Age-Optimal Updates of Multiple Information Flows

Yin Sun\textsuperscript{1}, Elif Uysal-Biyikoglu\textsuperscript{2}, and Sastry Kompella*  
\textsuperscript{1}Dept. of ECE, Auburn University, Auburn, AL  
\textsuperscript{2}Dept. of EEE, Middle East Technical University, Ankara, Turkey  
*Information Technology Division, Naval Research Laboratory, Washington, DC

Abstract—In this paper, we study an age of information minimization problem, where multiple flows of update packets are sent over multiple servers to their destinations. Two online scheduling policies are proposed. When the packet generation and arrival times are synchronized across the flows, the proposed policies are shown to be (near) optimal for minimizing any time-dependent, symmetric, and non-decreasing penalty function of the ages of the flows over time in a stochastic ordering sense.

I. INTRODUCTION

In many information-update and networked control systems, such as news updates, stock trading, autonomous driving, and robotics control, information has the greatest value when it is fresh. A metric on information freshness, called the age of information or simply age, was defined in [1], [2]. Consider a flow of update packets that are sent from a source to a destination through a queue. Let $U(t)$ be the time stamp (i.e., generation time) of the newest update that the destination has received by time $t$. The age of information, as a function of time $t$, is defined as $\Delta(t) = t - U(t)$, which is the time elapsed since the newest update was generated.

In recent years, there have been a lot of research efforts on how to reduce the age $\Delta(t)$ and keep the information fresh, e.g., [2]–[15]. When there is one flow of update packets, a Last Generated First Served (LGFS) update transmission policy has been shown to be (near) optimal for minimizing the age process $\{\Delta(t), t \geq 0\}$ in a stochastic ordering sense for multi-server and multi-hop networks [5]–[8]; these results hold for arbitrary packet generation times at the source and arbitrary packet arrival times at the queue, and also hold for minimizing any non-decreasing age penalty functionals $p(\{\Delta(t), t \geq 0\})$. These studies motivated us to consider whether age optimality can be also achieved in more general systems with multiple flows of update packets. In that case, the transmission scheduler needs to compare not only the packets from the same flow, but also the packets from different flows, which makes the scheduling problem more challenging.

In this paper, we study age-optimal online scheduling in multi-flow, multi-server queueing systems (as illustrated in Figure 1), where each server can be used to send update packets to any destination, one packet at a time. The contributions of this paper are summarized as follows:

- Let $\Delta(t)$ denote the age vector of multiple flows. We introduce an age penalty function $p_t(\Delta(t))$ to represent the level of dissatisfaction for having aged information at the destinations at time $t$, where $p_t$ can be any time-dependent, symmetric, and non-decreasing function of the age vector $\Delta(t)$.

- For single-server systems with i.i.d. exponential service times, we propose a Maximum Age First, Last Generated First Served (MAF-LGFS) policy. If the packet generation and arrival times are synchronized across the flows, then for all age penalty functions $p_t$ defined above, the preemptive MAF-LGFS policy is proven to minimize the age penalty process $\{p_t(\Delta(t)), t \geq 0\}$ among all causal policies in a stochastic ordering sense.

- For multi-server systems with i.i.d. New-Better-than-Used (NBU) service times, age-optimal multi-flow online scheduling is quite difficult to achieve. In this case, we consider an age lower bound called the Age of Service Information and propose a Maximum Age of Service Information First, Last Generated First Served (MASIF-LGFS) policy. For synchronized packet generations and arrivals, the non-preemptive MASIF-LGFS policy is shown to be within an additive gap from the optimum for minimizing the long-run average age of the flows, where the gap is equal to the mean service time of one packet. Numerical evaluations are provided to verify our (near) age optimality results.

A comparison with related work is presented in Section IV. Our results can be potentially applied to: (i) cloud-hosted Web services where the servers in Figure 1 represent a pool of threads (each for a TCP connection) connecting a front-end proxy node to clients [18], (ii) industrial robotics and factory automation systems where multiple sensor-output flows are sent to a wireless AP and then forwarded to a system monitor and/or controller [19], and (iii) Multi-access Edge Computing (MEC) that can process fresh data (e.g., data for video analytics, location services, and IoT) locally at the very edge of the mobile network [20].

\footnote{In practice, synchronized update generations and arrivals occur when there is a single source and multiple destinations, or in periodic sampling where multiple sources are synchronized by the same clock as in many monitoring and control applications, e.g., [16], [17].}
II. System Model

A. Notation and Definitions

We use lower case letters such as $x$ and $x$, respectively, to represent deterministic scalars and vectors. In the vector case, a subscript will index the components of a vector, such as $x_i$. We use $x[i]$ to denote the $i$-th largest component of vector $x$. Let $0$ denote the vector with all $0$ components. A function $f: \mathbb{R}^n \to \mathbb{R}$ is termed symmetric if $f(x) = f(x_1, \ldots, x_n)$ for all $x$. A function $f: \mathbb{R}^n \to \mathbb{R}$ is termed separable if there exists functions $f_1, \ldots, f_n$ of one variable such that $f(x) = \sum_{i=1}^n f_i(x_i)$. The composition of functions $f$ and $g$ is denoted by $f \circ g(x) = f(g(x))$. For any $n$-dimensional vectors $x$ and $y$, the elementwise vector ordering $x_i \leq y_i$, $i = 1, \ldots, n$, is denoted by $x \leq y$. Let $A$ and $\mathcal{U}$ denote sets and events. For all random variable $X$ and event $A$, let $[X]A$ denote a random variable with the conditional distribution of $X$ given $A$.

