OPTIMIZING DATA FRESHNESS IN INFORMATION UPDATE SYSTEMS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Ahmed Mohamed Abdelmohsen Bedewy, B.S. M.S.

Graduate Program in Electrical and Computer Engineering

The Ohio State University

2021

Dissertation Committee

Ness B. Shroff, Advisor

Yin Sun, Co-advisor

Atilla Eryilmaz

Abhishek Gupta

Qin Ma

ABSTRACT

In applications such as networked monitoring and control systems, wireless sensor networks, and autonomous vehicles, it is crucial for the destination node to receive timely status updates so that it can make accurate decisions. For example, a moving car with a speed of 65 mph will traverse almost 29 meters in 1 second, and hence, stale information (regarding the location of surrounding vehicles, velocities, etc.) has a dramatic serious impact on this situation. Age of information (AoI), or simply age, has been used to measure the freshness of status updates. More specifically, AoI is the age of the freshest update at the destination, i.e., it is the time elapsed since the freshest received update was generated. It should be noted that optimizing traditional network performance metrics, such as throughput or delay, does not attain the goal of timely updating. For instance, it is well known that AoI could become very large when the offered load is high or low. In other words, AoI captures the information lag at the destination, and is hence more apt for achieving the goal of timely updates.

In this thesis, we leverage rigorous theory to develop low-complexity scheduling algorithms that are apt for a wide range of information update systems. In particular, we consider the following systems:

- **Information update systems with stochastic packet arrivals:** We consider single and multihop networks with stochastic arrivals, where our goal is to answer the following fundamental questions: (i) Which queueing discipline can minimize the age? And (ii) under what conditions is the minimum age achievable? Towards this goal, we design low-complexity scheduling policies to achieve

(near) age-optimality in single and multihop networks with single source. The achieved results that we present here hold under quite general conditions, including (i) arbitrary packet generation and arrival processes, (ii) for minimizing both the age processes in stochastic ordering and any non-decreasing functional of the age processes, (iii) multi-server, single-hop networks with packet replications, and (iv) general multihop network topology.

- **Information update systems with controlled packet generation:** We consider multi-source systems, where sources can control the generation of the update packets. Our goal is to address the following information update systems: (i) We consider the system in which sources take turns to generate update packets, and forward them to their destinations one-by-one through a shared channel with random delay. There is a scheduler, that chooses the update order of the sources, and a sampler, that determines when a source should generate a new sample in its turn. We aim to solve a challenging joint sampling and scheduling optimization problem for minimizing AoI. Although the complexity that is usually inherited in joint optimization problems, we are able to develop low-complexity optimal scheduler-sampler pairs that minimize AoI, (ii) since many information update devices are battery limited, scheduling policies that prolong the battery-lifetime sometimes are just as crucial as those that ensure information freshness. Towards that end, we consider the problem of optimizing the freshness of status updates that are sent from a large number of low-power sources to a common access point. We develop asynchronized sleep-wake scheduling algorithm that minimizes AoI, subject to per-source battery lifetime constraints. Although the problem turns out to be non-convex, we devise a low-complexity near-optimal solution.

To my parents, Mohamed Bedewy and Afaf Farahat, and my wife Dalia Alshakly.

ACKNOWLEDGMENTS

It is a genuine pleasure to express my deep sense of thanks and gratitude to Prof. Ness B. Shroff and Prof. Yin Sun. They have been great advisors, and inspiring mentors. Their dedication and keen interest, above all their overwhelming attitude to help me, had been solely and mainly responsible for completing my work. Their timely advice, meticulous scrutiny, scholarly advice and scientific approach have helped me to a very great extent to accomplish this task. Without their encouragement and support, I would not have been able to reach this current level of success.

Secondly, I would like to thank my committee members, each of whom has provided patient advice and guidance throughout the research process. Thank you all for your unwavering support.

Last but not least, thank you to all of my friends who directly or indirectly helped me complete this dissertation.

