Minimizing the Age of Information through Queues

Ahmed M. Bedewy, Yin Sun, *Member, IEEE*, and Ness B. Shroff, *Fellow, IEEE*

Abstract—**In this paper, we investigate scheduling policies that minimize the age of information in single-hop queueing systems. We propose a Last-Generated, First-Serve (LGFS) scheduling policy, in which the packet with the earliest generation time is processed with the highest priority. If the service times are *i.i.d.* exponentially distributed, the preemptive LGFS policy is proven to be age-optimal in a stochastic ordering sense. If the service times are *i.i.d.* and satisfy a New-Better-than-Used (NBU) distributional property, the non-preemptive LGFS policy is shown to be within a constant gap from the optimum age performance. These age-optimality results are quite general: (i) They hold for arbitrary packet generation times and arrival times (including out-of-order packet arrivals), (ii) They hold for multi-server packet scheduling with the possibility of replicating a packet over multiple servers, (iii) They hold for minimizing not only the time-average age and mean peak age, but also for minimizing the age stochastic process and any non-decreasing functional of the age stochastic process. If the packet generation time is equal to packet arrival time, the LGFS policies reduce to the Last-Come, First-Serve (LCFS) policies. Hence, the age optimality results of LCFS-type policies are also established.**

I. Introduction

The ubiquity of mobile devices and applications has greatly boosted the demand for real-time information updates, such as news, weather reports, email notifications, stock quotes, social updates, mobile ads, etc. Also, timely status updates are crucial in networked monitoring and control systems. These include, but are not limited to, sensor networks used to measure temperature or other physical phenomena, and surrounding monitoring in autonomous driving.

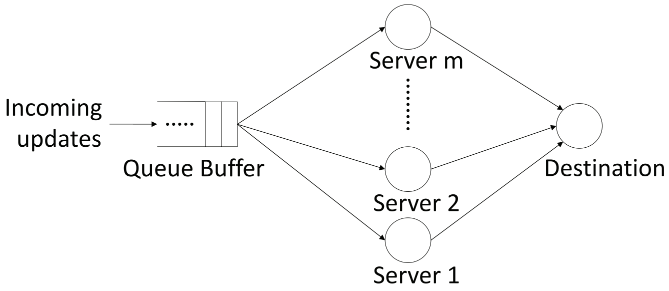


Figure 1: System model.

A common need in these real-time applications is to keep the destination (i.e., information consumer) updated with the freshest information. To identify the timeliness of the updates, a metric called the *age-of-information*, or simply *age*, was defined in, e.g., [2]–[5]. At time t, if the packet with the largest generation time at the destination was generated at time $U(t)$, the age $\Delta(t)$ is defined as

$$\Delta(t) = t - U(t). \quad (1)$$

Hence, age is the time elapsed since the freshest received packet was generated.

In recent years, a variety of approaches have been investigated to reduce the age. In [5]–[7], it was found in First-Come, First-Serve (FCFS) queueing systems that the time-average age first decreases with the update frequency and then increases with the update frequency. The optimal update frequency was obtained to minimize the age in FCFS systems. In [8]–[10], it was shown that the age can be further improved by discarding old packets waiting in the queue when a new sample arrives. Characterizing the age in Last-Come, First-Serve (LCFS) queueing systems with gamma distributed service times was considered in [11]. However, these studies cannot tell us (i) which queueing discipline can minimize the age and (ii) under what conditions the minimum age is achievable.

In this paper, we answer these two questions for an information-update system illustrated in Fig. 1, where a se-

This paper was presented in part at IEEE ISIT 2016 [1].

This work has been supported in part by ONR grants N00014-17-1-2417 and N00014-15-1-2166, Army Research Office grants W911NF-14-1-0368 and MURI W911NF-12-1-0385, National Science Foundation grants CNS-1446582, CNS-1421576, CNS-1518829, and CCF-1813050, and a grant from the Defense Thrust Reduction Agency HDTRA1-14-1-0058.

A. M. Bedewy is with the Department of ECE, The Ohio State University, Columbus, OH 43210 USA (e-mail: bedewy.2@osu.edu).

Y. Sun is with the Department of ECE, Auburn University, Auburn, AL 36849 USA (e-mail: yzs0078@auburn.edu).

N. B. Shroff is with the Department of ECE and the Department of CSE, The Ohio State University, Columbus, OH 43210 USA (e-mail: shroff.11@osu.edu).

quence of update packets arrive at a queue with m servers and a buffer size B. Each server can be used to model a channel in multi-channel communication systems [12], or a computer in parallel computing systems [13]. The service times of the update packets are *i.i.d.* across servers and the packets assigned to the same server. Let s_i be the generation time of the update packet i at an external source, and a_i be the arrival time of the update packet i at the queue. Out-of-order packet arrivals are allowed, such that the packets may arrive in an order different from their generation times, e.g., $s_i < s_j$ but $a_j < a_i$. Packet replication [14]–[16] is considered in this study. In particular, multiple replicas of a packet can be assigned to different servers, at possibly different service starting time epochs. The first completed replica is considered as the valid execution of the packet; after that, the remaining replicas of this packet are cancelled immediately to release the servers. Suppose that a packet can be replicated on at most r servers ($r \leq m$), where r is called the maximum replication degree. If $r = 1$, this reduces to the case where replication is not allowed at all. We propose a Last-Generated, First-Serve (LGFS) scheduling policy, in which the packet with the earliest generation time is served with the highest priority. The following are the key contributions of this paper:

- If the packet service times are *i.i.d.* exponentially distributed, then for *arbitrary* system parameters (including *arbitrary* packet generation times s_i, packet arrival times a_i, number of servers m, maximum replication degree r, and buffer size B), we prove that the preemptive LGFS with replication (prmp-LGFS-R) policy minimizes the age stochastic process and any non-decreasing functional of the age stochastic process among all policies in a stochastic ordering sense (Theorem 6). Note that this age penalty model is very general. Many age penalty metrics studied in the literature, such as the time-average age [5], [6], [8]–[11], [17]–[21], average peak age [7]–[9], [11], [20], [22], and time-average age penalty function [23], [24], are special cases of this age penalty model.
- We further investigate a more general class of packet service time distributions called New-Better-than-Used (NBU) distributions. We show that the non-preemptive Last-Generated, First-Serve with replication (non-prmp-LGFS-R) policy is within a constant age gap from the optimum average age, and that the gap is independent of the system parameters mentioned above (Theorem 12). Note that policy non-prmp-LGFS-R with a maximum replication degree r can be near age-optimal compared with policies with any maximum replication degree. This result was not anticipated: In [16], [25], [26], it was shown that non-replication policies are near delay-optimal and replication policies are far from the optimum delay and throughput performance for NBU service time distributions. From these studies, one would expect that replications may worsen the age performance. To our surprise, however, we found that a replicative policy (i.e., non-prmp-LGFS-R) is near-optimal in minimizing the age, even for NBU service time distributions.
- For a special case of the system settings where the update packets arrive in the same order of their generation times and there is no replication, the prmp-LGFS-R policy reduces to LCFS with preemption in service for a single source case in [17], and the non-prmp-LGFS-R when $B = 1$ reduces to LCFS with preemption only in waiting for a single source case in [17], or the "M/M/1/2*" in [8], [9]. Hence, our optimality results are also established for these LCFS-type policies. This relationship tells us that this policy can achieve age-optimality in this case.
- Finally, we investigate the throughput and delay performance of the proposed policies. We show that if the packet service times are i.i.d. exponentially distributed, then the prmp-LGFS-R policy is also throughput and delay optimal among all policies (Theorem 14). In addition, if the packet service times are *i.i.d.* NBU and replications are not allowed, then the non-prmp-LGFS policy is throughput and delay optimal among all non-preemptive policies (Theorem 15).

To the best of our knowledge, these are the first optimality results on minimizing the age-of-information in queueing systems. Moreover, this is the first paper that considers packet replication to minimize the age.

The remainder of this paper is organized as follows. After a brief overview of related work in Section II, we present the model and problem formulation in Section III. The age of the proposed policies is analyzed in Section IV, and the throughput and delay performance of these policies are investigated in Section V. Finally, we conclude in Section VI.

II. Related Work

A series of works studied the age performance of scheduling policies in a single queueing system with Poisson arrival process and exponential service time [5], [6], [8]–[10], [17], [18]. In [5], [6], the update frequency was optimized to improve data freshness in FCFS information-update systems. The effect of the packet management on the age was considered in [8]–[10]. It was found that a good policy is to discard the old updates waiting in the queue when a new sample arrives, which can greatly reduce the impact of queueing delay on data freshness. In [17], the time-average age was characterized for multiple sources Last-Come, First-Serve (LCFS) information-update systems with and without preemption. In this study, it was shown that sharing service facility among Poisson sources can improve the total age. Characterizing the time average age for FCFS queueing system with two and infinite number of servers was studied in [18]. The analysis in [18] showed that the model with infinite servers has a lower age in conjunction with more wasting of network resources due to the rise in the obsolete delivered packets. One open question in these studies on age analysis [5], [6], [8]–[10], [17], [18] is whether the preemptive LCFS policy is age-optimal for exponential service times. In this paper, we provide a confirmative answer to this question, and further investigate age-optimality for more general system settings such as arbitrary packet generation and arrival processes (including out-of-order packet arrivals), multi-server networks, as well as packet replications over multiple servers.

In [19], the average age was characterized in a pull model, where a customer sends requests to all servers to retrieve (pull) the interested information. In this model, the servers carry information with different freshness level and a user waits for the responses from these servers. The server updating process and the response times were assumed to be Poisson and exponential, respectively. In contrast with [19], where the authors assumed that a user contacts servers to check for updates, here we prove age-optimality in a multi-server queueing system where a user sends the updates to a destination through the servers and packet replication is considered.

Characterizing the age for a class of packet service time distributions that are more general than exponential was considered in [7], [11], [22]. In [7], the age was analyzed in multi-class M/G/1 and M/G/1/1 queues. The age performance in the presence of errors when the service times are exponentially distributed was analyzed in [22]. Gamma-distributed service times was considered in [11]. The studies in [11], [22] were carried out for LCFS queueing systems with and without preemption. In complement with the age analysis results in [7], [11], [22], we show that non-preemptive LGFS (and its special case non-preemptive LCFS) policies are near age-optimal for NBU service time distributions. Similar to the exponential case, these results for NBU service times hold for arbitrary packet general and arrival processes, multiple server networks, and packet replication over multiple servers. In addition, gamma distribution considered in [11], [22] is a special case of NBU service time distributions.

In our study, packet generation and arrival times are not controllable. Another line of research has been the joint optimization of packet generation and transmissions in [20], [21], [23], [24]. An information update policy was developed in [23], [24], which was proven to minimize a general class of non-negative, non-decreasing age penalty functions among all causally feasible policies. More recently, a real-time sampling problem of the Wiener process has been studied in [27]: If the sampling times are independent of the observed Wiener process, the optimal sampling problem in [27] reduces to an age-of-information optimization problem; otherwise, the optimal sampling policy can use knowledge of the Wiener process to achieve better performance than age-of-information optimization.

Recently, we generalized our results to multihop networks in [28], where we proved that age-optimality is achievable in multihop networks with arbitrary packet generation times, packet arrival times, and general network topologies. In particular, it was shown that the LGFS policy is age-optimal among all causal policies for exponential packet service times. In addition, for arbitrary distributions of packet service times, it was shown that the LGFS policy is age-optimal among all non-preemptive work-conserving policies.

The considered age penalty model in this paper is very general such that it includes, but is not limited to, the time-average age [5], [6], [8]–[11], [17]–[21], average peak age [7]–[9], [11], [20], [22], and time-average age penalty function [23], [24].

III. Model and Formulation

A. Notations and Definitions

For any random variable Z and an event A, let $[Z|A]$ denote a random variable with the conditional distribution of Z for given A, and $\mathbb{E}[Z|A]$ denote the conditional expectation of Z for given A.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be two vectors in $\mathbb{R}^n$, then we denote $\mathbf{x} \leq \mathbf{y}$ if $x_i \leq y_i$ for $i =$

$1, 2, \ldots, n$. We use $x_{[i]}$ to denote the i-th largest component of vector $\mathbf{x}$. A set $U \subseteq \mathbb{R}^n$ is called upper if $\mathbf{y} \in U$ whenever $\mathbf{y} \geq \mathbf{x}$ and $\mathbf{x} \in U$. We will need the following definitions:

Definition 1. ***Univariate Stochastic Ordering:*** *[29] Let X and Y be two random variables. Then, X is said to be stochastically smaller than Y (denoted as $X \leq_{st} Y$), if*

$$\mathbb{P}\{X > x\} \leq \mathbb{P}\{Y > x\}, \quad \forall x \in \mathbb{R}.$$

Definition 2. ***Multivariate Stochastic Ordering:*** *[29] Let $\mathbf{X}$ and $\mathbf{Y}$ be two random vectors. Then, $\mathbf{X}$ is said to be stochastically smaller than $\mathbf{Y}$ (denoted as $\mathbf{X} \leq_{st} \mathbf{Y}$), if*

$$\mathbb{P}\{\mathbf{X} \in U\} \leq \mathbb{P}\{\mathbf{Y} \in U\}, \quad \textit{for all upper sets} \quad U \subseteq \mathbb{R}^n.$$

Definition 3. ***Stochastic Ordering of Stochastic Processes:*** *[29] Let $\{X(t), t \in [0, \infty)\}$ and $\{Y(t), t \in [0, \infty)\}$ be two stochastic processes. Then, $\{X(t), t \in [0, \infty)\}$ 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 choices of an integer n and $t_1 < t_2 < \ldots < t_n$ in $[0, \infty)$, it holds that*

$$(X(t_1), X(t_2), \ldots, X(t_n)) \leq_{st} (Y(t_1), Y(t_2), \ldots, Y(t_n)), \quad (2)$$

where the multivariate stochastic ordering in (2) was defined in Definition 2.