Definition 1. Stochastic Ordering of Random Variables [27]: A random variable $X$ is said to be stochastically smaller than another random variable $Y$, denoted by $X \leq_{st} Y$, if

$$\Pr(X > t) \leq \Pr(Y > t), \ \forall \ t \in \mathbb{R}. \ \ \ \ \ \ (1)$$

Definition 2. Stochastic Ordering of Random Vectors [21]: A set $\mathcal{U} \subseteq \mathbb{R}^n$ is called upper, if $y \in \mathcal{U}$ whenever $y \geq x$ and $x \in \mathcal{U}$. Let $X$ and $Y$ be two $n$-dimensional random vectors, $X$ is said to be stochastically smaller than $Y$, denoted by $X \leq_{st} Y$, if

$$\Pr(X \in \mathcal{U}) \leq \Pr(Y \in \mathcal{U}), \ \ \forall \ \mathcal{U} \subseteq \mathbb{R}^n. \ \ \ \ \ \ (2)$$

Definition 3. Stochastic Ordering of Stochastic Processes [27]: Let $\{X(t), t \in [0, \infty)\}$ and $\{Y(t), t \in [0, \infty)\}$ be two stochastic processes, $\{X(t), t \in [0, \infty)\}$ is said to be stochastically smaller than $\{Y(t), t \in [0, \infty)\}$, denoted by $\{X(t), t \in [0, \infty)\} \leq_{st} \{Y(t), t \in [0, \infty)\}$, if for all integer $n$ and $0 \leq t_1 < t_2 < \ldots < t_n$, it holds that

$$(X(t_1), X(t_2), \ldots, X(t_n)) \leq_{st} (Y(t_1), Y(t_2), \ldots, Y(t_n)). \ \ \ \ \ \ (3)$$

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

$$\mathcal{V} = \{f : [0, \infty) \to \mathbb{R} \text{ is Lebesgue measurable}\}. \ \ \ \ \ \ (4)$$

A functional $\phi: \mathcal{V} \to \mathbb{R}$ is said to be non-decreasing if $\phi(f_1) \leq \phi(f_2)$ for all $f_1, f_2 \in \mathcal{V}$ satisfying $f_1(t) \leq f_2(t)$ for $t \in [0, \infty)$. We remark that $\{X(t), \ t \in [0, \infty)\} \leq_{st} \{Y(t), t \in [0, \infty)\}$ if and only if, when $\phi \in \mathcal{V}$.

$$\mathbb{E} [\phi(\{X(t), t \in [0, \infty)\})] \leq \mathbb{E} [\phi(\{Y(t), t \in [0, \infty)\})]. \ \ \ \ (5)$$

holds for all non-decreasing functional $\phi: \mathcal{V} \to \mathbb{R}$, provided that the expectations in (2) exist.

B. Queuing System Model

Consider a status update system illustrated in Fig. [1] where $N$ flows of update packets are sent through a queue with $M$ servers and an infinite buffer. Let $s_n$ and $d_n$ denote the source and destination nodes of flow $n$, respectively. Different flows can have different source and/or destination nodes. Each packet can be assigned to any server and a server can only process one packet at a time. The service times of the update packets are i.i.d. across the servers and time.

The system starts to operate at time $t = 0$. The $i$-th update packet of flow $n$ is generated at the source node $s_n$ at time $S_{n,i}$, arrives at the queue at time $A_{n,i}$, and is delivered to the destination $d_n$ at time $D_n,i$ such that $0 \leq S_{n,1} \leq S_{n,2} \leq \ldots$ and $S_{n,i} \leq A_{n,i} \leq D_{n,i}$. We consider the following class of synchronized packet generation and arrival processes:

Definition 4. Synchronized Sampling and Arrivals: The packet generation and arrival times are said to be synchronized across the $N$ flows, if there exists two sequences $\{S_1, S_2, \ldots\}$ and $\{A_1, A_2, \ldots\}$ such that $S_{n,i} = S_i$ and $A_{n,i} = A_i$ for all $i = 1, 2, \ldots$ and $n = 1, 2, \ldots, N$.

Note that in this paper, the sequences $\{S_1, S_2, \ldots\}$ and $\{A_1, A_2, \ldots\}$ are arbitrary. Hence, out-of-order arrivals, e.g., $S_i < S_{i+1}$ but $A_i > A_{i+1}$, are allowed. In addition, when there is a single flow ($N = 1$), synchronized sampling and arrivals reduce to arbitrary packet generation and arrival processes that were considered in [3–8].

Let $\pi$ represent a scheduling policy that determines the packet being sent by the servers over time. Let $\Pi$ denote the set of causal policies in which the scheduling decisions are made based on the history and current states of the system. A policy is said to be preemptive, if each server can switch to send another packet at any time; the preempted packet will be stored back to the queue, waiting to be sent at a later time. A policy is said to be non-preemptive, if each server must complete sending the current packet before starting to serve another packet. A policy is said to be work-conserving, if all servers are kept busy whenever the queue is non-empty. We use $\Pi_{ap}$ to denote the set of non-preemptive causal policies such that $\Pi_{ap} \subset \Pi$. Let

$$\mathcal{I} = \{S_i, A_i, \ i = 1, 2, \ldots\} \ \ \ \ (6)$$

denote the packet generation and arrival times of the flows. We assume that the packet generation/arrival times $\mathcal{I}$ and the packet service times are determined by two mutually independent external processes, both of which do not change according to the adopted scheduling policy.

C. Age Metrics

At any time $t \geq 0$, the freshest packet delivered to the destination node $d_n$ is generated at time $U_n(t)$.

$$U_n(t) = \max\{S_{n,i} : D_{n,i} \leq t, i = 1, 2, \ldots\}. \ \ \ \ (7)$$

The age of information, or simply the age, of flow $n$ is defined as $[1, 2]$

$$\Delta_n(t) = t - U_n(t), \ \ \ \ (8)$$

which is the time difference between the current time $t$ and the generation time of the freshest packet currently available at destination $d_n$. Let $\Delta(t) = (\Delta_1(t), \ldots, \Delta_N(t))$ denote the age vector of the $N$ flows at time $t$. 

