V. Conclusion and Future Work

In this paper, we have introduced the notion of lex-age-optimality that captures both the age-optimality and the order of time-cruciality between the streams in a general multi-class priority-based scheduling scenario. To that end, we have proposed an online scheduling policy in a general multi-class, multi-stream scheduling scenario. Using a sample-path argument, we were able to prove the lex-age-optimality of the proposed policy in the single exponential server case for any symmetric non-decreasing penalty function, and for all non-decreasing age penalty functionals. Numerical results were then presented to highlight the performance of our proposed policy. Moving forward, the next natural research direction is to relax the exponential service time assumption and establish lex-age-optimal policies for more general service time distributions, such as the New-Better-than-Used (**NBU**) distributions which were considered in [2], [5], [17], [36]. Other directions for future research also include the examination of asynchronized intra-class packet arrivals, multi-server service facility, and multi-hop networks.

References

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

[2] Y. Sun, I. Kadota, R. Talak, and E. Modiano, "Age of information: A new metric for information freshness," *Synthesis Lectures on Communication Networks*, vol. 12, no. 2, pp. 1–224, 2019.

[3] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2017, pp. 486–493.

[4] S. Farazi, A. G. Klein, J. A. McNeill, and D. R. Brown, "On the age of information in multi-source multi-hop wireless status update networks," in *IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2018, pp. 1–5.

[5] A. M. Bedewy, Y. Sun, and N. B. Shroff, "The age of information in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1248–1257, June 2019.

[6] P. Zou, O. Ozel, and S. Subramaniam, "Waiting before Serving: A Companion to Packet Management in Status Update Systems," *arXiv e-prints*, p. arXiv:1901.02873, Jan 2019.

[7] B. Buyukates and S. Ulukus, "Timely Distributed Computation with Stragglers," *arXiv e-prints*, p. arXiv:1910.03564, Oct 2019.

[8] A. Maatouk, M. Assaad, and A. Ephremides, "The age of updates in a simple relay network," in *IEEE Information Theory Workshop (ITW)*, Nov 2018, pp. 1–5.

[9] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age of information performance of multiaccess strategies with packet management," *Journal of Communications and Networks*, vol. 21, no. 3, pp. 244–255, June 2019.

[10] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *IEEE International Symposium on Information Theory*, June 2014, pp. 1583–1587.

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

[12] J. Zhong, R. D. Yates, and E. Soljanin, "Two freshness metrics for local cache refresh," in *IEEE International Symposium on Information Theory (ISIT)*, June 2018, pp. 1924–1928.

[13] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 534–556, Jan 2020.

[14] Y. Sun, Y. Polyanskiy, and E. Uysal, "Sampling of the wiener process for remote estimation over a channel with random delay," *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 1118–1135, 2020.

[15] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The Age of Incorrect Information: A New Performance Metric for Status Updates," *arXiv e-prints*, p. arXiv:1907.06604, Jul 2019.

[16] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Towards an effective age of information: Remote estimation of a markov source," in *IEEE INFOCOM Age of Information Workshop*, April 2018, pp. 367–372.

[17] Y. Sun, E. Uysal-Biyikoglu, and S. Kompella, "Age-optimal updates of multiple information flows," in *IEEE INFOCOM Age of Information Workshop*, 2018, pp. 136–141.

[18] 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.

[19] A. Maatouk, M. Assaad, and A. Ephremides, "Minimizing the age of information: Noma or oma?" in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2019, pp. 102–108.

[20] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "On The Optimality of The Whittle's Index Policy For Minimizing The Age of Information," *arXiv e-prints*, p. arXiv:2001.03096, Jan 2020.

[21] B. Zhou and W. Saad, "Minimum age of information in the internet of things with non-uniform status packet sizes," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2019.

[22] Y. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[23] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, "Timely Status Update in Massive IoT Systems: Decentralized Scheduling for Wireless Uplinks," *arXiv e-prints*, p. arXiv:1801.03975, Jan. 2018.

[24] A. Maatouk, M. Assaad, and A. Ephremides, "Minimizing The Age of Information in a CSMA Environment," *arXiv e-prints*, p. arXiv:1901.00481, Jan 2019, to appear in WiOpt 2019.

[25] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 584–616, Fourth 2011.

[26] J. Walraevens, B. Steyaert, M. Moeneclaey, and H. Bruneel, "Delay Analysis of a HOL Priority Queue," *Telecommun. Syst.*, vol. 30, no. 13, p. 8198, Nov. 2005.

[27] B. Hajek and P. Seri, "Lex-optimal online multiclass scheduling with hard deadlines," *Mathematics of Operations Research*, vol. 30, no. 3, pp. 562–596, 2005.

[28] S. K. Kaul and R. D. Yates, "Age of information: Updates with priority," in *2018 IEEE International Symposium on Information Theory (ISIT)*, June 2018, pp. 2644–2648.

[29] A. Maatouk, M. Assaad, and A. Ephremides, "Age of information with prioritized streams: When to buffer preempted packets?" in *2019 IEEE International Symposium on Information Theory (ISIT)*, July 2019, pp. 325–329.

[30] J. Xu and N. Gautam, "Towards Assigning Priorities in Queues Using Age of Information," *arXiv e-prints*, p. arXiv:1906.12278, Jun 2019.

[31] M. Shaked and J. G. Shanthikumar, Eds., *Stochastic Orders*. Springer New York, 2007.

[32] A. G. Phadke, B. Pickett, M. Adamiak, M. Begovic, G. Benmouyal, R. O. Burnett, T. W. Cease, J. Goossens, D. J. Hansen, M. Kezunovic, L. L. Mankoff, P. G. McLaren, G. Michel, R. J. Murphy, J. Nordstrom, M. S. Sachdev, H. S. Smith, J. S. Thorp, M. Trotignon, T. C. Wang, and M. A. Xavier, "Synchronized sampling and phasor measurements for relaying and control," *IEEE Transactions on Power Delivery*, vol. 9, no. 1, pp. 442–452, Jan 1994.

[33] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July 2004.

[34] A. Maatouk, Y. Sun, A. Ephremides, and M. Assaad, "Status Updates with Priorities: Lexicographic Optimality," *arXiv e-prints*, p. arXiv:2002.01916, Feb. 2020.

[35] H. Vahdat-Nejad, A. Ramazani, T. Mohammadi, and W. Mansoor, "A survey on context-aware vehicular network applications," *Vehicular Communications*, vol. 3, pp. 43 – 57, 2016.

[36] 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, 2019.