B. Preliminary Propositions

The following propositions will be used throughout the paper:

Proposition 1 ([29], Theorem 6.B.3)**.** *Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)$ be two n-dimensional random vectors. If*

$$X_1 \leq_{st} Y_1,$$

$$[X_2 | X_1 = x_1] \leq_{st} [Y_2 | Y_1 = y_1] \textit{ whenever } x_1 \leq y_1,$$

and in general, for $i = 2, 3, \ldots, n$,

$$[X_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}] \leq_{st} [Y_i | Y_1 = y_1, \ldots, Y_{i-1} = y_{i-1}]$$
$$\textit{whenever } x_j \leq y_j, \ j = 1, 2, \ldots, i-1,$$

then $\mathbf{X} \leq_{st} \mathbf{Y}$.

Proposition 2 ([29], Theorem 6.B.16.(a))**.** *Let $\mathbf{X}$ and $\mathbf{Y}$ be two n-dimensional random vectors. If $\mathbf{X} \leq_{st} \mathbf{Y}$ and $\mathbf{q} : \mathbb{R}^n \to \mathbb{R}^k$ is any k-dimensional increasing [decreasing] function, for any positive integer k, then the k-dimensional vectors $\mathbf{q}(\mathbf{X})$ and $\mathbf{q}(\mathbf{Y})$ satisfy $\mathbf{q}(\mathbf{X}) \leq_{st} [\geq_{st}] \mathbf{q}(\mathbf{Y})$.*

Proposition 3 ([29], Theorem 6.B.16.(b))**.** *Let $\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_d$ be a set of independent random vectors where the dimension of $\mathbf{X}_i$ is k_i, $i = 1, 2, \ldots, d$. Let $\mathbf{Y}_1, \mathbf{Y}_2, \ldots \mathbf{Y}_d$ be another set of independent random vectors where the dimension of $\mathbf{Y}_i$ is k_i, $i = 1, 2, \ldots, d$. Denote $k = k_1 + k_2 + \ldots + k_d$. If $\mathbf{X}_i \leq_{st} \mathbf{Y}_i$ for $i = 1, 2, \ldots, d$, then, for any increasing function $\psi : \mathbb{R}^k \to \mathbb{R}$, one has*

$$\psi(\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_d) \leq_{st} \psi(\mathbf{Y}_1, \mathbf{Y}_2, \ldots \mathbf{Y}_d).$$

Proposition 4 ([29], Theorem 6.B.16.(e))**.** *Let $\mathbf{X}, \mathbf{Y}$, and $\mathbf{\Theta}$ be random vectors such that $[\mathbf{X} | \mathbf{\Theta} = \theta] \leq_{st} [\mathbf{Y} | \mathbf{\Theta} = \theta]$ for all θ in the support of $\mathbf{\Theta}$. Then $\mathbf{X} \leq_{st} \mathbf{Y}$.*

In the next proposition, $=_{\text{st}}$ denotes equality in law.

Proposition 5 ([29], Theorem 6.B.30)**.** *The random processes $\{X(t), t \in [0, \infty)\}$ and $\{Y(t), t \in [0, \infty)\}$ satisfy $\{X(t), t \in [0, \infty)\} \leq_{st} \{Y(t), t \in [0, \infty)\}$ if, and only if, there exist two random processes $\{\widetilde{X}(t), t \in [0, \infty)\}$ and $\{\widetilde{Y}(t), t \in [0, \infty)\}$, defined on the same probability space, such that*

$$\{\widetilde{X}(t), t \in [0, \infty)\} =_{st} \{X(t), t \in [0, \infty)\},$$
$$\{\widetilde{Y}(t), t \in [0, \infty)\} =_{st} \{Y(t), t \in [0, \infty)\},$$

and

$$\mathbb{P}\{\widetilde{X}(t) \leq \widetilde{Y}(t), t \in [0, \infty)\} = 1.$$

C. Queueing System Model

We consider a queueing system with m servers as shown in Fig. 1. The system starts to operate at time $t = 0$. The update packets are generated exogenously to the system and then arrive at the queue. Thus, the update packets may not arrive at the queue instantly when they are generated. The i-th update packet, called packet i, is generated at time s_i, arrives at the queue at time a_i, and is delivered to the destination at time c_i such that $0 \leq s_1 \leq s_2 \leq \ldots$ and $s_i \leq a_i \leq c_i$. Note that in this paper, the sequences $\{s_1, s_2, \ldots\}$ and $\{a_1, a_2, \ldots\}$ are *arbitrary*. Hence, the update packets may not arrive at the system in the order of their generation times. For example, in Fig. 2, we have $s_1 < s_2$ but $a_2 < a_1$. Let B denote the buffer size of the queue which can be infinite, finite, or even zero. If B is finite, the packets that arrive to a full buffer are either dropped or replace other packets in the queue. 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 processes. Packet replication is considered in this model, where the maximum replication degree is r ($1 \leq r \leq m$). In this model, one packet can be replicated to at most r servers and the first completed replica is considered as the valid execution of the packet. After that, the remaining replicas of this packet are cancelled immediately to release the servers. Note that, the maximum replication degree r is fixed for a system; however, the number of replicas that can be created for a packet may vary between 1 and r.

D. Scheduling Policy

A scheduling policy, denoted by π, determines the packet assignments and replications over time; it also controls dropping or replacing packets when the queue buffer is full. Note that the packet delivery time to the destination c_i is a function of the scheduling policy π, while the sequences $\{s_1, s_2, \ldots\}$ and $\{a_1, a_2, \ldots\}$ do not change according to the scheduling policy. However, a policy π may have knowledge of the future packet generation and arrival times. Moreover, we assume that the packet service times are invariant of the scheduling policy and the realization of a packet service time is unknown until its service is completed (unless the service time is deterministic).

Define Π_r as the set of all policies, that includes *causal* and *non-causal* policies, when the maximum replication degree is r. Hence, $\Pi_1 \subset \Pi_2 \subset \ldots \subset \Pi_m$. Note that causal policies are those policies whose scheduling decisions are made based only on the history and current state of the system; while non-causal policies are those policies whose scheduling decisions are made based on the history, current, and future state of the system. We define several types of policies in Π_r:

A policy is said to be **preemptive**, if a server can preempt a packet being processed and switch to processing any other (including the preempted packet itself) packet at any time; only one copy of the preempted packet can be stored back into the queue if there is enough buffer space and sent at a later time when the servers are available again [1]. In contrast, in a **non-preemptive** policy, processing of a packet cannot be interrupted until the packet is completed or cancelled [2]; after completing or cancelling a packet, the server can switch to process another packet. A policy is said to be **work-conserving**, if no server is idle whenever there are packets waiting in the queue.

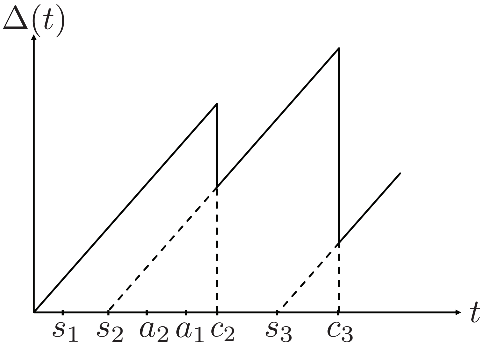


Figure 2: Sample path of the age process $\Delta(t)$.

E. Age Performance Metric

Let $U(t) = \max\{s_i : c_i \leq t\}$ be the largest generation time of the packets at the destination at time t. The *age-of-information*, or simply the *age*, is defined as [2]–[5]

$$\Delta(t) = t - U(t). \tag{3}$$

The initial state $U(0^-)$ at time $t = 0^-$ is invariant of the policy $\pi \in \Pi_r$, where we assume that $s_0 = U(0^-) = 0$. As shown in Fig. 2, the age increases linearly with t but is reset to a smaller value with the arrival of a packet with larger generation time. The age process is given by

$$\Delta = \{\Delta(t), t \in [0, \infty)\}. \tag{4}$$

In this paper, we introduce a non-decreasing *age penalty functional* $g(\Delta)$ to represent the level of dissatisfaction for data staleness at the receiver or destination.

Definition 4. ***Age Penalty Functional:*** *Let* $\mathbf{V}$ *be the set of* n*-dimensional Lebesgue measurable functions, i.e.,*

$$\mathbf{V} = \{f : [0, \infty)^n \mapsto \mathbb{R} \text{ such that } f \text{ is Lebesgue measurable}\}.$$

A functional $g : \mathbf{V} \mapsto \mathbb{R}$ *is said to be an* age penalty functional *if* g *is* non-decreasing *in the following sense:*

$$g(\Delta_1) \leq g(\Delta_2), \text{ whenever } \Delta_1(t) \leq \Delta_2(t), \forall t \in [0, \infty). \tag{5}$$

The age penalty functionals used in prior studies include:

- *Time-average age [5], [6], [8]–[11], [17]–[21]:* The time-average age is defined as

$$g_1(\Delta) = \frac{1}{T} \int_0^T \Delta(t)dt, \tag{6}$$

[1] If a preempted packet is served again, its service either starts over or it resumes the service from the preempted point. In case of exponential service times, both scenarios are equivalent because of the memoryless property.

[2] Recall that a packet is cancelled when a replica has completed processing at another server.

- *Average peak age [7]–[9], [11], [20], [22]:* The average peak is defined as

$$g_2(\Delta) = \frac{1}{K}\sum_{k=1}^{K} A_k, \tag{7}$$

where A_k denotes the k-th peak value of $\Delta(t)$ since time $t = 0$.

- *Time-average age penalty function [23], [24]:* The average age penalty function is

$$g_3(\Delta) = \frac{1}{T}\int_0^T h(\Delta(t))dt, \tag{8}$$

where $h : [0, \infty) \to [0, \infty)$ can be any non-negative and non-decreasing function. As pointed out in [24], a stair-shape function $h(x) = \lfloor x \rfloor$ can be used to characterize the dissatisfaction of data staleness when the information of interest is checked periodically, and an exponential function $h(x) = e^x$ is appropriate for online learning and control applications where the demand for updating data increases quickly with respect to the age. Also, an indicator function $h(x) = \mathbb{1}(x > d)$ can be used to characterize the dissatisfaction when a given age limit d is violated.

IV. Age-Optimality Results of LGFS Policies

In this section, we provide age-optimality and near age-optimality results for multi-server queueing networks with packet replication. We start by considering the exponential packet service time distribution and show that age-optimality can be achieved. Then, we consider the classes of NBU packet service time distributions and show that there exist simple policies that can come close to age-optimality.

A. Exponential Service Time Distribution

We study age-optimal packet scheduling when the packet service times are *i.i.d.* exponentially distributed. We start by defining the Last-Generated, First-Serve discipline as follows.

Definition 5. *A scheduling policy is said to follow the* ***Last-Generated, First-Serve (LGFS)*** *discipline, if the last generated packet is served first among all packets in the system.*

In the LGFS disciplines, packets are served according to their generation times such that the packet with the largest generation time is served first among all packets in the system. In contrast, in the LCFS disciplines, packets are served according to their arrival times such that the packet with the largest arrival time is served first among all packets in the system. Both disciplines are equivalent when the packets arrive to the queue in the same order of their generation times.

In this paper, we propose a policy called **preemptive Last-Generated, First-Serve with replication (prmp-LGFS-R)**. This policy follows the LGFS discipline. When there is no replication ($r = 1$), the implementation details of prmp-LGFS-R policy [3] are depicted in Algorithm 1.

Algorithm 1: Prmp-LGFS-R policy when $r = 1$.

```
$\alpha := 0$; // $\alpha$ is the smallest generation time of the packets under service
$I := m$;                              // $I$ is the number of idle servers
$Q := \emptyset$; // $Q$ is the set of distinct packets that are under service
while the system is ON do
    if a new packet $p_i$ with generation time $s$ arrives then
        if I=0 then                            // All servers are busy
            if $s \leq \alpha$ then                  // Packet $p_i$ is stale
                Store the packet in the queue;
            else           // Packet $p_i$ carries fresh information
                Find packet $p_j \in Q$ with generation time $\alpha$;
                Preempt packet $p_j$ and store it back to the queue;
                Assign packet $p_i$ to the idle server;
                $Q := Q \cup \{p_i\} - \{p_j\}$;
            end
        else                    // At least one of the servers is idle
            Assign packet $p_i$ to an idle server;
            $Q := Q \cup \{p_i\}$;
        end
        Update $I$;
        $\alpha := \min\{s_i : i \in Q\}$;
    end
    if a packet $p_l$ is delivered then
        $Q := Q - \{p_l\}$;
        if the queue is not empty then
            Pick the packet with the largest generation time in the
              queue $p_h$;
            Assign packet $p_h$ to an idle server;
            $Q := Q \cup \{p_h\}$;
        end
        Update $I$;
        $\alpha := \min\{s_i : i \in Q\}$;
    end
end
```

When there is a packet replication ($r > 1$), the prmp-LGFS-R policy acts as follows. We replicate the packet with the largest generation time in the system on r servers. Then, we replicate the packet with the second largest generation time in the system on the remaining idle servers such that the total number of replicas does not exceed r, and so on (i.e., the replicas of the packet with a larger generation time are sent with a higher priority than those of the packet with a lower generation time). In other words, since we may not have $m = ar$ for some positive integer a, packets under service may

[3] The decision related to dropping or replacing packets in the full buffer case does not affect the age performance of prmp-LGFS-R policy. Hence, we don't specify this decision under the prmp-LGFS-R policy in all related algorithms.

not be evenly distributed among the servers if all servers are busy. In this case, we give the highest priority to the k ($k = \lfloor \frac{m}{r} \rfloor$) packets under service with the largest generation times and each one of them is replicated on r servers. The packet under service with the smallest generation time is replicated on the remaining idle servers (whose number is less than r). If $m = ar$ for some positive integer a, then all packets under service are evenly distributed among the servers and each one of them is replicated on r servers. The implementation details of prmp-LGFS-R policy when $r \geq 1$ are depicted in Algorithm 2: This algorithm explains the procedures that the prmp-LGFS-R policy follows in the case of packet arrival and departure events as follows.