2
We introduce an *age penalty function* \( p(\Delta) = p \circ \Delta \) to represent the level of dissatisfaction for having aged information at the \( N \) destinations, where \( p : \mathbb{R}^N \to \mathbb{R} \) can be any non-decreasing function of the \( N \)-dimensional age vector \( \Delta \). Some examples of the age penalty function are:

1. The average age of the \( N \) flows is
   \[
   p_{\text{avg}}(\Delta) = \frac{1}{N} \sum_{n=1}^{N} \Delta_n. \tag{6}
   \]

2. The maximum age of the \( N \) flows is
   \[
   p_{\text{max}}(\Delta) = \max_{n=1}^{N} \Delta_n. \tag{7}
   \]

3. The mean square age of the \( N \) flows is
   \[
   p_{\text{ms}}(\Delta) = \frac{1}{N} \sum_{n=1}^{N} (\Delta_n)^2. \tag{8}
   \]

4. The \( l \)-norm of the age vector of the \( N \) flows is
   \[
   p_{\text{sym}}(\Delta) = \left[ \sum_{n=1}^{N} (\Delta_n)^l \right]^\frac{1}{l}, \quad l \geq 1. \tag{9}
   \]

5. The sum age penalty function of the \( N \) flows is
   \[
   p_{\text{sum}}(\Delta) = \sum_{n=1}^{N} g(\Delta_n), \tag{10}
   \]

where \( g : [0, \infty) \to \mathbb{R} \) is the age penalty function for each flow, which can be any non-decreasing function of the age \( \Delta \) of the flow [3], [4]. For example, a stair-shape function \( g_1(\Delta) = \lfloor a\Delta \rfloor \) with \( a \geq 0 \) can be used to characterize the dissatisfaction of data staleness when the information of interests is checked periodically, and an exponential function \( g_2(\Delta) = e^{a\Delta} \) is appropriate for online learning and control applications where the desire for information refreshing grows quickly with respect to the age [4].

In this paper, we consider a class of symmetric and non-decreasing age penalty functions, i.e.,
\[
\mathcal{P}_{\text{sym}} = \{ p : [0, \infty)^N \to \mathbb{R} \text{ is symmetric and non-decreasing} \}.
\]
This is a fairly large class of age penalty functions, where the function \( p \) can be discontinuous, non-convex, or non-separable. It is easy to see
\[
\{ p_{\text{avg}}, p_{\text{max}}, p_{\text{ms}}, p_{\text{sym}}, p_{\text{sum}} \} \subset \mathcal{P}_{\text{sym}}.
\]
Note that the age vector \( \Delta \) is a function of time \( t \) and policy \( \pi \), and the age penalty function \( p \) can change over time. We use \( \{ p_t \circ \Delta_\pi(t), t \in [0, \infty) \} \) to represent the stochastic process generated by the time-dependent age penalty function \( p_t \) in policy \( \pi \). We assume that the initial age \( \Delta_\pi(0^-) \) at time \( t = 0^- \) remains the same for all \( \pi \in \Pi \).

### III. Multi-Flow Update Scheduling

In this section, we investigate update scheduling of multiple information flows. We first consider a system setting with a single server and exponential service times, where an age optimality result is established. Next, we study a more general system setting with multiple servers and NBU service times. In this case, age optimality is inherently difficult to achieve and we present a near age-optimal result.

#### A. Multiple Flows, Single Server, Exponential Service Times

To address the multi-flow online scheduling problem, we consider a flow selection discipline called Maximum Age First (MAF) [12], [13], [22], in which the flow with the maximum age is served the first, with ties broken arbitrarily. A scheduling policy is defined by combining the MAF and LGFS disciplines as follows:

**Definition 5.** Maximum Age First, Last Generated First Served (MAF-LGFS) policy: In this policy, the last generated packet from the flow with the maximum age is served the first among all packets from all flows, with ties broken arbitrarily.

The age optimality of the preemptive MAF-LGFS policy is established in the following theorem.

**Theorem 1.** If (i) there is a single server \( (M = 1) \), (ii) the packet generation and arrival times are synchronized across the \( N \) flows, and (iii) the packet service times are exponentially distributed and i.i.d. across time, then it holds that for all \( \mathcal{I} \), all \( p_t \in \mathcal{P}_{\text{sym}} \), and all \( \pi \in \Pi \)
\[
\{ p_t \circ \Delta_{\text{prmp}, \text{MAF-LGFS}}(t), t \geq 0 \} \mid \mathcal{I}
\]
\[
\leq \min_{\pi \in \Pi} \mathbb{E} \{ \phi(p_t \circ \Delta_{\pi}(t), t \geq 0) \mid \mathcal{I} \}, \tag{11}
\]

or equivalently, for all \( \mathcal{I} \), all \( p_t \in \mathcal{P}_{\text{sym}} \), and all non-decreasing functional \( \phi : \mathcal{V} \to \mathbb{R} \)
\[
\mathbb{E} \{ \phi(p_t \circ \Delta_{\text{prmp}, \text{MAF-LGFS}}(t), t \geq 0) \mid \mathcal{I} \}
\]
\[
= \min_{\pi \in \Pi} \mathbb{E} \{ \phi(p_t \circ \Delta_{\pi}(t), t \geq 0) \mid \mathcal{I} \}, \tag{12}
\]

provided that the expectations in (12) exist, where \( \mathcal{V} \) is the set of Lebesgue measurable functions defined in (1).

**Proof idea.** If the packet generation and arrival times are synchronized across the flows, the preemptive MAF-LGFS policy satisfies the following property: When a packet is delivered to its destination, the flow with the maximum age before the packet delivery will have the minimum age among the \( N \) flows once the packet is delivered. This is a key idea used in the proof. See Appendix [A] for the details.