VITA

October, 1989	Born in Alexandria, Egypt
2011	B.S., Electrical Engineering, Alexandria University, Egypt
2015	M.S., Electrical Engineering, Alexandria University, Egypt
2015 - Present	Graduate Research Associate, The Ohio State University

PUBLICATIONS

Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff "Optimizing data freshness, throughput, and delay in multi-server information-update systems," in *Proceedings of IEEE International Symposium on Information Theory*, 2016, pp. 2569–2573.

Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff "Age-Optimal Information Updates in Multihop Networks," in *Proceedings of IEEE International Symposium on Information Theory*, 2017, pp. 576–580.

Ahmed M. Bedewy, Yin Sun, Sastry Kompella, and Ness B. Shroff "Age-optimal Sampling and Transmission Scheduling in Multi-Source Systems," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19)*, 2019, pp. 121–130.

Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff "Minimizing the Age of Information through Queues, " *IEEE Transactions on Information Theory,* vol. 65, no. 8, pp. 5215–5232, Aug 2019.

Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff "The Age of Information in Multihop Networks," *IEEE/ACM Transactions on Networking,* vol. 27, no. 3, pp. 1248–1257, June 2019.

Ahmed M. Bedewy, Yin Sun, Rahul Singh, and Ness B. Shroff, "Optimizing Information Freshness using Low-Power Status Updates via Sleep-Wake Scheduling," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '20)*, 2020, pp. 121–130. [Running-up Award]

Jiayu Pan, Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff, "Minimizing Age of Information via Scheduling over Heterogeneous Channels," submitted to *The Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '21),* 2021.

Guidan Yao, Ahmed M. Bedewy, and Ness B. Shroff, "Battle between Rate and Error in Minimizing Age of Information," submitted to *The Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '21),* 2021.

Guidan Yao, Ahmed M. Bedewy, and Ness B. Shroff, "Age-Optimal Low-Power Status Update over Time-Correlated Fading Channel," submitted to *IEEE International Symposium on Information Theory,* 2021.

Ahmed M. Bedewy, Yin Sun, Sastry Kompella, and Ness B. Shroff, "Optimal Sampling and Scheduling for Timely Status Updates in Multi-source Networks," accepted in *IEEE Transactions on Information Theory,* 2021.

Ahmed M. Bedewy, Yin Sun, Rahul Singh, and Ness B. Shroff, "Low-Power Status Updates via Sleep-Wake Scheduling," submitted to *IEEE/ACM Transactions on Networking,* 2021.

FIELDS OF STUDY

Major Field: Electrical and Computer Engineering

Specialization: Network Science, Age of Information

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

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.

The growth in the demand for real-time information updates is expected to continue its astonishing march, driven by a plethora of applications that will affect every aspect of our lives, from entertainment and autonomy to business and healthcare. For example, in autonomous driving, real-time information related to traffic congestion and road conditions is crucial to avoid collisions and reduce congestion. Thus, a key challenge that has attracted a number of researchers is how to design efficient systems in order to ensure freshness of information at the destination. To identify the timeliness of the updates, a metric called the *age of information* (AoI), or simply *age*, was defined in, e.g., [1–4]. 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