- **Packet arrival event:** If a new packet p_i arrives, we first check whether this new packet preempts an older packet that is being processed or not in Steps 6-19. After that, if packet p_i is served, we specify the number of replicas that we need to create for packet p_i in Steps 21-26. In particular, if packet p_i is served, we have two possible cases.
  Case 1: The generation time of packet p_i is greater than the one with the smallest generation time in the set Q (set Q is defined at the beginning of the algorithm). In this case, we need to replicate packet p_i on r idle servers. Therefore, if the number of available servers (I) is less than r, we preempt $(r - I)$ more replicas of the packet with the smallest generation time in the set Q and replicate packet p_i on r servers. These procedures are depicted in Steps 21-23.
  Case 2: The generation time of packet p_i is the smallest one among the packets in the set Q. 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 depicted in Steps 24-26.
- **Packet departure event:** If a packet p_l is delivered, we cancel all the remaining replicas of packet p_l. Moreover, if the queue is not empty, we pick the freshest packet in the queue and replicate it on the available idle servers such that the total number of replicas of this packet does not exceed r. These procedures are illustrated in Steps 29-39.

Algorithm 2: Prmp-LGFS-R policy when $r \geq 1$.

```
1  $\alpha := 0$; // $\alpha$ is the smallest generation time of the packets under service
2  $I := m$;                                   // $I$ is the number of idle servers
3  $Q := \emptyset$;          // $Q$ is the set of distinct packets that are under service
4  $k := \lfloor \frac{m}{r} \rfloor$; // $k$ is the number of distinct packets that each one of them
     can be replicated on $r$ servers
5  while the system is ON do
6      if a new packet $p_i$ with generation time $s$ arrives then
7          if $I = 0$ then                              // All servers are busy
8              if $s \leq \alpha$ then                          // Packet $p_i$ is stale
9                  Store packet $p_i$ in the queue;
10             else                       // Packet $p_i$ carries fresh information
11                 Find packet $p_j \in Q$ with generation time $\alpha$;
12                 Preempt all replicas of packet $p_j$;
13                 Packet $p_j$ is stored back to the queue;
14                 $Q := Q \cup \{p_i\} - \{p_j\}$;
15                 Update $I$;
16             end
17         else                              // At least one of the servers is idle
18             $Q := Q \cup \{p_i\}$;
19         end
20         $\alpha := \min\{s_i : i \in Q\}$;
21         if $p_i \in Q$ and generation time of packet $p_i > \alpha$ and $I < r$
             then            // Specify the number of replicas of packet $p_i$
22             Preempt $(r - I)$ replicas of the packet with generation
                 time $\alpha$;
23             Replicate packet $p_i$ on $r$ idle servers;
24         else if $p_i \in Q$ and generation time of packet $p_i = \alpha$ then
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 generation time in the
                 queue $p_h$;
34             $Q := Q \cup \{p_h\}$;
35             Replicate packet $p_h$ on $\min\{r, I\}$ idle servers;
36             Update $I$;
37         end
38         $\alpha := \min\{s_i : i \in Q\}$;
39     end
40 end
```

Note that the prmp-LGFS-R policy is a causal policy, i.e., its scheduling decisions are made based on the history and current state of the system and do not require the knowledge of the future packet generation and arrival times. Define a set of parameters $\mathcal{I} = \{B, m, r, s_i, a_i, i = 1, 2, \ldots\}$, where B is the queue buffer size, m is the number of servers, r is the maximum replication degree, s_i is the generation time of packet i, and a_i is the arrival time of packet i. Let $\Delta_\pi = \{\Delta_\pi(t), t \in [0, \infty)\}$ be the age processes under policy π. The age performance of the prmp-LGFS-R policy is characterized as follows.

Theorem 6. *Suppose that the packet service times are exponentially distributed, and* i.i.d. *across servers and the packets assigned to the same server, then for all $\mathcal{I}$ and $\pi \in \Pi_r$*

$$[\Delta_{\textit{prmp-LGFS-R}}|\mathcal{I}] \leq_{st} [\Delta_\pi|\mathcal{I}], \tag{9}$$

or equivalently, for all $\mathcal{I}$ and non-decreasing functional g

$$\mathbb{E}[g(\Delta_{\textit{prmp-LGFS-R}})|\mathcal{I}] = \min_{\pi\in\Pi_r} \mathbb{E}[g(\Delta_\pi)|\mathcal{I}], \quad (10)$$

provided the expectations in (10) *exist.*

Proof. See Appendix A. □

Theorem 6 tells us that for arbitrary sequence of packet generation times $(s_1, s_2, \ldots)$, sequence of packet arrival times $(a_1, a_2, \ldots)$, buffer size B, number of servers m, and maximum replication degree r, the prmp-LGFS-R policy achieves optimality of the age process within the policy space Π_r. In addition, (10) tells us that the prmp-LGFS-R policy minimizes any *non-decreasing functional* of the age process, including the time-average age (6), average peak age (7), and time-average age penalty function (8) as special cases. It is important to emphasize that the prmp-LGFS-R policy can achieve optimality compared with all causal and non-causal policies in Π_r. Also, when the update packets arrive in the same order of their generation times and there is no replication, the prmp-LGFS-R policy becomes LCFS with preemption in service (LCFS-S) for a single source case that was proposed in [17]. Thus, this policy can achieve age-optimality in this case.

As a result of Theorem 6, we can deduce the following corollaries:

A weaker version of Theorem 6 can be obtained as follows.

Corollary 7. *If the conditions of Theorem 6 hold, then for any arbitrary packet generation and arrival processes, and for all $\pi \in \Pi_r$*

$$\Delta_{\textit{prmp-LGFS-R}} \leq_{st} \Delta_\pi.$$

Proof. We consider the mixture over multiple sample paths of the packet generation and arrival processes to prove the result. In particular, by using the result of Theorem 6 and Proposition 4, the corollary follows. □

Corollary 8. *Under the conditions of Theorem 6, if one packet can be replicated to all m servers (i.e., $r = m$), then for all $\mathcal{I}$, the prmp-LGFS-R policy when $r = m$ is an age-optimal among all policies in Π_m.*

Proof. This corollary is a direct result of Theorem 6. □

It is important to recall that $\Pi_1 \subset \Pi_2 \subset \ldots \subset \Pi_m$. Therefore, Corollary 8 tells us that the prmp-LGFS-R policy when $r = m$ achieves age-optimality compared with all policies with any maximum replication degree.

Corollary 9. *If the conditions of Theorem 6 hold, then for all $\mathcal{I}$, the age performance of the prmp-LGFS-R policy remains the same for any queue size $B \geq 0$.*

Proof. From the operation of policy prmp-LGFS-R, its queue is used to store the preempted packets and outdated arrived packets. The age process of the prmp-LGFS-R policy is not affected no matter these packets are delivered or not. Hence, the age performance of the prmp-LGFS-R policy is invariant for any queue size $B \geq 0$. This completes the proof. □

The next corollary clarifies the relationship between the prmp-LGFS-R policy and the LCFS-S policy.

Corollary 10. *Under the conditions of Theorem 6, if the packets arrive to the queue in the same order of their generation times and replications are not allowed, then for all $\mathcal{I}$, the LCFS-S policy is age-optimal, i.e., the LCFS-S satisfies* (9) *and* (10)*.*

Proof. This corollary is a direct result of Theorem 6. □

1) Simulation Results: We present some simulation results to compare the age performance of the prmp-LGFS-R policy with other policies. The packet service times are exponentially distributed with mean $1/\mu = 1$. The inter-generation times are *i.i.d.* Erlang-2 distribution with mean $1/\lambda$. The number of servers is m. Hence, the traffic intensity is $\rho = \lambda/m\mu$. [4] The queue size is B, which is a non-negative integer.

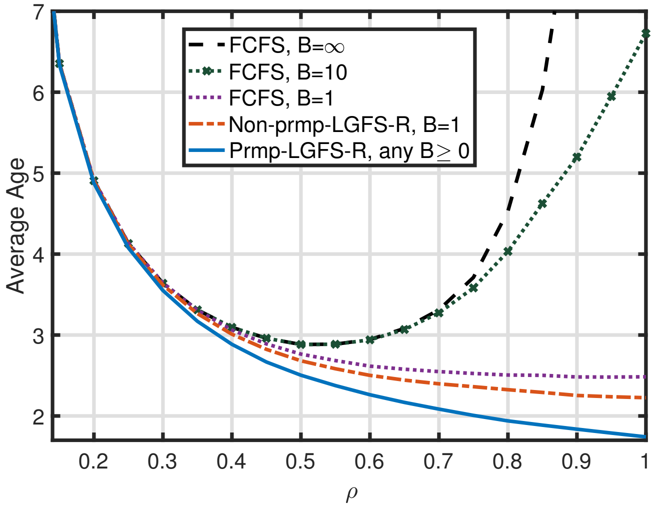


Figure 3: Average age versus traffic intensity ρ for an update system with $m = 1$ server, queue size B, and *i.i.d.* exponential service times.

[4] Throughout this paper, the traffic intensity ρ is computed without considering replications (i.e., ρ is calculated when $r = 1$).

Figure 3 illustrates the time-average age versus ρ for an information-update system with $m = 1$ server. The time difference $(a_i - s_i)$ between packet generation and arrival is zero, i.e., the update packets arrive in the same order of their generation times. We can observe that the prmp-LGFS-R policy achieves a smaller age than the FCFS policy analyzed in [5], and the non-preemptive LGFS policy with queue size $B = 1$ which is equivalent to "M/M/1/2*" in [8], [9] in this case. Note that in these prior studies, the age was characterized only for the special case of Poisson arrival process. Moreover, with ordered arrived packets at the server, the LGFS policy and LCFS policy have the same age performance.

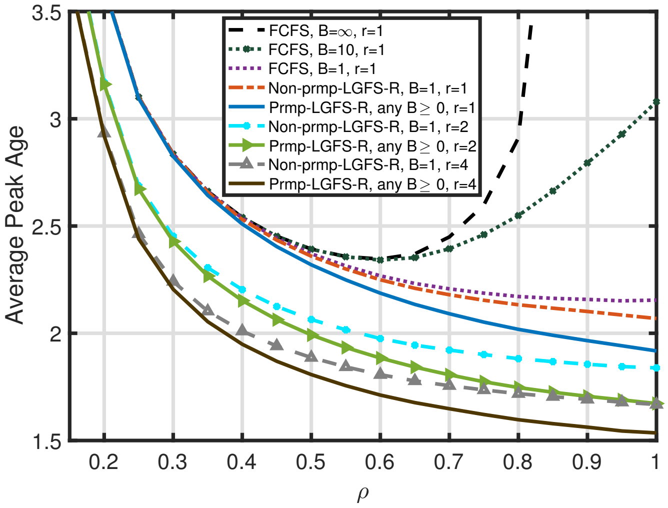


Figure 4: Average peak age versus traffic intensity ρ for an update system with $m = 4$ servers, queue size B, maximum replication degree r, and *i.i.d.* exponential service times.

Figure 4 plots the average peak age versus ρ for an information-update system with $m = 4$ servers. The time difference between packet generation and arrival, i.e., $a_i - s_i$, is modeled to be either 1 or 100, with equal probability. The maximum replication degree r is either 1, 2, or 4. For each r, we found that the prmp-LGFS-R policy achieves better age performance than other policies that belong to the policy space Π_r. For example, the age performance of the prmp-LGFS-R policy when $r = 2$ is better than the age performance of the other policies that are plotted when r equal to 1 and 2. Note that the age performance of the prmp-LGFS-R policy remains the same for any queue size $B \geq 0$. However, the age performance of the non-prmp-LGFS-R policy and FCFS policy varies with the queue size B. We also observe that the average peak age in case of FCFS policy with $B = \infty$ blows up when the traffic intensity is high. This is due to the increased congestion in the network which leads to a delivery of stale packets. Moreover, in case of FCFS policy with $B = 10$, the average peak age is high but bounded at high traffic intensity, since the fresh packet has a better opportunity to be delivered in a relatively short period compared with FCFS policy with $B = \infty$. These numerical results agree with Theorem 6.

B. NBU Service Time Distributions

The next question we proceed to answer is whether for an important class of distributions that are more general than exponential, age-optimality or near age-optimality can be achieved. We consider the class of NBU packet service time distributions, which are defined as follows.

Definition 6. ***New-Better-than-Used distributions:*** *Consider a non-negative random variable Z with complementary cumulative distribution function (CCDF) $\bar{F}(z) = \mathbb{P}[Z > z]$. Then, Z is **New-Better-than-Used (NBU)** if for all $t, \tau \geq 0$*

$$\bar{F}(\tau + t) \leq \bar{F}(\tau)\bar{F}(t). \tag{11}$$

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

Next, we show that near age-optimality can be achieved when the service times are NBU. We propose a policy called non-preemptive LGFS with replication (non-prmp-LGFS-R). The non-prmp-LGFS-R policy has the same main features of the prmp-LGFS-R policy except that the non-prmp-LGFS-R policy does not allow packet preemption. Moreover, under the non-prmp-LGFS-R policy, the fresh packet replaces the packet with the smallest generation time in the queue when it has a finite buffer size that is full. The description of the non-prmp-LGFS-R policy is depicted in Algorithm 3: This algorithm explains the procedures that the non-prmp-LGFS-R policy follows in the case of packet arrival and departure events as follows.

- **Packet arrival event:** If a new packet p_i arrives and all servers are busy, then we have two cases.
  Case 1: The buffer is full. In this case, packet p_i is either dropped or replaces another packet in the queue depending on its generation time, as depicted in Steps 7-12.
  Case 2: The buffer is not full. In this case, packet p_i is stored directly in the queue, as depicted in Steps 13-15.