Theorem [1] tells us that, for all age penalty functions in \( \mathcal{P}_{\text{sym}} \), all number of flows \( N \), and all synchronized packet generation and arrival times \( \mathcal{I} \), the preemptive MAF-LGFS policy minimizes the stochastic process \( \{ p_t \circ \Delta_{\pi}(t), t \geq 0 \} \) among all causal policies in a stochastic ordering sense.

#### B. Multiple Flows, Multiple Servers, NBU Service Times

Next, we consider a more general system setting with multiple servers and a class of New-Better-than-Used (NBU)

\[\text{NBU Service Times}\]

- In the special case that there is a single flow \( (N = 1) \), the MAF-LGFS policy reduces to the LGFS policy studied in [5]–[8].

\[\text{NBU Service Times}\]

- Note that this property does not hold when packet generations and arrivals are asynchronized.
Fig. 3. The age of served information $\Xi_n(t)$ as a lower bound of age $\Delta_n(t)$. service time distributions that include exponential distribution as a special case.

**Definition 6.** New-Better-than-Used Distributions: Consider a non-negative random variable $X$ with complementary cumulative distribution function (CCDF) $F(x) = \Pr[X > x]$. Then, $X$ is said to be New-Better-than-Used (NBU) if for all $t$, $\tau \geq 0$

$$F(\tau + t) \leq F(\tau)F(t).$$

Examples of NBU distributions include constant service time, exponential distribution, shifted exponential distribution, Erlang distribution, negative binomial distribution, etc.

In the scheduling literature, optimal online scheduling has been successfully established for delay minimization in single-server queueing systems, e.g., [13, 24], which can become inherently difficult in the multi-server cases [25–27]. Similarly, age-optimal online scheduling is quite challenging in multi-flow, multi-server systems. In the sequel, we consider a slightly relaxed goal to seek for near age-optimal online scheduling of multiple information flows, where our proposed scheduling policy is shown to be within a small additive gap from the optimum age performance.

Notice that the age $\Delta_n(t)$ in (5) is determined by the packets that have been delivered to the destination $d_n$ by time $t$. To establish near age optimality, we consider an alternative age metric call the Age of Served Information, which is determined by the packets that have started service by time $t$: Let $V_{n,i}$ denote the time that the $i$-th packet of flow $n$ is assigned to a server, i.e., the service starting time of the $i$-th packet of flow $n$, which is shown in Fig. [2]. By definition, one can get $S_{n,i} \leq A_{n,i} \leq V_{n,i} \leq D_{n,i}$. The Age of Served Information of flow $n$ is defined as

$$\Xi_n(t) = t - \max\{S_{n,i} : V_{n,i} \leq t, i = 1, 2, \ldots\},$$

which is the time difference between the current time $t$ and the generation time of the freshest packet that has started service by time $t$. As shown in Fig. [3] $\Xi_n(t) \leq \Delta_n(t)$. Let $\Xi(t) = (\Xi_1(t), \ldots, \Xi_N(t))$ denote the age of served information vector at time $t$.

We propose a new scheduling discipline called Maximum Age of Served Information First (MASIF), in which the flow with the maximum age of served information is served the first, with ties broken arbitrarily. Using this discipline, we define the following scheduling policy:

**Definition 7.** Maximum Age of Served Information First, Last Generated First Served (MASIF-LGFS) policy: In this policy, the last generated packet from the flow with the maximum age of served information is served the first among all packets from all flows, with ties broken arbitrarily.

Next, we will show that the age of service information of the non-preemptive MASIF-LGFS policy provides a uniform age lower bound for all non-preemptive and causal policies.

**Theorem 2.** If (i) the packet generation and arrival times are synchronized across the $N$ flows and (ii) the packet service times are NBU and $i.i.d.$ across the servers and time, then it holds that for all $T$, all $p_t \in P_{sym}$, and all $\pi \in \Pi_{sym}$

$$[\{p_t \circ \Xi_{\text{non-premp, MASIF-LGFS}}(t), t \geq 0\}] = \Xi_n([\{p_t \circ \Delta_{\pi}(t), t \geq 0\}],$$

or equivalently, for all $T$, all $p_t \in P_{sym}$, and all non-decreasing functional $\phi : \mathbb{R} \rightarrow \mathbb{R}$

$$\mathbb{E}[\phi(\{p_t \circ \Xi_{\text{non-premp, MASIF-LGFS}}(t), t \geq 0\})] \leq \min_{\pi \in \Pi_{sym}} \mathbb{E}[\phi(\{p_t \circ \Delta_{\pi}(t), t \geq 0\})],$$

provided that the expectations in (16) exist.

**Proof idea.** Under synchronized packet generations and arrivals, the non-preemptive MASIF-LGFS policy satisfies: When a packet starts service, the flow with the maximum age of served information before the service starts will have the minimum age of served information among the $N$ flows once the service starts. Theorem 2 is proven by using this idea and the sample-path method developed in [28, 29]. See Appendix B for the details.

Hence, the non-preemptive MASIF-LGFS policy is near age-optimal in the sense of (16). In particularly, for the average age of the $N$ flows in (6) (i.e., $p_t = p_{avg}$), we can obtain

**Theorem 3.** Under the conditions of Theorem 2 it holds that for all $T$

$$\min_{\pi \in \Pi_{sym}} [\Delta_{\pi}(t)] \leq [\Delta_{\text{non-premp, MASIF-LGFS}}(t)] \leq \min_{\pi \in \Pi_{sym}} [\Delta_{\pi}(t)] + \mathbb{E}[X],$$

where $\Delta_{\pi} = \limsup_{T \rightarrow \infty} \frac{1}{T} \mathbb{E}[\int_0^T p_{avg} \circ \Delta_{\pi}(t) dt]$, $\mathbb{E}[X]$ denotes the expected time-average of the average age of the $N$ flows, and $\mathbb{E}[X]$ denotes the mean service time of one packet.
The proof of Theorem 3 is similar to that of Theorem 4 in [6] and hence is omitted here. By Theorem 3, the average age of the non-preemptive MASIF-LGFS policy is within an additive gap from the optimum, and the gap $E[X]$ is invariant of the packet arrival and generation times $T$, the number of flows $N$, and the number of servers $M$. Finally, we note that, similar to the results in [28], [29], Theorem 2 and Theorem 3 can be generalized to the case that the servers have different NBU service time distributions.