$$\Delta(t) = t - U(t). \tag{1.0.1}$$

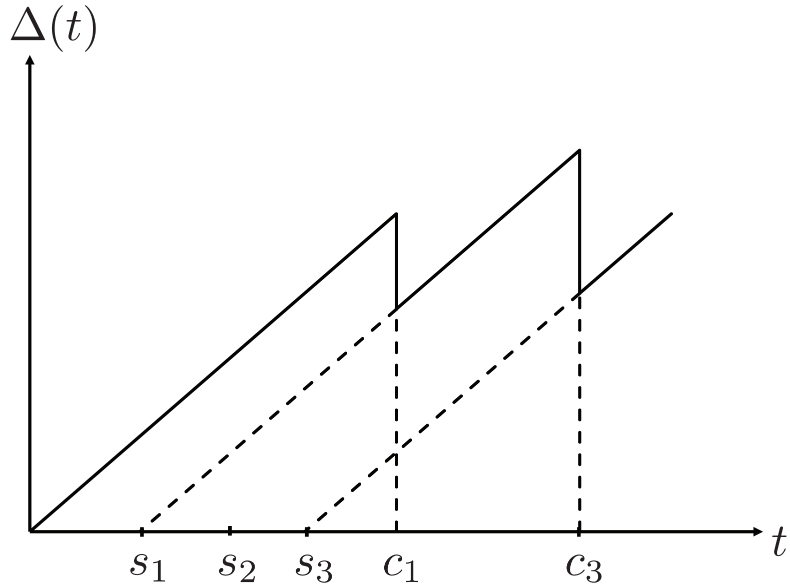


Figure 1.1: A sample path of the age process $\Delta(t)$, where s_i and c_i are the generation time and the delivery time of packet i, respectively.

Hence, age is the time elapsed since the freshest received packet was generated. As shown in Fig. 1.1, the age increases linearly with time t but is reset to a smaller value with the delivery of an even fresher packet.

Unlike traditional packet-based performance metrics, such as throughput and delay, age is a destination-based metric. For example, consider the case when a flow of information update packets are stored in a buffer and forwarded to a controller over a communication medium. If the update rate is low, then the buffer is empty and hence the delay is low. However, because the controller is updated infrequently, the AoI is high. On the other hand, if the update rate is high, then the queueing delay at the buffer is high, and so is the AoI because packets have to wait for a long time in the buffer. Hence, while delay increases monotonically with the update rate, the AoI (or staleness) first decreases and then increases. Thus, AoI captures the information lag at the destination, and is hence more apt for achieving the goal of timely updates.

In this dissertation, our goal has been to design low-complexity algorithms that optimize the data freshness for variants of information update systems (e.g., systems with stochastic packet arrivals, and systems with controlled packet generation).

1.1 Information Update Systems with Stochastic Packet Arrivals

In recent years, a variety of approaches have been investigated to reduce the age on systems with a stochastic arrival process. In [4–6], it was found in First-Come, First-Served (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 [7–9], 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 [10]. However, these studies cannot tell us: *Which queueing discipline can minimize the age? Under what conditions is the minimum age achievable?* In this dissertation, we aim to answer these two questions in single-hop networks with multiple servers and multihop networks.

1.1.1 Single-Hop Networks

We begin with answering the aforementioned questions for a single-hop network with a single source as illustrated in Fig. 1.2, where a sequence of update packets arrive at a queue with m servers and a buffer size B. Each server can be used to model a channel