Algorithm 3: Non-prmp-LGFS-R policy.

```
1  $\delta := 0$;  // $\delta$ is the smallest generation time of the packets in the queue
2  $I := m$;  // $I$ is the number of idle servers
3  $k := \lfloor \frac{m}{r} \rfloor$;  // $k$ is number of packets that each one of them can be replicated on $r$ servers
4  while the system is ON do
5      if a new packet $p_i$ with generation time $s$ arrives then
6          if I=0 then  // All servers are busy
7              if Buffer is full then
8                  if $s > \delta$ then  // Packet $p_i$ carries fresh information
9                      Packet $p_i$ replaces the packet with generation time $\delta$ in the queue;
10                 else  // Packet $p_i$ is stale
11                     Drop packet $p_i$;
12                 end
13             else
14                 Store packet $p_i$ in the queue;
15             end
16             Update $\delta$;
17         else  // At least one of the servers is idle
18             Replicate packet $p_i$ on $\min\{r, I\}$ idle servers;
19             Update $I$;
20         end
21         if a packet $p_l$ is delivered then
22             Cancel the remaining replicas of packet $p_l$;
23             Update $I$;
24             Find packet $p_j$ that is replicated on $(m - kr)$ servers;
25             if the queue is empty and packet $p_j$ exists then
26                 Replicate packet $p_j$ on extra $((k+1)r - m)$ idle servers;
27             else if the queue is not empty then
28                 Pick the packet with the largest generation time in the queue $p_h$;
29                 if packet $p_j$ exists and generation time of packet $p_j$ > generation time of packet $p_h$ then
30                     Replicate packet $p_j$ on extra $((k+1)r - m)$ idle servers;
31                     Update $I$;
32                 end
33                 Replicate packet $p_h$ on $\min\{r, I\}$ idle servers;
34             end
35             Update $I$;
36             Update $\delta$;
37         end
38     end
39 end
```

If there are idle servers, then 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 illustrated in Steps 17-20.

- **Packet departure event:** If a packet p_l is delivered, we cancel all the remaining replicas of packet p_l. Also, if there is a packet p_j that is replicated on fewer servers than r servers, then packet p_j is replicated on extra $((k+1)r - m)$ servers under two cases.

  Case A: If the queue is empty, as depicted in Steps 24-26.

  Case B: If the queue is not empty, but the generation time of packet p_j is greater than the largest generation time of the packets in the queue, as depicted in Steps 27-32.

  Finally, if the queue is not empty, the packet with the largest generation time in the queue is replicated on the available idle servers such that the total number of replicas of this packet does not exceed r, as illustrated in Step 33.

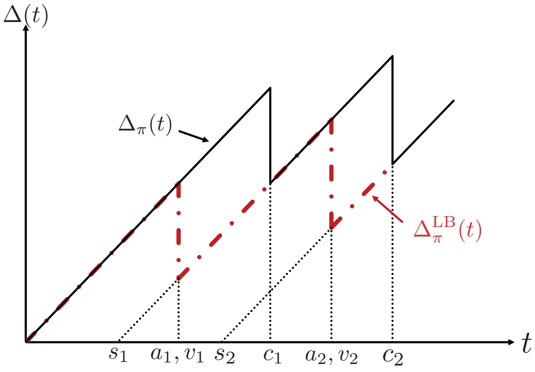


Figure 5: The evolution of Δ_π^{LB} and Δ_π in a single server queue. We assume that $a_1 > s_1$ and $a_2 > c_1 > s_2$. Thus, we have $v_1 = a_1$ and $v_2 = a_2$.

It is important to emphasize that the non-prmp-LGFS-R policy is a causal policy, i.e., its scheduling decisions are made based on the history and current state of the system and do not require the knowledge of the future packet generation and arrival times. To show that policy non-prmp-LGFS-R can come close to age-optimal, we need to construct an age lower bound as follows:

Let v_i denote the earliest time that packet i has started service (the earliest assignment time of packet i to a server), which is a function of the scheduling policy π. Define a function $\Delta_\pi^{\text{LB}}(t)$ as

$$\Delta_\pi^{\text{LB}}(t) = t - \max\{s_i : v_i(\pi) \le t\}. \qquad (12)$$

The process of $\Delta_\pi^{\text{LB}}(t)$ is given by $\Delta_\pi^{\text{LB}} = \{\Delta_\pi^{\text{LB}}(t), t \in [0, \infty)\}$. The definition of the process $\Delta_\pi^{\text{LB}}(t)$ is similar to that of the age process of policy π except that the packet completion times are replaced by their assignment times to the servers. In this case, the process $\Delta_\pi^{\text{LB}}(t)$ increases linearly with t but is reset to a smaller value with the assignment of a fresher packet to a server under policy π, as shown in Fig. 5. The process $\Delta_{\text{non-prmp-LGFS-R}}^{\text{LB}}$ is a lower bound of all policies in Π_m in the following sense.

Lemma 11. *Suppose that the packet service times are NBU, and* i.i.d. *across servers and the packets assigned to the same* *server, then for all $\mathcal{I}$ satisfying $B \geq 1$, and $\pi \in \Pi_m$*

$$[\Delta^{LB}_{\textit{non-prmp-LGFS-R}}|\mathcal{I}] \leq_{st} [\Delta_\pi|\mathcal{I}]. \tag{13}$$

Proof. See Appendix B. □

We can now proceed to characterize the age performance of policy non-prmp-LGFS-R. Let $X_1, \ldots, X_m$ denote the *i.i.d.* packet service times of the m servers, with mean $E[X_l] = E[X] < \infty$. We use Lemma 11 to prove the following theorem.

Theorem 12. *Suppose that the packet service times are NBU, and* i.i.d. *across servers and the packets assigned to the same server, then for all $\mathcal{I}$ satisfying $B \geq 1$*

(a)
$$\min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] \leq [\bar{\Delta}_{\textit{non-prmp-LGFS-R}}|\mathcal{I}] \leq \min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] + \mathbb{E}[X]. \tag{14}$$

(b) *If there is a positive integer a such that $m = ar$, then*
$$\min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] \leq [\bar{\Delta}_{\textit{non-prmp-LGFS-R}}|\mathcal{I}] \leq \min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] + \mathbb{E}[\min_{l=1,\ldots,r} X_l], \tag{15}$$

where $\bar{\Delta}_\pi = \limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T \Delta_\pi(t)dt]}{T}$ is the average age under policy π.

Proof. See Appendix C. □

Theorem 12 tells us that for arbitrary sequence of packet generation times $(s_1, s_2, \ldots)$, sequence of packet arrival times $(a_1, a_2, \ldots)$, number of servers m, maximum replication degree r, and buffer size $B \geq 1$, the non-prmp-LGFS-R policy is within a constant age gap from the optimum average age among policies in Π_m. It is important to emphasize that policy non-prmp-LGFS-R with a maximum replication degree r can be near age-optimal compared with policies with any maximum replication degree. Also, when the update packets arrive in the same order of their generation times and there is no replication, the non-prmp-LGFS-R policy when $B = 1$ becomes LCFS with preemption only in waiting (LCFS-W) for a single source case in [17], or the "M/M/1/2*" in [8], [9]. Thus, these policies can achieve near age-optimality in this case. The following corollary emphasizes this relationship.

Corollary 13. *Under the conditions of Theorem 12, if the packets arrive to the queue in the same order of their generation times, replications are not allowed ($r = 1$), and $B = 1$, then for all $\mathcal{I}$, the LCFS-W policy and the "M/M/1/2*" policy are near age-optimal, i.e., these policies satisfy* (14).

Proof. This corollary is a direct result of Theorem 12. □

1) Simulation Results: We now provide simulation results to illustrate the age performance of different policies when the service times are NBU. The inter-generation times are *i.i.d.* Erlang-2 distribution with mean $1/\lambda$. The time difference $(a_i - s_i)$ between packet generation and arrival is zero. The maximum replication degree r is either 1 or 4.

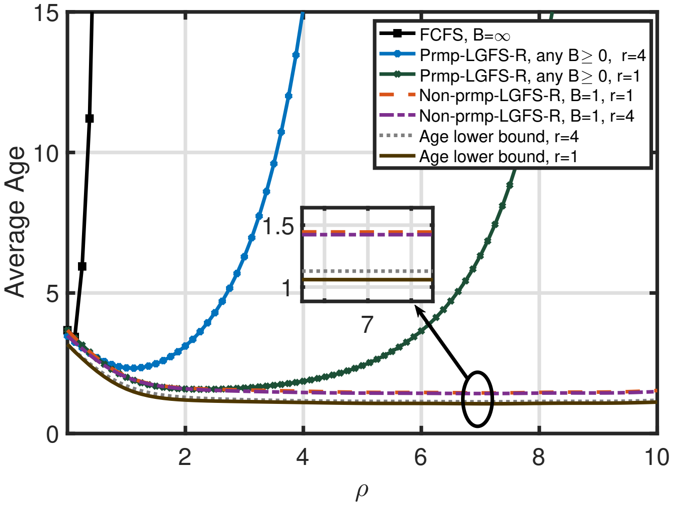


Figure 6: Average age versus traffic intensity ρ for an update system with $m = 4$ servers, queue size B, maximum replication degree r, and *i.i.d* NBU service times.

Figure 6 plots the average age versus ρ for an information-update system with $m = 4$ servers. The packet service times are the sum of a constant .25 and a value drawn from an exponential distribution with mean .25. Hence, the mean service time is $1/\mu = .5$. The "Age lower bound" curves are generated by using $\frac{\int_0^T \Delta^{\text{LB}}_{\text{non-prmp-LGFS-R}}(t)dt}{T}$ when r is 1 and 4, and $B = 1$ which, according to Lemma 11, are lower bounds of the optimum average age. We can observe that the gap between the "Age lower bound" curves and the average age of the non-prmp-LGFS-R policy when $r = 1$ and $r = 4$ is no larger than $E[X] = 1/\mu = .5$, which agrees with Theorem 12. This is a surprising result since it was shown in [16], [25], [26] that replication policies are far from the optimum delay and throughput performance for NBU service time distributions. Moreover, we can observe that the average age of the prmp-LGFS-R policies blows up when the traffic intensity is high. This is because the packet service times do not have the memoryless property in this case. Hence, when a packet is preempted, the service time of a new packet is probably longer than the remaining service time of the preempted packet. Because the arrival rate is high, packet preemption happens frequently, which leads to infrequent packet delivery and increases the age, as observed in [8].

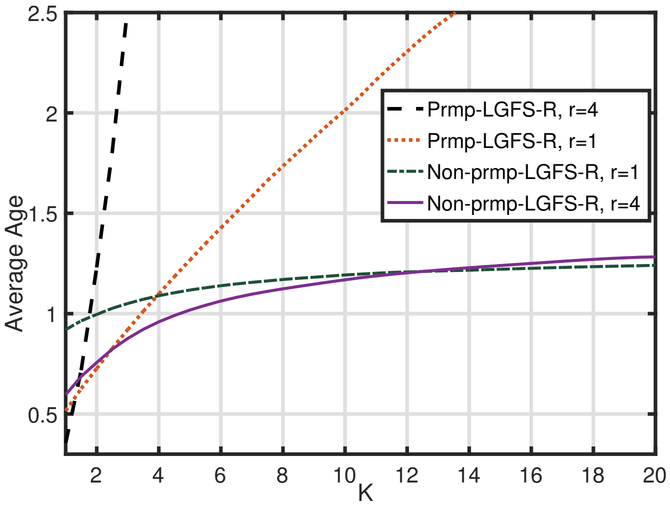


Figure 7: Average age under gamma service time distributions with different shape parameter K, where $m = 4$ servers, queue size $B = \infty$, and maximum replication degree r.

Figure 7 plots the average age under gamma service time distributions with different shape parameter K, where $m = 4$, $B = \infty$, and the traffic intensity $\rho = \lambda/m\mu = 1.8$. The mean of the gamma service time distributions is normalized to $1/\mu = 1$. Note that the average age of the FCFS policy in this case is extremely high and hence is not plotted in this figure. One can notice that packet replication and preemption affect the age performance of the plotted policies. In particular, we found that packet replication improves the age performance of the non-prmp-LGFS-R policy when the shape parameter $K \leq 12.5$, where the non-prmp-LGFS-R policy for $r = 4$ outperforms the case of $r = 1$. This is because the variance (variability) of the normalized gamma distribution is high for small values of K. Thus, packet replication can exploit the diversity provided by the four servers in this case. For the same reason, we can observe that packet replication improves the age performance of the preemptive policies when $K = 1$, where the prmp-LGFS-R policy for $r = 4$ achieves the best age performance among all plotted policies. Another reason behind the latter observation is that a gamma distribution with shape parameter $K = 1$ is an exponential distribution and hence is memoryless. Thus, packet preemption improves the age performance in this case and age-optimality can be achieved by the prmp-LGFS-R policy when $r = m$ as stated in Theorem 6 and Corollary 8. On the other hand, as the shape parameter K increases, the variance (variability) of the normalized gamma distribution decreases. This, in turn, reduces the benefit gained from the diversity provided by four servers and hence worsens the age performance of the policies that use packet replication. Moreover, as can be seen in the figure, preemption further worsens the age performance as the shape parameter K increases, and the average age of the prmp-LGFS-R policies blows up in this case. This is because of the reduction in the variability of the packet service time when the shape parameter K increases as well as the loss of the memoryless property when $K \neq 1$. Thus, preemption is not useful in this case.

C. Discussion

In this subsection, we discuss our results and compare it with prior works.

1) Preemption vs. Non-Preemption: The effect of the preemption on the age performance depends basically on the distribution of the packet service time. More specifically, when the packet service times are exponentially distributed, preemptive policies (i.e., prmp-LGFS-R) can achieve age-optimality (Theorem 6). This is because the remaining service time of a preempted packet has the same distribution as the service time of a new packet. For example, in Fig. 7, preemptive policies provide the best age performance when $K = 1$ (gamma distribution with shape parameter $K = 1$ is an exponential distribution). It is important to notice that preemptive policies can achieve age-optimality regardless of the value of ρ, even if the system is unstable when $\rho > 1$ ($\rho = 1.8$ in Fig. 7). Thus, we suggest using preemption when the packet service times are exponentially distributed. However, when the packet service times are NBU, we suggest to not use preemption. This is because the service times are no longer memoryless. Hence, when a packet is preempted, the service time of a new packet is probably longer than the remaining service time of the preempted packet. As shown in Fig. 7, the age of the preemptive LGFS policy grows to infinity at high traffic intensity for gamma distributed service times with $K > 1$. Thus, we suggest using non-preemptive policies (i.e., non-prmp-LGFS-R) instead when the packet service times are NBU.