IV. RELATED WORK

The age performance of multiple sources has been analyzed in [9]–[11]. Status updates over a multiaccess channel was studied in [14], an age minimization problem for single-hop wireless networks with interference constraints was shown to be NP hard, and tractable cases were identified. In [12], the expected time-average of the weighted sum age of multiple sources was minimized in a broadcast network with an ON-OFF channel and periodic arrivals, where only one source is scheduled at a time and the scheduler does not know the current ON-OFF channel state. When the network is symmetric and the weights are equal, a sample-path method was used to show that the maximum age first (MAF) policy is optimal. Further, a sub-optimal Whittle’s index method was used to handle the general asymmetric cases. In [13], for symmetric Bernoulli arrivals and an always-ON channel with no buffers, the MAF policy was shown to be optimal for minimizing the expected time-average of the sum age of multiple sources. In addition, Markov decision process (MDP) methods were used to handle the general scenarios with asymmetric arrivals and a buffer, where the optimal policies are shown to be switch-type. Theorem 1 is similar to the symmetric age optimality results in [12], [13], with an extension to general time-dependent, symmetric, and non-decreasing age penalty functions $p_t$. In addition, Theorem 2 takes one more step to consider multi-flow, multi-server scheduling. This paper also complements the studies in [5]–[8] on (near) age-optimal online scheduling with a single information flow.

V. NUMERICAL RESULTS

In this section, we evaluate the age performance of several multi-flow online scheduling policies. These scheduling policies are defined by combining the flow selection disciplines {MAF, RAND, MAF} and the packet selection disciplines {FCFS, LGFS}, where RAND represents randomly choosing a flow among the flows with un-served packets. The packet generation times $S_i$ follow a Poisson process with rate $\lambda$, and the time difference $(A_i - S_i)$ between packet generation and arrival is equal to either 0 or $4/\lambda$ with equal probability. The mean service time of each server is set as $E[X] = 1/\mu = 1$. Hence, the traffic intensity is $\rho = \lambda N/M$, where $N$ is the number of flows and $M$ is the number of servers.

Figure 4 illustrates the expected time-average of the maximum age $p_{\text{max}}(\Delta(t))$ of 10 flows in a system with a single server and $i.i.d.$ exponential service times. One can see that the preemptive MAF-LGFS policy has the best age performance, and both the RAND and FCFS disciplines lead to performance loss. Note that there is no need for preemptions under the FCFS discipline. Figure 5 plots the expected time-average of the average age $p_{\text{avg}}(\Delta(t))$ of 30 flows in a system with 3 servers and $i.i.d.$ NBU service times. In particular, the service time $X$ follows the following shifted exponential distribution:

$$\Pr[X > x] = \begin{cases} 1, & \text{if } x < \frac{1}{3}, \\ \exp[-\frac{3}{2}(x - \frac{1}{3})], & \text{if } x \geq \frac{1}{3}. \end{cases}$$ (17)

One can observe that the non-preemptive MASIF-LGFS policy is better than the other policies, and is quite close to the age lower bound where the gap from the lower bound is no more than the mean service time $E[X] = 1$. These numerical results are in accordance with our theoretical analysis in Section III.

VI. CONCLUSION

We have developed online scheduling policies and shown they are (near) optimal for minimizing the age of information in multi-flow, multi-server systems. Similar with [6], the results in this paper can be generalized to consider packet replications over multiple servers. Other future research directions include asynchronized packet arrivals, packet transmissions with errors, and multi-flow updates in multi-hop networks.

REFERENCES

[1] X. Song and J. W. S. Liu, “Performance of multiversion concurrency control algorithms in maintaining temporal consistency,” in Fourteenth Annual International Computer Software and Applications Conference, Oct 1990, pp. 132–139.
A. G. Phadke, B. Pickett, M. Adamiak, and et. al., “Synchronized QoS management,” in *Proc. IEEE INFOCOM Mini Conference*, 2012, pp. 2731-2735.

Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” in *IEEE INFOCOM*, 2016.

---, “Update or wait: How to keep your data fresh,” *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

A. M. Bedewy, Y. Sun, and N. B. Shroff, “Optimizing data freshness, throughput, and delay in multi-server information-update systems,” in *IEEE ISIT*, 2016.

---, “Minimizing the age of the information through queues,” submitted to *IEEE Trans. Inf. Theory*, 2017, http://arxiv.org/abs/1709.04956.

Y. C. He, D. Yuan, and A. Ephremides, “Optimal link scheduling for age of information updates in multihop networks,” in *IEEE ISIT*, 2017.

---, “The age of information in multihop networks,” submitted to *IEEE Trans. Inf. Theory*, 2017, https://arxiv.org/abs/1712.10061.

R. D. Yates and S. K. Kaul, “Real-time status updating: Multiple sources,” in *IEEE ISIT*, July 2012, pp. 2666–2670.

---, “The age of information: Real-time status updating by multiple sources,” submitted to *IEEE Trans. Inf. Theory*, 2016, http://arxiv.org/abs/1608.08622.

Y. C. He, D. Yuan, and A. Ephremides, “Optimal link scheduling for age minimization in wireless systems,” *IEEE Trans. Inf. Theory*, in press, 2018.