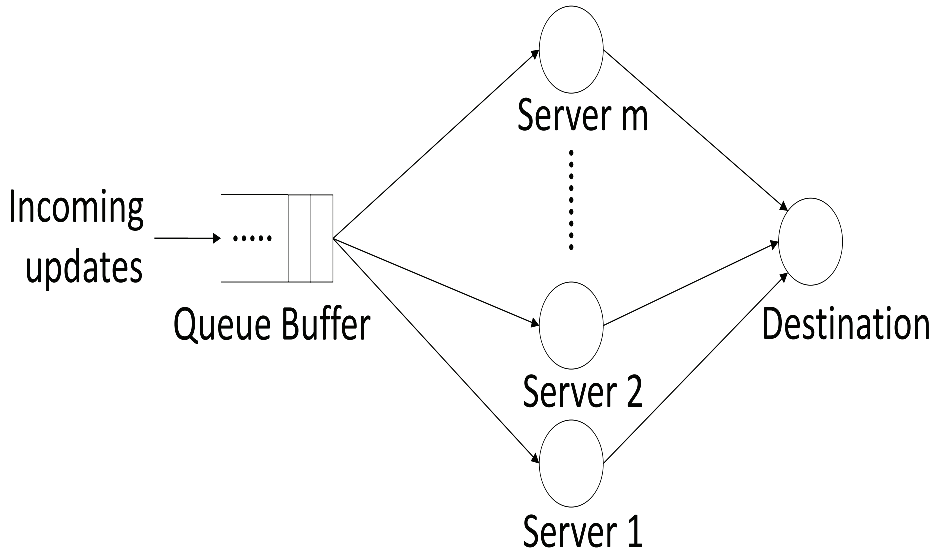
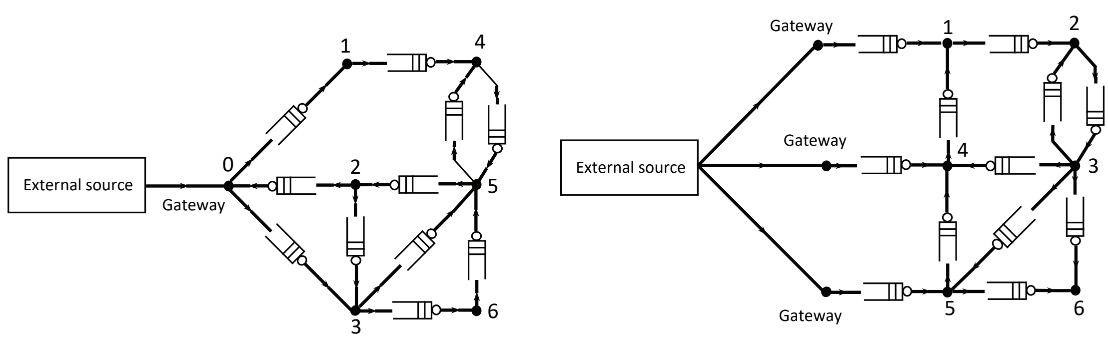


Figure 1.2: Single-source information update system.

in multi-channel communication systems [11], or a computer in parallel computing systems [12]. 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 [13–15] is considered in this network. 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. 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. We show that if the service times are *i.i.d.* exponentially distributed, the preemptive LGFS policy is 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, but also for minimizing the age stochastic process and any non-decreasing functional of the age stochastic process. The details of this part are presented in Chapter 2.

1.1.2 Multihop Networks



(a) Single-gateway model (b) Multiple-gateway model

Figure 1.3: Information updates in single-source multihop networks.

Now, we answer the above questions for a multihop network represented by a directed graph, as shown in Fig. 1.1.2, where the update packets are generated at an external source and are then dispersed throughout the network via one or multiple gateway nodes. The case of multiple gateway nodes is motivated by news spreading

Preemption type	**Transmission time distribution**	**Network topology**	**Policy Space**	**Proposed policy**	**Optimality result**
Preemption is allowed	Exponential	General	Causal policies	Preemptive LGFS	Age-optimal
Preemption is allowed	New-Better-than-Used	Each node has only one incoming link	Causal policies	Non-preemptive LGFS	Near age-optimal
Preemption is not allowed	Arbitrary	General	Work-conserving causal policies	Non-preemptive LGFS	Age-optimal

Table 1.1: Summary of age-optimality results in multihop networks.

in social media where news is usually posted by multiple social accounts or webpages. Moreover, we suppose that the packet generation times at the external source and the packet arrival times at the gateway node (gateway nodes) are arbitrary. This is because, in some applications, such as sensor and environment monitoring networks, the arrival process is not necessarily Poisson. For example, if a sensor observes an environmental phenomenon and sends an update packet whenever a change occurs, the arrival process of these update packets does not follow a Poisson process. The packet transmission times are independent but not necessarily identically distributed across the links, and *i.i.d.* across time. Interestingly, we find that some low-complexity scheduling policies can achieve (near) age-optimal performance in this setting. The main results in multihop networks are summarized in Table 1.1 and the details are presented in Chapter 3.