Similar observations have been made in previous studies [11], [17]. For exponential service time distribution, Yates and Kaul showed in Theorem 3(a) of [17] that the average

age of the preemptive LCFS policy is a decreasing function of the traffic intensity ρ in M/M/1 queues as ρ grows to infinite. This agrees with our study, in which we proved that the preemptive LCFS policy is age-optimal for exponential service times and general system parameters. For NBU service time distributions, our study agrees with [11]. In particular, in [11, Numerical Results], the authors showed that the non-preemptive LCFS policy can achieve better average age than the preemptive LCFS policy. In this paper, we further show that the non-prmp-LGFS-R policy is within a small constant gap from the optimum age performance for all NBU service time distributions, which include gamma distribution as one example.

In general, our study was carried out for system settings that are more general than [17] and [11].

2) Replication vs. Non-Replication: The replication technique has gained significant attention in recent years to reduce the delay in queueing systems [14]–[16]. However, it was shown in [16], [25], [26] that replication policies are far from the optimum delay and throughput performance for NBU service time distributions. A simple explanation of this result is as follows: Let $X_1, \ldots, X_m$ be *i.i.d.* NBU random variables with mean $\mathbb{E}[X_l] = \mathbb{E}[X] < \infty$. From the properties of the NBU distributions, we can obtain [29]

$$\frac{1}{\mathbb{E}[\min_{l=1,\ldots,m} X_l]} \leq \frac{m}{\mathbb{E}[X]}. \tag{16}$$

Now, if X_l represents the packet service time of server l, then the left-hand side of (16) represents the service rate when each packet is replicated to all servers; and the right-hand side of (16) represents the service rate when there is no replication. This gives insight why packet replication can worsen the delay and throughput performance when the service times are NBU.

Somewhat to our surprise, we found that the non-prmp-LGFS-R policy is near-optimal in minimizing the age, even for NBU service time distributions. The intuition behind this result is that the age is affected by only the freshest packet, instead of all the packets in the queue. In other words, to reduce the age, we need to deliver the freshest packet as soon as possible. Obviously, we have

$$\mathbb{E}[\min_{l=1,\ldots,m} X_l] \leq \mathbb{E}[X]. \tag{17}$$

Thus, packet replication can help to reduce the age by exploiting the diversity provided by multiple servers. As shown in Fig. 7, we can observe that packet replication can improve the age performance. In particular, the age performance of the non-prmp-LGFS-R policy with $r = 4$ is better than that of the non-prmp-LGFS-R policy with $r = 1$ when $K \leq 12.5$.

V. Throughput-Delay Analysis

Recent studies on information-update systems have shown that the age-of-information can be reduced by intelligently dropping stale packets. However, packet dropping may not be appropriate in many applications such as (but not limited to):

- **News feeds:** In addition to the latest breaking news, the older news may be relevant to the user as well (e.g., to provide context or outline a different story that the user may have missed, etc.).
- **Social updates:** Users may need to be up to date with the freshest events and social posts. Nonetheless, they may also be interested in the previous posts. Thus, social applications need to update users with latest posts and previous ones as well.
- **Stock quotes:** Although the latest price in the market is very important for the traders, they may also use the history of the price change to predict the short-term price movement and attempt to profit from this. Thus, both the latest prices and historical price data are important in this case.
- **Autonomous driving or sensor information:** In such applications, while it is important to receive the latest information, historical information may also be relevant to exploit trends. For example, historical data on location information can predict the trajectory, velocity, and acceleration of the automobile. Similarly, certain types of historical sensed data may be useful to predict forest fires, earthquakes, Tsunamis, etc.

In these applications, users are interested in not just the latest updates, but also past information. Therefore, all packets may need to be successfully delivered. This motivates us to study whether it is possible to simultaneously optimize multiple performance metrics, such as age, throughput, and delay. In the sequel, we investigate the throughput and delay performance of the proposed policies. We first consider the exponential service time distribution. Then, we generalize the service time distribution to the NBU distributions. We need the following definitions:

Definition 7. ***Throughput-optimality:*** *A policy is said to be* ***throughput-optimal****, if it maximizes the expected number of delivered packets among all policies.*

The average delay under policy π is defined as

$$D_{\text{avg}}(\pi) = \frac{1}{n}\sum_{i=1}^{n}[c_i(\pi) - a_i], \tag{18}$$

where the delay of packet i under policy π is $c_i(\pi) - a_i$.[5]

Definition 8. ***Delay-optimality:*** *A policy is said to be* ***delay-optimal****, if it minimizes the expected average delay among all policies.*

Note that to maximize the throughput, we need to maximize the total number of distinct delivered packets. Moreover, to minimize the expected average delay, we need to minimize the total number of distinct packets in the system along the time. Based on these two key ideas, we prove our results in the next subsections.

A. Exponential Service Time Distribution

We study the throughput and delay performance of the prmp-LGFS-R policy when the service times are *i.i.d.* exponentially distributed. The delay and throughput performance of the prmp-LGFS-R policy are characterized as follows:

Theorem 14. *Suppose that the packet service times are exponentially distributed, and* i.i.d. *across servers and the packets assigned to the same server, then for all $\mathcal{I}$ such that $B = \infty$, the prmp-LGFS-R policy is throughput-optimal and delay-optimal among all policies in Π_m.*

Proof. We provide a proof sketch of Theorem 14. We use the coupling and forward induction to prove it. We first consider the comparison between the prmp-LGFS-R policy and an arbitrary work-conserving policy π. We couple the packet departure processes at each server such that they are identical under both policies. Then, we use the forward induction over the packet arrival and departure events to show that the total number of distinct packets in the system (excluding packet replicas) and the total number of distinct delivered packets are the same under both policies. By this, we show that the prmp-LGFS-R policy has the same throughput and mean-delay performance as any work-conserving policy (indeed, all work-conserving policies have the same throughput and delay performance). Finally, since the packet service times are *i.i.d.* across servers and the packets assigned to the same server, service idling only postpones the delivery of packets. Therefore, both throughput and delay under non-work-conserving policies will be worse. For more details, see Appendix D. □

It is worth pointing out that when the packet service times are *i.i.d* exponentially distributed, packet replication does not affect the throughput and delay performance of the replicative policies. The reasons for this observation can be summarized as follows. Because the packet service times are *i.i.d.* across the servers and the CCDF $\bar{F}$ is continuous, the probability for any two servers to complete their packets at the same time is zero. Therefore, in the replicative policies, if one copy of a replicated packet is completed on a server, the remaining replicated copies of this packet are still being processed on the other servers; these replicated packet copies are cancelled immediately and a new packet is replicated on these servers. Due to the memoryless property of the exponential distribution, the service times of the new packet copies and the remaining service times of the cancelled packets have the same distribution. Thus, packet replication does not affect the throughput and delay performance of the replicative policies.

B. NBU Service Time Distributions

Now, we consider a class of NBU service time distributions. We study the throughput and delay performance of the non-prmp-LGFS-R policy when there is no replication. The delay and throughput performance of the non-prmp-LGFS-R policy are characterized as follows:

Theorem 15. *Suppose that the packet service times are NBU, and* i.i.d. *across servers and the packets assigned to the same server, then for all $\mathcal{I}$ such that $B = \infty$ and $r = 1$, the non-prmp-LGFS-R policy is throughput-optimal and delay-optimal among all non-preemptive policies in Π_1.*

We omit the proof of Theorem 15, because it is similar to that of Theorem 14.

VI. Conclusions

In this paper, we studied the age-of-information optimization in multi-server queues. Packet replication was considered in this model, where the maximum replication degree is constrained. We considered general system settings including arbitrary arrival processes where the incoming update packets may arrive *out of order* of their generation times. We developed scheduling policies that can achieve age-optimality for any maximum replication degree when the packet service times are exponentially distributed. This optimality result not only holds for the age process, but also for any *non-decreasing functional* of the age process. Interestingly, the proposed policies can

[5] The lim sup operator is enforced on the right hand side of (18) if $n \to \infty$.

also achieve throughput and delay optimality. In addition, we investigated the class of NBU packet service time distributions and showed that LGFS policies with replication are near age-optimal for any maximum replication degree.

Appendix A
Proof of Theorem 6

We need to define the system state of any policy π:

Definition 9. *Define $U_\pi(t)$ as the largest generation time of the packets at the destination at time t under policy π. Let $\alpha_{i,\pi}(t)$ be the generation time of the packet that is being processed by server i at time t under policy π, where we set $\alpha_{i,\pi}(t) = U_\pi(t)$ if server i is idle. Then, at any time t, the system state of policy π is specified by $\mathbf{V}_\pi(t) = (U_\pi(t), \alpha_{[1],\pi}(t), \ldots, \alpha_{[m],\pi}(t))$. Note that if there is a replication, we may have $\alpha_{[i],\pi}(t) = \alpha_{[i+1],\pi}(t)$ for some i's. Without loss of generality, if h servers are sending packets with generation times less than $U_\pi(t)$ (i.e., $\alpha_{[m],\pi}(t) \leq \alpha_{[m-1],\pi}(t) \leq \ldots \leq \alpha_{[m-h+1],\pi}(t) \leq U_\pi(t)$) or h servers are idle, then we set $\alpha_{[m],\pi}(t) = \ldots = \alpha_{[m-h+1],\pi}(t) = U_\pi(t)$. Hence,*

$$U_\pi(t) \leq \alpha_{[m],\pi}(t) \leq \ldots \leq \alpha_{[1],\pi}(t). \tag{19}$$

Let $\{\mathbf{V}_\pi(t), t \in [0,\infty)\}$ be the state process of policy π, which is assumed to be right-continuous. For notational simplicity, let policy P represent the prmp-LGFS-R policy. Throughout the proof, we assume that $\mathbf{V}_P(0^-) = \mathbf{V}_\pi(0^-)$ for all $\pi \in \Pi_r$.

The key step in the proof of Theorem 6 is the following lemma, where we compare policy P with any work-conserving policy π.

Lemma 16. *Suppose that $\mathbf{V}_P(0^-) = \mathbf{V}_\pi(0^-)$ for all work conserving policies π, then for all $\mathcal{I}$*

$$[\{\mathbf{V}_P(t), t \in [0,\infty)\}|\mathcal{I}] \geq_{st} [\{\mathbf{V}_\pi(t), t \in [0,\infty)\}|\mathcal{I}]. \tag{20}$$

We use coupling and forward induction to prove Lemma 16. For any work-conserving policy π, suppose that stochastic processes $\widetilde{\mathbf{V}}_P(t)$ and $\widetilde{\mathbf{V}}_\pi(t)$ have the same stochastic laws as $\mathbf{V}_P(t)$ and $\mathbf{V}_\pi(t)$. The state processes $\widetilde{\mathbf{V}}_P(t)$ and $\widetilde{\mathbf{V}}_\pi(t)$ are coupled in the following manner: If the packet with generation time $\widetilde{\alpha}_{[i],P}(t)$ is delivered at time t as $\widetilde{\mathbf{V}}_P(t)$ evolves, then the packet with generation time $\widetilde{\alpha}_{[i],\pi}(t)$ is delivered at time t as $\widetilde{\mathbf{V}}_\pi(t)$ evolves. Such a coupling is valid because the service times are exponentially distributed and thus memoryless. Moreover, policy P and policy π have identical packet generation times $(s_1, s_2, \ldots)$ and packet arrival times $(a_1, a_2, \ldots)$. According to Proposition 5, if we can show

$$\mathbb{P}[\widetilde{\mathbf{V}}_P(t) \geq \widetilde{\mathbf{V}}_\pi(t), t \in [0,\infty)|\mathcal{I}] = 1, \tag{21}$$

then (20) is proven. To ease the notational burden, we will omit the tildes on the coupled versions in this proof and just use $\mathbf{V}_P(t)$ and $\mathbf{V}_\pi(t)$. Next, we use the following lemmas to prove (21):

Lemma 17. *At any time t, suppose that the system state of policy P is $\{U_P, \alpha_{[1],P}, \ldots, \alpha_{[m],P}\}$, and meanwhile the system state of policy π is $\{U_\pi, \alpha_{[1],\pi}, \ldots, \alpha_{[m],\pi}\}$. If*

$$U_P \geq U_\pi, \tag{22}$$

then,

$$\alpha_{[i],P} \geq \alpha_{[i],\pi}, \quad \forall i = 1, \ldots, m. \tag{23}$$

Proof. Let S denote the set of packets that have arrived to the system at the considered time t. It is important to note that the set S is invariant of the scheduling policy. If S is empty, then since $\mathbf{V}_P(0^-) = \mathbf{V}_\pi(0^-)$, Lemma 17 follows directly. Thus, we assume that S is not empty during the proof. We use $s_{[i]}$ to denote the i-th largest generation time of the packets in S. Define $k = \lfloor \frac{m}{r} \rfloor$. From the definition of the system state, condition (19), and the definition of policy P, we have

$$\begin{aligned} \alpha_{[i],P} =& \max\{s_{[j]}, U_P\}, \\ & \forall i = (j-1)r+1, \ldots, jr, \ \forall j = 1, \ldots, k, \\ \alpha_{[i],P} =& \max\{s_{[k+1]}, U_P\}, \ \forall i = kr+1, \ldots, m. \end{aligned} \tag{24}$$