R. D. Yates and S. K. Kaul, “Status updates over unreliable multiaccess channels,” in *IEEE ISIT*, 2017, pp. 331–335.

A. G. Phadke, B. Pickett, M. Adamiak, and et. al., “Synchronized sampling and phaser measurements for relaying and control,” *IEEE Transactions on Power Delivery*, vol. 9, no. 1, pp. 442–452, Jan 1994.

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

A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, and P. Gauthier, “Cluster-based scalable network services,” *SIGOPS Oper. Syst. Rev.*, vol. 31, no. 5, pp. 78–91, Oct. 1997.

V. C. Gungor and G. P. Hancke, “Industrial wireless sensor networks: Challenges, design principles, and technical approaches,” *IEEE Transactions on Industrial Informatics*, vol. 56, no. 10, pp. 4258–4265, Oct 2009.

Nokia, https://networks.nokia.com/solutions/multi-access-edge-computing.

M. Shahid and J. G. Shanthikumar, *Stochastic Orders*. Springer, 2007.

B. Li, A. Eryilmaz, and R. Srikant, “On the universality of age-based scheduling in wireless networks,” in *IEEE INFOCOM*, April 2015, pp. 1302–1310.

L. Schrage, “A proof of the optimality of the shortest remaining processing time discipline,” *Operations Research*, vol. 16, pp. 687–690, 1968.

J. R. Jackson, “Scheduling a production line to minimize maximum tardiness,” management Science Research Report, University of California, Los Angeles, CA, 1955.

S. Leonardi and D. Raz, “Approximating total flow time on parallel machines,” in *ACM STOC*, 1997.

G. Weiss, “Turnpike optimality of smith’s rule in parallel machines stochastic scheduling,” *Math. Oper. Res.*, vol. 17, no. 2, pp. 255–270, May 1992.

---, “On almost optimal priority rules for preemptive scheduling of stochastic jobs on parallel machines,” *Advances in Applied Probability*, vol. 27, no. 3, pp. 821-839, 1995.

Y. Sun, C. E. Koksal, and N. B. Shroff, “On delay-optimal scheduling in queuing systems with replications,” 2016, https://arxiv.org/abs/1603.07322.

M. Shahid, “Near delay-optimal scheduling of batch jobs in multi-server systems,” http://www.auburn.edu/~yxs0078/parallel_servers.pdf, 2017.

L. Kleinrock, *Queueing Systems*. John Wiley and Sons, 1975, vol. 1:Theory.

J. Nino-Mora, “Conservation laws and related applications,” in *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc., 2010.

J. C. Gittins, K. Glazebrook, and R. Weber, *Multi-armed Bandit Allocation Indices*, 2nd ed. Wiley, Chichester, NY, 2011.
time \( t \), because no packets that has arrived is generated later than \( W(t) \), we can obtain

\[
\Delta_{i,p_i} \geq \Delta'_{i,P_1} \geq t - W(t), \quad i = 1, \ldots, N; \\
\Delta_{i,p_i} \geq \Delta'_{i,\pi_i} \geq t - W(t), \quad i = 1, \ldots, N. \tag{20}
\]

Because there is only one server and policy \( P_1 \) follows the same scheduling discipline with the preemptive MAF-LGFS policy, each delivered packet in policy \( P_1 \) must be from the flow with the maximum age \( \Delta_{[i],P_1} \) (denoted as flow \( n^* \)), and the delivery packet must be the last generated packet that is time-stamped with \( W(t) \). In other words, the age of flow \( n^* \) is reduced from the maximum age \( \Delta_{[i],P_1} \) to the minimum age \( \Delta'_{[i],P_1} = t - W(t) \), and the ages of the other \( (N - 1) \) flows remain unchanged. Hence,

\[
\Delta'_{[i],P_1} = \Delta_{[i+1],P_1}, \quad i = 1, \ldots, N - 1, \tag{21}
\]

\[
\Delta'_{[N],P_1} = t - W(t). \tag{22}
\]

In policy \( \pi_1 \), the delivered packet can be any packet from any flow. For all possible cases of policy \( \pi_1 \), it must hold that

\[
\Delta'_{[i],\pi_1} \geq \Delta_{[i+1],\pi_1}, \quad i = 1, \ldots, N - 1. \tag{23}
\]

By combining (18), (21), and (23), we have

\[
\Delta'_{[i],\pi_1} \geq \Delta_{[i+1],\pi_1} \geq \Delta'_{[i+1],P_1} = \Delta'_{[i],P_1}, \quad i = 1, \ldots, N - 1.
\]

In addition, combining (20) and (22), yields

\[
\Delta'_{[N],\pi_1} \geq t - W(t) = \Delta'_{[N],P_1}.
\]

By this, (19) is proven. \( \square \)

Now we are ready to prove Theorem 1

**Proof of Theorem 1** Consider any work-conserving policy \( \pi \in \Pi \). By Lemma 1 there exist policy \( P_1 \) and policy \( \pi_1 \) satisfying the same scheduling disciplines with policy \( P \) and policy \( \pi \), respectively, and the packet delivery times in policy \( P_1 \) and policy \( \pi_1 \) are synchronized almost surely.