1.2 Information Update Systems with Controlled packet generation

There have been two major lines of research on age in single-source networks: One direction is on systems with a stochastic arrival process (as we discussed earlier). The second direction is for the case that the packet arrivals follow a sampling schedule that needs to be controlled to minimize AoI. We further investigate the latter case and develop simple algorithms to optimize data freshness. Towards that end, we consider the following information update systems:

1.2.1 Multi-source Information Update Systems

An important problem is how to jointly optimize packet generation and transmissions to maximize data freshness in single source networks [16–19]. In this dissertation, we consider a non-trivial generalization of [18, 19] and develop optimal sampling and scheduling strategies for multi-source networks.

We consider random, yet discrete, transmission times such that a packet has to be processed for a random period before delivered to the destination. In practice, such random transmission times occur in many applications, such as autonomous vehicles. In particular, there are many electronic control units (ECUs) in a vehicle, that are connected to one or more sensors and actuators via a controller area network (CAN) bus [20, 21]. These ECUs are given different priority, based on their criticality level (e.g., ECUs in the powertrain have a higher priority compared to those connected to infotainment systems). Since high priority packets usually have hard deadlines, the transmissions of low priority packets are interrupted whenever the higher priority ones are transmitted. Therefore, information packets with lower priority see a time-varying bandwidth, and hence encounter a random transmission time.

Another example is the wireless sensor networks that are used for environmental monitoring, human-related activities, etc. In such networks, sensor nodes may be deployed in remote areas and information is gathered from these sensors by an access point (AP) through a shared wireless channel [22]. Since this channel is influenced by uncertain factors, the channel delay varies with time.

When the transmission time is highly random, one can observe an interesting

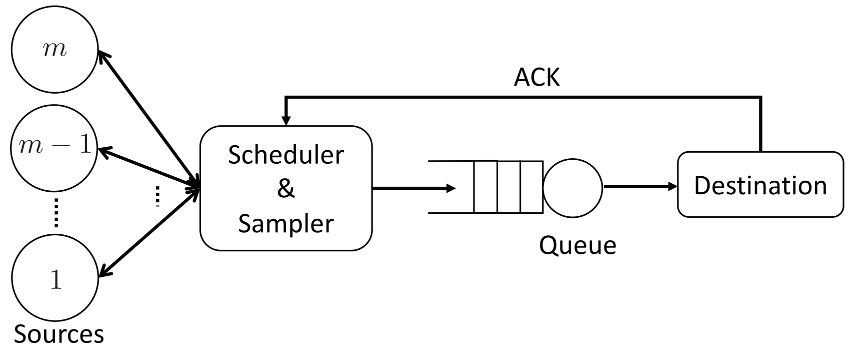


Figure 1.4: Sampling and scheduling in a multi-source network.

phenomenon: It is not necessarily optimal to generate a new packet as soon as the channel becomes available. This phenomenon was revealed in [16] and further explored in [18] and [19]. In the case of autonomous vehicles, many sensors may share the same CAN bus. As a result, the decision maker needs to control both the sampling times and service order of these sensors. The same observations are also applied to wireless sensor networks.

This motivates us to investigate timely status updates in multi-source systems with random transmission times, as depicted in Fig. 1.4. Sources take turns to generate update packets, and forward the packets to their destinations one-by-one through a shared channel with random delay. This results in a joint design problem of scheduling and sampling, where the scheduler chooses the update order of the sources, and the sampler determines when a source should generate a new packet in its turn. We find that it is optimal to first serve the source with the highest age, and, similar to the single-user case, it is not always optimal to generate packets as soon as the channel becomes available. In Chapter 4, we provide details about the challenges we face in this joint optimization problem and the methodology we use to overcome them.