Since policy π is an arbitrary policy, the servers under policy π may not process the packets with the largest generation times in the set S or policy π may replicate packets with lower generation times more than those that have larger generation times in the set S. Hence, we have

$$\begin{aligned} \alpha_{[i],\pi} \leq& \max\{s_{[j]}, U_\pi\}, \\ & \forall i = (j-1)r+1, \ldots, jr, \ \forall j = 1, \ldots, k, \\ \alpha_{[i],\pi} \leq& \max\{s_{[k+1]}, U_\pi\}, \ \forall i = kr+1, \ldots, m. \end{aligned} \tag{25}$$

where the maximization here follows from the definition of the system state. Since the set S is invariant of the scheduling policy and $U_P \geq U_\pi$, this with (24) and (25) imply

$$\alpha_{[i],P} \geq \alpha_{[i],\pi}, \quad \forall i = 1, \ldots, m, \tag{26}$$

which completes the proof. □

Lemma 18. *Suppose that under policy P, $\{U'_P, \alpha'_{[1],P}, \ldots, \alpha'_{[m],P}\}$ is obtained by delivering a packet with generation time $\alpha_{[l],P}$ to the destination in the system whose state is $\{U_P, \alpha_{[1],P}, \ldots, \alpha_{[m],P}\}$. Further, suppose that under policy π, $\{U'_\pi, \alpha'_{[1],\pi}, \ldots, \alpha'_{[m],\pi}\}$ is obtained by delivering a packet with generation time $\alpha_{[l],\pi}$ to the destination in the system whose state is $\{U_\pi, \alpha_{[1],\pi}, \ldots, \alpha_{[m],\pi}\}$. If*

$$\alpha_{[i],P} \geq \alpha_{[i],\pi}, \quad \forall i = 1, \ldots, m, \tag{27}$$

then,

$$U'_P \geq U'_\pi, \alpha'_{[i],P} \geq \alpha'_{[i],\pi}, \quad \forall i = 1, \ldots, m. \tag{28}$$

Proof. Since the packet with generation time $\alpha_{[l],P}$ is delivered under policy P, the packet with generation time $\alpha_{[l],\pi}$ is delivered under policy π, and $\alpha_{[l],P} \geq \alpha_{[l],\pi}$, we get

$$U'_P = \alpha_{[l],P} \geq \alpha_{[l],\pi} = U'_\pi. \tag{29}$$

This, together with Lemma 17, implies

$$\alpha'_{[i],P} \geq \alpha'_{[i],\pi}, \quad i = 1, \ldots, m. \tag{30}$$

Hence, (28) holds for any queue size $B \geq 0$, which completes the proof. □

Lemma 19. *Suppose that under policy P, $\{U'_P, \alpha'_{[1],P}, \ldots, \alpha'_{[m],P}\}$ is obtained by adding a packet to the system whose state is $\{U_P, \alpha_{[1],P}, \ldots, \alpha_{[m],P}\}$. Further, suppose that under policy π, $\{U'_\pi, \alpha'_{[1],\pi}, \ldots, \alpha'_{[m],\pi}\}$ is obtained by adding a packet to the system whose state is $\{U_\pi, \alpha_{[1],\pi}, \ldots, \alpha_{[m],\pi}\}$. If*

$$U_P \geq U_\pi, \tag{31}$$

then

$$U'_P \geq U'_\pi, \alpha'_{[i],P} \geq \alpha'_{[i],\pi}, \quad \forall i = 1, \ldots, m. \tag{32}$$

Proof. Since there is no packet delivery, we have

$$U'_P = U_P \geq U_\pi = U'_\pi. \tag{33}$$

This, together with Lemma 17, implies

$$\alpha'_{[i],P} \geq \alpha'_{[i],\pi}, \quad i = 1, \ldots, m. \tag{34}$$

Hence, (32) holds for any queue size $B \geq 0$, which completes the proof. □

Proof of Lemma 16. For any sample path, we have that $U_P(0^-) = U_\pi(0^-)$ and $\alpha_{[i],P}(0^-) = \alpha_{[i],\pi}(0^-)$ for $i = 1, \ldots, m$. According to the coupling between the system state processes $\{\mathbf{V}_P(t), t \in [0, \infty)\}$ and $\{\mathbf{V}_\pi(t), t \in [0, \infty)\}$, as well as Lemma 18 and 19, we get

$$[U_P(t)|\mathcal{I}] \geq [U_\pi(t)|\mathcal{I}], [\alpha_{[i],P}(t)|\mathcal{I}] \geq [\alpha_{[i],\pi}(t)|\mathcal{I}],$$

holds for all $t \in [0, \infty)$ and $i = 1, \ldots, m$. Hence, (21) follows which implies (20) by Proposition 5. This completes the proof. □

Proof of Theorem 6. As a result of Lemma 16, we have

$$[\{U_P(t), t \in [0, \infty)\}|\mathcal{I}] \geq_{\text{st}} [\{U_\pi(t), t \in [0, \infty)\}|\mathcal{I}],$$

holds for all work-conserving policies π, which implies

$$[\{\Delta_P(t), t \in [0, \infty)\}|\mathcal{I}] \leq_{\text{st}} [\{\Delta_\pi(t), t \in [0, \infty)\}|\mathcal{I}], \tag{35}$$

holds for all work-conserving policies π.

For non-work-conserving policies, since the packet service times are *i.i.d.* exponentially distributed, service idling only increases the waiting time of the packet in the system. Therefore, the age under non-work-conserving policies will be greater. As a result, we have

$$[\{\Delta_P(t), t \in [0, \infty)\}|\mathcal{I}] \leq_{\text{st}} [\{\Delta_\pi(t), t \in [0, \infty)\}|\mathcal{I}], \ \forall \pi \in \Pi_r.$$

Finally, (10) follows directly from (9) using the properties of stochastic ordering [29]. This completes the proof. □

Appendix B
Proof of Lemma 11

The proof of Lemma 11 is motivated by the proof idea of [16, Lemma 1]. For notation simplicity, let policy P represent the non-prmp-LGFS-R policy. We need to define the following parameters:

Define Γ_i and D_i as

$$\Gamma_i = \min\{v_j : s_j \geq s_i\}, \tag{36}$$

$$D_i = \min\{c_j : s_j \geq s_i\}. \tag{37}$$

where Γ_i and D_i are the smallest assignment time and completion time, respectively, of all packets that have generation times greater than that of packet i. An illustration of these parameters is provided in Fig. 8. Suppose that there are n update packets, where n is an arbitrary positive integer, no matter finite or infinite. Define the vectors $\mathbf{\Gamma} = (\Gamma_1, \ldots, \Gamma_n)$,

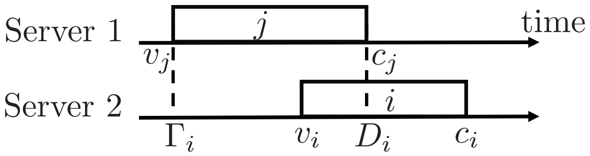


Figure 8: An illustration of v_i, c_i, Γ_i, and D_i. There are 2 servers, and $s_j > s_i$. There is no packet with generation time greater than s_i that is assigned to any of the servers before time v_j. Packet j is assigned to Server 1 at time v_j and delivered to the destination at time c_j; while packet i is assigned to Server 2 at time v_i and delivered to the destination at time c_i. The service starting time and completion time of packet j are earlier than those of packet i. Thus, we have $\Gamma_i = v_j$ and $D_i = c_j$.

and $\mathbf{D} = (D_1, \ldots, D_n)$. All these quantities are functions of the scheduling policy π.

Notice that we can deduce from (3) that the age process $\{\Delta_\pi(t), t \in [0, \infty)\}$ under any policy π is an increasing function of $\mathbf{D}(\pi)$. Moreover, we can deduce from (12) that the process $\{\Delta_P^{\text{LB}}(t), t \in [0, \infty)\}$ is an increasing function of $\mathbf{\Gamma}(P)$. According to Proposition 2, if we can show

$$[\mathbf{\Gamma}(P)|\mathcal{I}] \leq_{\text{st}} [\mathbf{D}(\pi)|\mathcal{I}], \tag{38}$$

holds for all $\pi \in \Pi_m$, then (13) is proven. Hence, (38) is what we need to show. We pick an arbitrary policy $\pi \in \Pi_m$ and prove (38) using Proposition 1 into two steps.

Step 1: Consider packet 1. Define $i^* = \text{argmin}_i\, a_i$, where $s_{i^*} \geq s_1$. Since all servers are idle by time a_{i^*} and policy P is work-conserving policy, packet i^* will be assigned to a server under policy P once it arrives. Thus, from (36), we obtain

$$[\Gamma_1(P)|\mathcal{I}] = [v_{i^*}(P)|\mathcal{I}] = a_{i^*}. \tag{39}$$

Under policy π, the completion times of all packets must be no smaller than a_{i^*}. Hence, we have

$$[c_i(\pi)|\mathcal{I}] \geq a_{i^*},\ \forall i \geq 1. \tag{40}$$

This with (37) imply

$$[D_1(\pi)|\mathcal{I}] \geq a_{i^*}. \tag{41}$$

Combining (39) and (41), we get

$$[\Gamma_1(P)|\mathcal{I}] \leq [D_1(\pi)|\mathcal{I}]. \tag{42}$$

Step 2: Consider a packet j, where $2 \leq j \leq n$. We suppose that there is no packet with generation time greater than s_j that has been delivered before packet j under policy π. We need to prove that

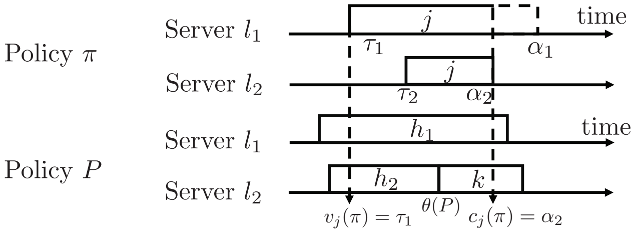


Figure 9: Illustration of packet assignments under policy π and policy P. In policy π, two copies of packet j are replicated on the server l_1 and server l_2 at time τ_1 and τ_2, where $v_j(\pi) = \min\{\tau_1, \tau_2\} = \tau_1$. Server l_2 completes one copy of packet j at time $c_j(\pi) = \alpha_2$, server l_1 cancels its redundant copy of packet j at time $c_j(\pi)$. Hence, the service duration of packet j is $[v_j(\pi), c_j(\pi)]$ in policy π. In policy P, at least one of the servers l_1 and l_2 becomes idle before time $c_j(\pi)$. In this example, server l_2 becomes idle at time $\theta(P) < c_j(\pi)$ and a fresh packet k with $s_k \geq s_j$ starts its service on server l_2 at time $\theta(P)$.

$$\begin{aligned}
&[\Gamma_j(P)|\mathcal{I}, \Gamma_1(P) = \gamma_1, \ldots, \Gamma_{j-1}(P) = \gamma_{j-1}] \\
\leq_{\text{st}}\ &[D_j(\pi)|\mathcal{I}, D_1(\pi) = d_1, \ldots, D_{j-1}(\pi) = d_{j-1}] \\
&\text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1.
\end{aligned} \tag{43}$$

For notational simplicity, define $\Gamma^{j-1} \triangleq \{\Gamma_1(P) = \gamma_1, \ldots, \Gamma_{j-1}(P) = \gamma_{j-1}\}$ and $D^{j-1} \triangleq \{D_1(\pi) = d_1, \ldots, D_{j-1}(\pi) = d_{j-1}\}$. We will show that there is at least one server under policy P that can serve a new packet at a time that is stochastically smaller than the completion time of packet j under policy π. At this time, there are two possible cases under policy P. One of them is that the idle server processes a packet with generation time greater than s_j. The other one is that the idle server processes a packet with generation time less than s_j or there is no packet to be processed. We will show that (43) holds in either case.

As illustrated in Fig. 9, suppose that u copies of packet j are replicated on the servers $l_1, l_2, \ldots, l_u$ at the time epochs $\tau_1, \tau_2, \ldots, \tau_u$ in policy π, where $v_j(\pi) = \min_{w=1,\ldots,u} \tau_w$.[6] In addition, suppose that server l_w will complete serving its copy of packet j at time α_w if there is no cancellation. Then, one of these u servers will complete one copy of packet j at time $c_j(\pi) = \min_{w=1,\ldots,u} \alpha_w$, which is the earliest among these u servers. Hence, packet j starts service at time $v_j(\pi)$ and completes service at time $c_j(\pi)$ in policy π. In policy P, let h_w

[6] If $u = 1$, then either there is no replication or policy π decides not to replicate packet j.

represent the index of the last packet that has been assigned to server l_w before time τ_w. Suppose that under policy P, server l_w has spent χ_{l_w} ($\chi_{l_w} \geq 0$) seconds on serving packet h_w before time τ_w. Let R_{l_w} denote the remaining service time of server l_w for serving packet h_w after time τ_w in policy P. Let $X^{\pi}_{l_w} = \alpha_w - \tau_w$ denote the service time of one copy of packet j in server l_w under policy π and $X^{P}_{l_w} = \chi_{l_w} + R_{l_w}$ denote the service time of packet h_w in server l_w under policy P. The CCDF of R_{l_w} is given by

$$\mathbb{P}[R_{l_w} > s] = \mathbb{P}[X^{P}_{l_w} - \chi_{l_w} > s | X^{P}_{l_w} > \chi_{l_w}]. \quad (44)$$

Because the packet service times are NBU, we can obtain that for all $s, \chi_{l_w} \geq 0$

$$\begin{aligned} &\mathbb{P}[X^{P}_{l_w} - \chi_{l_w} > s | X^{P}_{l_w} > \chi_{l_w}] = \\ &\mathbb{P}[X^{\pi}_{l_w} - \chi_{l_w} > s | X^{\pi}_{l_w} > \chi_{l_w}] \leq \mathbb{P}[X^{\pi}_{l_w} > s]. \end{aligned} \quad (45)$$

By combining (44) and (45), we obtain

$$R_{l_w} \leq_{\text{st}} X^{\pi}_{l_w}. \quad (46)$$

Because the packet service times are independent across the servers, by Lemma 13 of [16], $R_{l_1}, \ldots, R_{l_u}$ are mutually independent. By Proposition 3 and (46), we can obtain

$$\min_{w=1,\ldots,u} \tau_w + R_{l_w} \leq_{\text{st}} \min_{w=1,\ldots,u} \tau_w + X^{\pi}_{l_w} = \min_{w=1,\ldots,u} \alpha_w. \quad (47)$$

From (47) we can deduce that at least one of the servers $l_1, \ldots, l_u$, say server l_z, becomes available to serve a new packet under policy P at a time that is stochastically smaller than the time $c_j(\pi) = \min_{w=1,\ldots,u} \alpha_w$. Let $\theta(P)$ denote the time that server l_z becomes available to serve a new packet in policy P. According to (47), we have

$$\begin{aligned} &[\theta(P) | \mathcal{I}, \Gamma^{j-1}] \leq_{\text{st}} [c_j(\pi) | \mathcal{I}, D^{j-1}] \\ &\text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1. \end{aligned} \quad (48)$$

At time $\theta(P)$, we have two possible cases under policy P:

Case 1: A fresh packet k is assigned at time $\theta(P)$ to server l_z under policy P such that $s_k \geq s_j$, as shown in Fig. 10(a). Hence, we obtain

$$\begin{aligned} &[v_k(P) | \mathcal{I}, \Gamma^{j-1}] = [\theta(P) | \mathcal{I}, \Gamma^{j-1}] \leq_{\text{st}} [c_j(\pi) | \mathcal{I}, D^{j-1}] \\ &\text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1. \end{aligned} \quad (49)$$

Since $s_k \geq s_j$, (36) implies

$$[\Gamma_j(P) | \mathcal{I}, \Gamma^{j-1}] \leq [v_k(P) | \mathcal{I}, \Gamma^{j-1}] \quad (50)$$

Since there is no packet with generation time greater than s_j that has been delivered before packet j under policy π, (37) implies

$$[D_j(\pi) | \mathcal{I}, D^{j-1}] = [c_j(\pi) | \mathcal{I}, D^{j-1}] \quad (51)$$

By combining (49), (50), and (51), (43) follows.