For any given sample path of policy \( P_1 \) and policy \( \pi_1 \), \( \Delta_{P_1}(0^-) = \Delta_{\pi_1}(0^-) \) at time \( t = 0^- \). We consider two cases:

**Case 1:** When there is no packet delivery, the age of each flow grows linearly with a slope 1.

**Case 2:** When a packet is delivered, the evolution of the age is governed by Lemma 2.

By induction over time, we obtain

\[
\Delta_{[i],P_1}(t) \leq \Delta_{[i],\pi_1}(t), \quad i = 1, \ldots, N, \quad t \geq 0. \tag{24}
\]

For any symmetric and non-decreasing function \( p_i \), it holds from (24) that for all sample paths and all \( t \geq 0 \)

\[
p_i \circ \Delta_{P_1}(t) = p_i(\Delta_{1,P_1}(t), \ldots, \Delta_{N,P_1}(t)) = p_i(\Delta_{[1],P_1}(t), \ldots, \Delta_{[N],P_1}(t)) \leq p_i(\Delta_{[1],\pi_1}(t), \ldots, \Delta_{[N],\pi_1}(t)) = p_i(\Delta_{1,\pi_1}(t), \ldots, \Delta_{N,\pi_1}(t)) = p_i \circ \Delta_{\pi_1}(t). \tag{25}
\]

By Lemma 1 the state process \( \{\Delta_{P_1}(t), t \geq 0\} \) of policy \( P_1 \) has the same distribution with the state process \( \{\Delta_{P}(t), t \geq 0\} \) of policy \( P \); the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( \pi_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( \pi \). By (23) and Theorem 6.B.30 in [21], (11) holds for all work-conserving policy \( \pi \in \Pi \).

For non-work-conserving policies \( \pi \), because the service times are exponentially distributed and i.i.d. across servers and time, server idling only postpones the delivery times of the packets. One can construct a coupling to show that for any non-work-conserving policy \( \pi \), there exists a work-conserving policy \( \pi' \) whose age process is smaller than that of policy \( \pi \) in stochastic ordering; the details are omitted. As a result, (11) holds for all policies \( \pi \in \Pi \).

Finally, the equivalence between (11) and (12) follows from [2]. This completes the proof. \( \square \)

**APPENDIX B**

**PROOF OF THEOREM 2**

This proof is motivated by the sample-path method developed in [28], [29] for near delay-optimal scheduling in multi-server queueing systems.

We first provide two useful lemmas. Let \( (\Delta_{s}(t), \Xi_{s}(t)) \) denote the system state of policy \( \pi \) at time \( t \) and \( \{(\Delta_{s}(t), \Xi_{s}(t)), t \geq 0\} \) denote the state process of policy \( \pi \). For notational simplicity, let policy \( P \) represent the non-preemptive MASIF-LGFS policy.

In single-server queueing systems, the following work conservation law (or its generalizations) plays an important role in the analysis of scheduling performance: At any time, the expected total amount of time for completing the packets in the queue is invariant among all work-conserving policies [30–32]. However, the work conservation law does not hold in multi-server queueing systems, where it is difficult to fully utilize all the servers to process the packets. Specifically, it may happen that some servers are busy while the remaining servers are idle, where the idleness leads to inefficient packet service and a performance gap from the optimum. In the sequel, we introduce an ordering to compare the efficiency of packet service in different policies in a near-optimal sense, which is called weak work-efficiency ordering [3].

**Definition 8.** Weak Work-efficiency Ordering [28], [29]: For any given \( I \) and a sample path of two policies \( \pi_1, \pi_2 \in \Pi_{NP} \), policy \( \pi_1 \) is said to be weakly more work-efficient than policy \( \pi_2 \), if the following assertion is true: For each packet \( j \) executed in policy \( \pi_2 \), if

1. In policy \( \pi_2 \), packet \( j \) starts service at time \( \tau \) and completes service at time \( \nu \) (\( \tau \leq \nu \)),
2. In policy \( \pi_1 \), the queue is not empty during \([\tau, \nu]\),
then there always exists one corresponding packet \( j' \) in policy \( \pi_1 \) which starts service during \([\tau, \nu]\).

4Two work-efficiency orderings were used in [28], [29] to study (near) delay-optimal online scheduling in multi-server queueing systems.
An sample-path illustration of the weak work-efficiency ordering is provided in Fig. 6. In particular, if policy $\pi_1$ is weakly more work-efficient than policy $\pi_2$, then each packet in policy $\pi_1$ must start service during the service duration $[\tau, \nu]$ of its corresponding packet in policy $\pi_2$, or the queue is empty during $[\tau, \nu]$ in policy $\pi_1$. Note that the weak work-efficient ordering does not require to specify which server is used to serve each packet.

The following coupling lemma was established in [29] by using the property of NBU distributions and the fact that policy $P$ (i.e., the non-preemptive MASIF-LGFS policy) is work-conserving:

**Lemma 3. (Coupling Lemma)** [29] Lemma 2 Consider two policies $P, \pi \in \Pi_{np}$. If (i) policy $P$ is work-conserving and (ii) the packet service times are NBU, independent across the servers, and i.i.d. across the packets assigned to the same server, then there exist policy $P_1$ and policy $\pi_1$ in the same probability space which satisfy the same scheduling disciplines with policy $P$ and policy $\pi$, respectively, such that

1. The state process $\{(\Delta_{P_1}(t), \Xi_{P_1}(t)), t \geq 0\}$ of policy $P_1$ has the same distribution with the state process $\{(\Delta_{P}(t), \Xi_{P}(t)), t \geq 0\}$ of policy $P$,
2. The state process $\{(\Delta_{\pi_1}(t), \Xi_{\pi_1}(t)), t \geq 0\}$ of policy $\pi_1$ has the same distribution with the state process $\{(\Delta_{\pi}(t), \Xi_{\pi}(t)), t \geq 0\}$ of policy $\pi$,
3. Policy $P_1$ is weakly more work-efficient than policy $\pi_1$ with probability one.

The proof of Lemma 3 is provided in [29]. Note that Lemma 3 holds if policy $P$ is replaced by any non-preemptive work-conserving policy.

We will compare the age lower bound of policy $P_1$ and the age of policy $\pi_1$ on a sample path by using the following lemma:

**Lemma 4. (Inductive Comparison)** Under the conditions of Lemma 1 suppose that a packet starts service in policy $P_1$ and a packet completes service (i.e., delivered to the destination) in policy $\pi_1$ at the same time $t$. The state system of policy $P_1$ is $\{(\Delta_{P_1}, \Xi_{P_1})\}$ before the service starts, which becomes $\{(\Delta'_{P_1}, \Xi'_{P_1})\}$ after the service starts. The state system of policy $\pi_1$ is $\{(\Delta_{\pi_1}, \Xi_{\pi_1})\}$ before the service completes, which becomes $\{(\Delta'_{\pi_1}, \Xi'_{\pi_1})\}$ after the service completes. If the packet generation and arrival times are synchronized across the $N$ flows and

$$\Xi_{[i],P_1} \leq \Delta_{[i],P_1}, \ i = 1, \ldots, N,$$  \hspace{1cm} (26)

then

$$\Xi_{[i],\pi_1} \leq \Delta'_{[i],\pi_1}, \ i = 1, \ldots, N.$$  \hspace{1cm} (27)

**Proof.** For synchronized packet generation and arrivals, let $W(t) = \max\{S_i : A_i \leq t\}$ be the time-stamp of the freshest packet of each flow that has arrived to the queue by time $t$. At time $t$, because no packets that has arrived is generated later than $W(t)$, we can obtain

$$\Xi_{i,P_1} \geq \Xi'_{i,P_1} \geq t - W(t), \ i = 1, \ldots, N,$$

$$\Delta_{i,\pi_1} \geq \Delta'_{i,\pi_1} \geq t - W(t), \ i = 1, \ldots, N.$$ \hspace{1cm} (28)

Because there is only one server and policy $P_1$ follows the same scheduling discipline with the non-preemptive MASIF-LGFS policy, each packet starts service in policy $P_1$ must be from the flow with the maximum age of served information $\Xi_{[i],P_1}$ (denoted as flow $n^*$), and the delivery packet must be the last generated packet that is time-stamped with $W(t)$. In other words, the age of served information of flow $n^*$ is reduced from the maximum age of served information $\Xi_{[i],P_1}$ to the minimum age of served information $\Xi_{[N],P_1} = t - W(t)$, and the ages of served information of the other $(N - 1)$ flows remain unchanged. Hence,

$$\Xi_{[i],P_1} = \Xi_{[i+1],P_1}, \ i = 1, \ldots, N - 1,$$  \hspace{1cm} (29)

$$\Xi'_{[N],P_1} = t - W(t).$$  \hspace{1cm} (30)

In policy $\pi_1$, the delivered packet can be any packet from any flow. For all possible cases of policy $\pi_1$, it must hold that

$$\Delta'_{[i],\pi_1} \geq \Delta_{[i+1],\pi_1}, \ i = 1, \ldots, N - 1.$$ \hspace{1cm} (31)

By combining (26), (29), and (31), we have

$$\Delta'_{[i],\pi_1} \geq \Delta_{[i+1],\pi_1} \geq \Xi_{[i+1],P_1} = \Xi'_{[i],P_1}, \ i = 1, \ldots, N - 1.$$ 

In addition, combining (28) and (30), yields

$$\Delta_{[N],\pi_1} \geq t - W(t) = \Xi_{[N],P_1}.$$ 

By this, (27) is proven. \hfill \Box

Now we are ready to prove Theorem 2.

**Proof of Theorem 2.** Consider any policy $\pi \in \Pi_{np}$. By Lemma 3, there exist policy $P_1$ and policy $\pi_1$ satisfying the same scheduling disciplines with policy $P$ and policy $\pi$, respectively, and policy $P_1$ is weakly more work-efficient than policy $\pi_1$ with probability one.

Next, we construct a policy $\pi'_1$ in the same probability space with policy $P_1$ and policy $\pi_1$: Let $\Delta_{\pi'_1}(0^-) = \Xi_{P_1}(0^-) = \Delta_{\pi_1}(0^-)$ at time $t = 0^-$. For each pair of corresponding packet $j$ and packet $j'$ mentioned in the definition of the weak work-efficiency ordering, if

- In policy $\pi_1$, packet $j$ starts service at time $\tau$ and completes service at time $\nu (\tau \leq \nu)$,
- In policy $P_1$, the queue is not empty during $[\tau, \nu]$,
it holds from (32) and (33) that for all sample paths and all services in policy \( P \) in policy \( \pi \), the queue is empty (all packets are delivered or under service) state is governed by Lemma 4.

By Lemma 4, the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( P_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( P \); the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( \pi_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( \pi \). By (34) and Theorem 6.B.30 in [21], (15) holds for all policy \( \pi \in \Pi_{np} \). Finally, the equivalence between (15) and (16) follows from (2). This completes the proof. \( \square \)

By Lemma 3, the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( P_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( P \); the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( \pi_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( \pi \). By (34) and Theorem 6.B.30 in [21], (15) holds for all policy \( \pi \in \Pi_{np} \). Finally, the equivalence between (15) and (16) follows from (2). This completes the proof. \( \square \)

By Lemma 3, the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( P_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( P \); the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( \pi_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( \pi \). By (34) and Theorem 6.B.30 in [21], (15) holds for all policy \( \pi \in \Pi_{np} \). Finally, the equivalence between (15) and (16) follows from (2). This completes the proof. \( \square \)

By Lemma 3, the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( P_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( P \); the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( \pi_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( \pi \). By (34) and Theorem 6.B.30 in [21], (15) holds for all policy \( \pi \in \Pi_{np} \). Finally, the equivalence between (15) and (16) follows from (2). This completes the proof. \( \square \)

By Lemma 3, the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( P_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( P \); the state process \( \{\Delta_{\pi_1}(t), t \geq 0\} \) of policy \( \pi_1 \) has the same distribution with the state process \( \{\Delta_{\pi}(t), t \geq 0\} \) of policy \( \pi \). By (34) and Theorem 6.B.30 in [21], (15) holds for all policy \( \pi \in \Pi_{np} \). Finally, the equivalence between (15) and (16) follows from (2). This completes the proof. \( \square \)