Case 2: A packet with generation time smaller than s_j is assigned to server l_z or there is no packet assignment to server l_z at time $\theta(P)$ under policy P. Since policy P is a work-conserving policy, policy P serves the packet with the largest generation time first, and the packet generation times $(s_1, s_2, \ldots)$ and arrival times $(a_1, a_2, \ldots)$ are invariant of the scheduling policy, a packet k with $s_k \geq s_j$ must have been assigned to another server, call it server l', before time $\theta(P)$, as shown in Fig. 10(b). Hence, we obtain

$$\begin{aligned} &[v_k(P) | \mathcal{I}, \Gamma^{j-1}] \leq [\theta(P) | \mathcal{I}, \Gamma^{j-1}] \leq_{\text{st}} [c_j(\pi) | \mathcal{I}, D^{j-1}] \\ &\text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1. \end{aligned} \quad (52)$$

Similar to Case 1, we can use (36), (37), and (52) to show that (43) follows in this case.

It is important to note that if there is a packet y with $s_y > s_j$ and $c_y(\pi) < c_j(\pi)$ (this may occur if packet y preempts the service of packet j under policy π or packet y arrives to the system before packet j), then we replace packet j by packet y in the arguments and equations from (43) to (52) to obtain

$$\begin{aligned} &[\Gamma_y(P) | \mathcal{I}, \Gamma^{j-1}] \leq_{\text{st}} [D_y(\pi) | \mathcal{I}, D^{j-1}] \\ &\text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1. \end{aligned} \quad (53)$$

Observing that $s_y > s_j$, (36) implies

$$[\Gamma_j(P) | \mathcal{I}, \Gamma^{j-1}] \leq [\Gamma_y(P) | \mathcal{I}, \Gamma^{j-1}]. \quad (54)$$

Since $c_y(\pi) < c_j(\pi)$ and $s_y > s_j$, (37) implies

$$[D_j(\pi) | \mathcal{I}, D^{j-1}] = [D_y(\pi) | \mathcal{I}, D^{j-1}]. \quad (55)$$

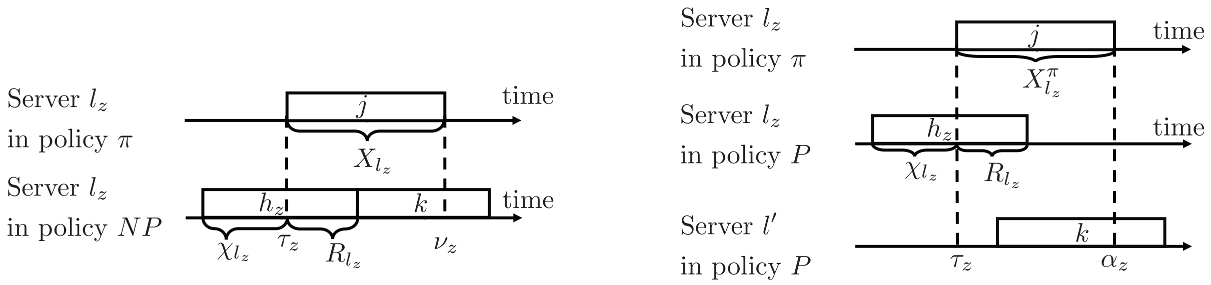
By combining (53), (54), and (55), we can prove (43) in this case too. Now, substituting (42) and (43) into Proposition 1, (38) is proven. This completes the proof.

APPENDIX C
PROOF OF THEOREM 12

For notation simplicity, let policy P represent the non-prmp-LGFS-R policy.

Proof of Theorem 12.(a). We prove Theorem 12.(a) into two steps:

Step 1: We will show that the average gap between Δ^{LB}_{P} and Δ_P is upper bounded by $\mathbb{E}[X]$. Recall the definitions of



(a) Case 1: Packet k is assigned to server l_z after the completion of packet h_z.

(b) Case 2: Packet k is assigned to server l' before the completion of packet h_z.

Figure 10: The possible cases to occur after the completion of packet h_z.

Γ_i and D_i from (36) and (37), respectively. Define $d_i(P) = D_i(P) - \Gamma_i(P)$. We know that there is a packet k with $s_k \geq s_i$ that starts service at time $\Gamma_i(P)$ under policy P. Without loss of generality, suppose that a server l is processing a copy of packet k. Because of replications, packet k completes service under policy P as soon as one of its replica completes service. Hence, packet k is delivered at time $c_k(P)$ that is no later than $\Gamma_i(P) + X_l$ under policy P. This implies that $c_k(P) - \Gamma_i(P) \leq X_l$. From (37), we can deduce that $D_i(P) - \Gamma_i(P) \leq c_k(P) - \Gamma_i(P) \leq X_l$. From this, we can obtain

$$\mathbb{E}[d_i] \leq \mathbb{E}[X], \ \forall i. \tag{56}$$

We now proceed to characterize the gap between Δ_P^{LB} and Δ_P. We use $\{G(t), t \in [0, \infty)\}$ to denote the gap process between Δ_P^{LB} and Δ_P. The average gap is given by

$$[\bar{G}|\mathcal{I}] = \limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T G(t)dt]}{T}. \tag{57}$$

Let τ_i denote the inter-generation time between packet i and packet $i-1$ (i.e., $\tau_i = s_i - s_{i-1}$), where $\tau = \{\tau_i, i \geq 1\}$. Note that, since the packet service times are independent of the packet generation process, we have d_i's are independent of τ. Define $N(T) = \max\{i : s_i \leq T\}$ as the number of generated packets by time T. Note that $[0, s_{N(T)}] \subseteq [0, T]$, where the length of the interval $[0, s_{N(T)}]$ is $\sum_{i=1}^{N(T)} \tau_i$. Thus, we have

$$\sum_{i=1}^{N(T)} \tau_i \leq T. \tag{58}$$

The area defined by the integral in (57) can be decomposed into a sum of disjoint geometric parts. Observing Fig. 11, the area can be approximated to the concatenation of the parallelograms $G_1, G_2, \ldots$ (G_i's are highlighted in Fig. 11).

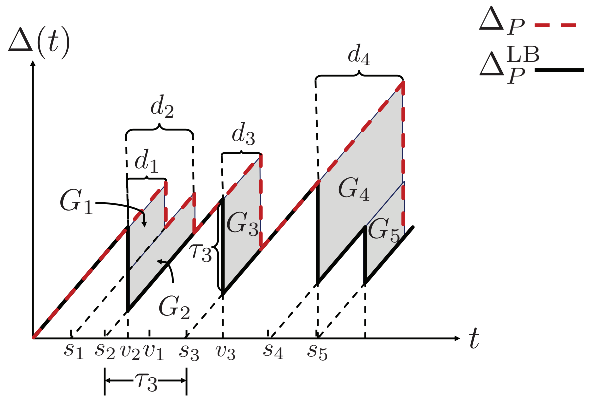


Figure 11: The evolution of Δ_P^{LB} and Δ_P in a queue with 4 servers and $r = 2$.

Note that the parallelogram G_i results after the generation of packet i (i.e., the gap that is corresponding to the packet i occurs after its generation). Since the observing time T is chosen arbitrary, when $T \geq s_i$, the total area of the parallelogram G_i is accounted in the summation $\sum_{i=1}^{N(T)} G_i$, while it may not be accounted in the integral $\int_0^T G(t)dt$. This implies that

$$\sum_{i=1}^{N(T)} G_i \geq \int_0^T G(t)dt. \tag{59}$$

Combining (58) and (59), we get

$$\frac{\int_0^T G(t)dt}{T} \leq \frac{\sum_{i=1}^{N(T)} G_i}{\sum_{i=1}^{N(T)} \tau_i}. \tag{60}$$

Then, take conditional expectation given τ and $N(T)$ on both sides of (60), we obtain

$$\begin{aligned}&\frac{\mathbb{E}[\int_0^T G(t)dt|\tau, N(T)]}{T} \leq \\ &\frac{\mathbb{E}[\sum_{i=1}^{N(T)} G_i|\tau, N(T)]}{\sum_{i=1}^{N(T)} \tau_i} = \frac{\sum_{i=1}^{N(T)} \mathbb{E}[G_i|\tau, N(T)]}{\sum_{i=1}^{N(T)} \tau_i},\end{aligned} \tag{61}$$

where the second equality follows from the linearity of the expectation. From Fig. 11, G_i can be calculated as

$$G_i = \tau_i d_i. \tag{62}$$

Substituting by (62) into (61), yields

$$\begin{aligned}&\frac{\mathbb{E}[\int_0^T G(t)dt|\tau, N(T)]}{T} \leq \\ &\frac{\sum_{i=1}^{N(T)} \mathbb{E}[\tau_i d_i|\tau, N(T)]}{\sum_{i=1}^{N(T)} \tau_i} = \frac{\sum_{i=1}^{N(T)} \tau_i \mathbb{E}[d_i|\tau, N(T)]}{\sum_{i=1}^{N(T)} \tau_i}.\end{aligned} \tag{63}$$

Note that d_i's are independent of τ. Thus, we have $\mathbb{E}[d_i|\tau, N(T)] = \mathbb{E}[d_i] \leq \mathbb{E}[X]$ for all i. Substituting this into (63), yields

$$\frac{\mathbb{E}[\int_0^T G(t)dt|\tau, N(T)]}{T} \leq \frac{\sum_{i=1}^{N(T)} \tau_i \mathbb{E}[X]}{\sum_{i=1}^{N(T)} \tau_i} = \mathbb{E}[X], \tag{64}$$

by the law of iterated expectations, we have

$$\frac{\mathbb{E}[\int_0^T G(t)dt]}{T} \leq \mathbb{E}[X]. \tag{65}$$

Taking the $\limsup$ of both side of (65) when $T \to \infty$, yields

$$\limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T G(t)dt]}{T} \leq \mathbb{E}[X]. \tag{66}$$

Equation (66) tells us that the average gap between Δ_P^{LB} and Δ_P is no larger than $\mathbb{E}[X]$.

Step 2: We prove (14). Since Δ_P^{LB} is a lower bound of the age process of policy P and the average gap between Δ_P^{LB} and Δ_P is no larger than $\mathbb{E}[X]$, we obtain

$$[\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}] \leq [\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] + \mathbb{E}[X], \tag{67}$$

where $\bar{\Delta}_P^{\text{LB}} = \limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T \Delta_P^{\text{LB}}(t)dt]}{T}$. From Lemma 11, we have for all $\mathcal{I}$ satisfying $B \geq 1$, and $\pi \in \Pi_m$

$$[\{\Delta_P^{\text{LB}}(t), t \in [0, \infty)\}|\mathcal{I}] \leq_{\text{st}} [\{\Delta_\pi(t), t \in [0, \infty)\}|\mathcal{I}], \tag{68}$$

which implies that

$$[\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] \leq [\bar{\Delta}_\pi|\mathcal{I}], \tag{69}$$

holds for all $\pi \in \Pi_m$. As a result, we get

$$[\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] \leq \min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}]. \tag{70}$$

Since policy non-prmp-LGFS-R is a feasible policy, we get

$$\min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}]. \tag{71}$$

Combining (67), (70), and (71), we get

$$\min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}] \leq \min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] + \mathbb{E}[X], \tag{72}$$

which completes the proof. □

Proof of Theorem 12.(b). The proof of part (b) is similar to that of part (a). Define $d_i(P) = D_i(P) - \Gamma_i(P)$. We know that there is a packet k with $s_k \geq s_i$ that starts service at time $\Gamma_i(P)$ under policy P. Since $m = ar$ for a positive integer a, packet k is processed by r servers in policy P. Let $\mathcal{S}_k \subseteq \{1, \ldots, m\}$ be the set of servers that process packet k under policy P, which satisfies $|\mathcal{S}_k| = r$. Because of replications, packet k completes service under policy P as soon as one of its replica is completes service. Hence, packet k is delivered at time $c_k(P) = \Gamma_i(P) + \min_{l\in\mathcal{S}_k} X_l$ under policy P. This implies that $c_k(P) - \Gamma_i(P) = \min_{l\in\mathcal{S}_k} X_l$. From (37), We can deduce that $D_i(P) - \Gamma_i(P) \leq c_k(P) - \Gamma_i(P) = \min_{l\in\mathcal{S}_k} X_l$. From this, we can obtain

$$\mathbb{E}[d_i] \leq \mathbb{E}[\min_{l=1,\ldots,r} X_l], \forall i. \tag{73}$$

Similar to part a, we use $\{G(t), t \in [0, \infty)\}$ to denote the gap process between Δ_P^{LB} and Δ_P. The average gap is given by

$$[\bar{G}|\mathcal{I}] = \limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T G(t)dt]}{T}. \tag{74}$$

Following the same steps as in the proof of part (a), we can show that

$$\limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T G(t)dt]}{T} \leq \mathbb{E}[\min_{l=1,\ldots,r} X_l]. \tag{75}$$

Equation (75) tells us that the average gap between Δ_P^{LB} and Δ_P is no larger than $\mathbb{E}[\min_{l=1,\ldots,r} X_l]$. This and the fact that Δ_P^{LB} is a lower bound of the age process of policy P, imply

$$[\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}] \leq [\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] + \mathbb{E}[\min_{l=1,\ldots,r} X_l]. \tag{76}$$

Similar to part (a), we can use (76) with Lemma 11 to show that

$$\min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}] \leq \min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] + \mathbb{E}[\min_{l=1,\ldots,r} X_l], \tag{77}$$

which completes the proof. □

Appendix D
Proof of Theorem 14

We follow the same proof technique of Theorem 6. We start by comparing policy P (prmp-LGFS-R policy) with an arbitrary work-conserving policy π. For this, we need to define the system state of any policy π:

Definition 10. *At any time t, the system state of policy π is specified by $H_\pi(t) = (N_\pi(t), \gamma_\pi(t))$, where $N_\pi(t)$ is the total number of distinct packets in the system at time t (excluding packet replicas). Define $\gamma_\pi(t)$ as the total number of distinct packets that are delivered to the destination at time t. Let $\{H_\pi(t), t \in [0,\infty)\}$ be the state process of policy π, which is assumed to be right-continuous.*

To prove Theorem 14, we will need the following lemma.

Lemma 20. *For any work-conserving policy π, if $H_P(0^-) = H_\pi(0^-)$ and $B = \infty$, then $[\{H_P(t), t \in [0,\infty)\}|\mathcal{I}]$ and $[\{H_\pi(t), t \in [0,\infty)\}|\mathcal{I}]$ are of the same distribution.*

Suppose that $\{\widetilde{H}_P(t), t \in [0,\infty)\}$ and $\{\widetilde{H}_\pi(t), t \in [0,\infty)\}$ are stochastic processes having the same stochastic laws as $\{H_P(t), t \in [0,\infty)\}$ and $\{H_\pi(t), t \in [0,\infty)\}$. Now, we couple the packet delivery times during the evolution of $\widetilde{H}_P(t)$ to be identical with the packet delivery times during the evolution of $\widetilde{H}_\pi(t)$. Such a coupling is valid because the service times are exponentially distributed, and hence, memoryless.

To ease the notational burden, we will omit the tildes henceforth on the coupled versions and just use $\{H_P(t)\}$ and $\{H_\pi(t)\}$. The following two lemmas are needed to prove Lemma 20:

Lemma 21. *Suppose that under policy P, $\{N'_P, \gamma'_P\}$ is obtained by delivering a packet to the destination in the system whose state is $\{N_P, \gamma_P\}$. Further, suppose that under policy π, $\{N'_\pi, \gamma'_\pi\}$ is obtained by delivering a packet to the destination in the system whose state is $\{N_\pi, \gamma_\pi\}$. If*

$$N_P = N_\pi, \gamma_P = \gamma_\pi,$$

then

$$N'_P = N'_\pi, \gamma'_P = \gamma'_\pi. \tag{78}$$

Proof. Because the packet service times are *i.i.d.* and the CCDF $\bar{F}$ is continuous, the probability for any two servers to complete their packets at the same time is zero. Therefore, in policy P, if one copy of a replicated packet is completed on a server, the remaining replicated copies of this packet are still being processed on the other servers; these replicated packet copies are cancelled immediately and a new packet is replicated on these servers. Since there is a packet delivery, we have

$$N'_P = N_P - 1 = N_\pi - 1 = N'_\pi,$$
$$\gamma'_P = \gamma_P + 1 = \gamma_\pi + 1 = \gamma'_\pi.$$

Hence, (78) holds, which complete the proof. □

Lemma 22. *Suppose that under policy P, $\{N'_P, \gamma'_P\}$ is obtained by adding a new packet to the system whose state is $\{N_P, \gamma_P\}$. Further, suppose that under policy π, $\{N'_\pi, \gamma'_\pi\}$ is obtained by adding a new packet to the system whose state is $\{N_\pi, \gamma_\pi\}$. If*

$$N_P = N_\pi, \gamma_P = \gamma_\pi,$$

then

$$N'_P = N'_\pi, \gamma'_P = \gamma'_\pi. \tag{79}$$

Proof. Because $B = \infty$, no packet is dropped in policy P and policy π. Since there is a new added packet to the system, we have

$$N'_P = N_P + 1 = N_\pi + 1 = N'_\pi.$$

Also, there is no packet delivery, hence

$$\gamma'_P = \gamma_P = \gamma_\pi = \gamma'_\pi.$$

Thus, (79) holds, which complete the proof. □

Proof of Lemma 20. For any sample path, we have that $N_P(0^-) = N_\pi(0^-)$ and $\gamma_P(0^-) = \gamma_\pi(0^-)$. According to the coupling between the system state processes $\{H_P(t), t \in [0,\infty)\}$ and $\{H_\pi(t), t \in [0,\infty)\}$, as well as Lemma 21 and 22, we get

$$[N_P(t)|\mathcal{I}] = [N_\pi(t)|\mathcal{I}], [\gamma_P(t)|\mathcal{I}] = [\gamma_\pi(t)|\mathcal{I}],$$

holds for all $t \in [0,\infty)$. This implies that $[\{H_P(t), t \in [0,\infty)\}|\mathcal{I}]$ and $[\{H_\pi(t), t \in [0,\infty)\}|\mathcal{I}]$ are of the same distribution, which completes the proof. □

Proof of Theorem 14. As a result of Lemma 20, $[\{\gamma_P(t), t \in [0,\infty)\}|\mathcal{I}]$ and $[\{\gamma_\pi(t), t \in [0,\infty)\}|\mathcal{I}]$ are of the same distribution. This implies that policy P and policy π have the same throughput performance. Also, from Lemma 20, we have that $[\{N_P(t), t \in [0,\infty)\}|\mathcal{I}]$ and $[\{N_\pi(t), t \in [0,\infty)\}|\mathcal{I}]$ are of the same distribution. Hence, policy P and policy π have the same delay performance. These imply that policy P has the same throughput and delay performance as any work-conserving policy.

Finally, since the service times are *i.i.d.*, service idling only increases the waiting time of the packet in the system. Therefore, the throughput and delay performance under non-work-conserving policies will be worse. As a result, the prmp-LGFS-R policy is throughput-optimal and delay-optimal among all policies in Π_m (indeed, all work-conserving policies with infinite buffer size $B = \infty$ have the same throughput and delay performance, and hence, they are throughput-optimal and delay-optimal). □

References

[1] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Optimizing data freshness, throughput, and delay in multi-server information-update systems," in *Proc. IEEE ISIT*, July 2016, pp. 2569–2573.

[2] B. Adelberg, H. Garcia-Molina, and B. Kao, "Applying update streams in a soft real-time database system," in *ACM SIGMOD Record*, 1995, vol. 24, pp. 245–256.

[3] J. Cho and H. Garcia-Molina, "Synchronizing a database to improve freshness," in *ACM SIGMOD Record*, 2000, vol. 29, pp. 117–128.

[4] L. Golab, T. Johnson, and V. Shkapenyuk, "Scheduling updates in a real-time stream warehouse," in *Proc. IEEE 25th International Conference on Data Engineering*, March 2009, pp. 1207–1210.

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

[6] R. D. Yates and S. Kaul, "Real-time status updating: Multiple sources," in *IEEE International Symposium on Information Theory (ISIT)*, July 2012, pp. 2666–2670.

[7] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *Proc. IEEE ISIT*, June 2015, pp. 1681–1685.

[8] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1897–1910, April 2016.

[9] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *Proc. IEEE ISIT*, June 2014, pp. 1583–1587.

[10] N. Pappas, J. Gunnarsson, L. Kratz, M. Kountouris, and V. Angelakis, "Age of information of multiple sources with queue management," in *Proc. IEEE ICC*, June 2015, pp. 5935–5940.

[11] E. Najm and R. Nasser, "Age of information: The gamma awakening," in *Proc. IEEE ISIT*, July 2016, pp. 2574–2578.

[12] B. Ji, G. R. Gupta, X. Lin, and N. B. Shroff, "Performance of low-complexity greedy scheduling policies in multi-channel wireless networks: Optimal throughput and near-optimal delay," in *2013 Proceedings IEEE INFOCOM*, April 2013, pp. 2589–2597.

[13] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to parallel computing: design and analysis of algorithms*, vol. 400, Benjamin/Cummings Redwood City, CA, 1994.

[14] S. Chen, Y. Sun, U. C. Kozat, L. Huang, P. Sinha, G. Liang, X. Liu, and N. B. Shroff, "When queueing meets coding: Optimal-latency data retrieving scheme in storage clouds," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 1042–1050.

[15] Y. Sun, Z. Zheng, C. E. Koksal, K. H. Kim, and N. B. Shroff, "Provably delay efficient data retrieving in storage clouds," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 585–593.

[16] Y. Sun, C. E. Koksal, and N. B. Shroff, "On delay-optimal scheduling in queueing systems with replications," CoRR, abs/1603.07322, Jan. 2016.

[17] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," IEEE Trans. Inf. Theory, vol. 65, no. 3, pp. 1807–1827, 2019.

[18] C. Kam, S. Kompella, G. D. Nguyen, and A. Ephremides, "Effect of message transmission path diversity on status age," *IEEE Trans. Inf. Theory*, vol. 62, no. 3, pp. 1360–1374, March 2016.

[19] Y. Sang, B. Li, and B. Ji, "The power of waiting for more than one response in minimizing the age-of-information," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[20] T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *Proc. Info. Theory and Appl. Workshop*, Feb. 2015.

[21] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *Proc. IEEE Int. Symp. Inform. Theory*, 2015.

[22] K. Chen and L. Huang, "Age-of-information in the presence of error," in *Proc. IEEE ISIT*, July 2016, pp. 2579–2583.

[23] 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 *Proc. IEEE INFOCOM*, April 2016.

[24] 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 Trans. Inf. Theory*, in press, 2017.

[25] N. B. Shah, K. Lee, and K. Ramchandran, "When do redundant requests reduce latency?," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 715–722, 2016.

[26] G. Joshi, E. Soljanin, and G. Wornell, "Efficient redundancy techniques for latency reduction in cloud systems," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 2, no. 2, pp. 12, 2017.

[27] Y. Sun, Y. Polyanskiy, and E. Uysal-Biyikoglu, "Remote estimation of the Wiener process over a channel with random delay," in *Proc. IEEE ISIT*, 2017.

[28] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Age-optimal information updates in multihop networks," in *Proc. IEEE ISIT*, 2017.

[29] M. Shaked and J. G. Shanthikumar, *Stochastic orders*, Springer Science & Business Media, 2007.

Ahmed M. Bedewy received the B.S. and M.S. degrees in electrical and electronics engineering from Alexandria University, Alexandria, Egypt, in 2011 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, at the Ohio State University, OH, USA. His research interests include wireless communication, cognitive radios, resource allocation, communication networks, information freshness, optimization, and scheduling algorithms. He received the Awarded

Certificate of Merit, First Class Honors, for being one of the top ten undergraduate students during 2006-2008 and **1-st** during 2008-2011 in electrical and electronics engineering.

Yin Sun (S'08-M'11) received his B.Eng. and Ph.D. degrees in Electronic Engineering from Tsinghua University, in 2006 and 2011, respectively. At Tsinghua, he received the Excellent Doctoral Thesis Award of Tsinghua University, among many awards and scholarships. He was a postdoctoral scholar and research associate at the Ohio State University during 2011-2017. Since Fall 2017, Dr. Sun joined Auburn University as an assistant professor in the Department of Electrical and Computer Engineering.

His research interests include wireless communications, communication networks, information freshness, information theory, and machine learning. He is the founding co-chair of the first and second Age of Information Workshops, in conjunction with the IEEE INFOCOM 2018 and 2019. The paper he coauthored received the best student paper award at IEEE WiOpt 2013.

Ness B. Shroff (S'91–M'93–SM'01–F'07) received the Ph.D. degree in electrical engineering from Columbia University in 1994. He joined Purdue University immediately thereafter as an Assistant Professor with the School of Electrical and Computer Engineering. At Purdue, he became a Full Professor of ECE and the director of a university-wide center on wireless systems and applications in 2004. In 2007, he joined The Ohio State University, where he holds the Ohio Eminent Scholar Endowed Chair in networking and communications, in the departments of ECE and CSE. He holds or has held visiting (chaired) professor positions at Tsinghua University, Beijing, China, Shanghai Jiaotong University, Shanghai, China, and IIT Bombay, Mumbai, India. He has received numerous best paper awards for his research and is listed in Thomson Reuters' on The World's Most Influential Scientific Minds, and is noted as a Highly Cited Researcher by Thomson Reuters. He also received the IEEE INFOCOM Achievement Award for seminal contributions to scheduling and resource allocation in wireless networks. He currently serves as the steering committee chair for ACM Mobihoc and Editor at Large of the IEEE/ACM Transactions on Networking.