1.2.2 Low-Power Information Update Systems

In a variety of information update systems, energy consumption is also a critical concern. For example, wireless sensor networks are used for monitoring crucial natural and human-related activities, e.g. forest fires, earthquakes, tsunamis, etc. Since such applications often require the deployment of sensor nodes in remote or hard-to-reach areas, they need to be able to operate unattended for long durations. Likewise, in medical sensor networks, battery replacement/recharging involves a series of medical procedures, leading to disutility to patients. Hence, energy consumption must be constrained in order to support a long battery life of 10-15 years [23][1]. For networks serving such real-time applications, prolonging battery-life is crucial. Existing works on multi-source networks, e.g., [26–36], focused exclusively on minimizing the AoI and overlooked the need to reduce power consumption. This motivates us to derive scheduling algorithms that achieve a trade-off between the competing tasks of minimizing AoI and reducing the energy consumption in multi-source networks.

Additionally, some status-update systems consist of a large number (e.g., hundreds of thousands) of densely packed wireless nodes, which are serviced by a single access point (AP). Examples include massive machine-type communications [37]. The dataloads in such "dense networks" [37, 38] are created by applications such as home security and automation, oilfield and pipeline monitoring, smart agriculture, animal and livestock tracking, etc. This introduces high variability in the data packet sizes so that the transmission times of data packets are random. Thus, scheduling algorithms designed for time-slotted systems with a fixed transmission duration, are not applicable to these systems. Besides that, synchronized scheduler for time-slotted systems

[1] The computations performed in [23] are based on the specifications of commercially used devices. For example, the used transceiver is 2.4 GHz chipset from Chipcon, the CC2420 [24], and the used microcontroller is the Motorola 8-bit microcontroller MC9508RE8 [25]. For more detail about the supply voltage and current consumption, please see the aforementioned references.

are feasible when there are relatively few sources and each source has sufficient energy. However, if there are a huge number of sources, the signaling overhead could be quite high. Since, each source may have limited energy and low traffic rate, the system could be highly inefficient. This motivates us to design asynchronized medium access protocols that coordinate the transmissions of multiple conflicting transmitters connected to a single AP.

Towards that end, we consider a wireless network with M sources that contend for channel access and communicate their update packets to an AP. Each source is equipped with a battery that may get charged by a renewable source of energy, e.g., solar. Moreover, each source utilizes carrier sensing to reduce collisions and adopt an asynchronized sleep-wake scheme [39] under which the source generates and transmits a packet if the channel is idle; and sleeps if either: (i) The channel is busy, (ii) it has completed a packet transmission. This enables each source to save the precious battery energy by switching off when it is unlikely to gain channel access for packet transmissions. However, since a source cannot transmit during the sleep period, this causes the AoI to increase. We carefully design the sleep-wake parameters to minimize the weighted-sum peak age of the sources, while ensuring that the battery lifetime constraint of each source is satisfied. When the sensing time (i.e., the time duration of carrier sensing) is zero, this sleep-wake design problem can be solved by resorting to a two-layer nested convex optimization procedure; however, for positive sensing times, the problem is non-convex. In Chapter 5, we devise a low-complexity solution to solve this problem and prove that, for practical sensing times that are short and positive, the solution is within a small gap from the optimum AoI performance.

1.3 Thesis Organization

The thesis is organized as follows: In Chapter 2, we develop low-complexity scheduling policies that can achieve (near) age-optimality in single-source single-hop networks for a variety of transmission time distributions including (i) exponential distribution, (ii) NBU distribution. We then extend these optimality results to multihop networks in Chapter 3. In Chapter 4, we consider the problem of how to jointly design the scheduler and sampler in multi-source networks. In Chapter 5, we design asynchronous sleep-wake scheduling algorithms that can optimize the information freshness while simultaneously meeting the energy requirements of battery operated information sources. Finally, concluding remarks and possible future research directions are presented in Chapter 6.

CHAPTER 2

SINGLE-HOP INFORMATION UPDATE SYSTEM WITH STOCHASTIC ARRIVALS

2.1 Introduction

A series of works studied the age performance of scheduling policies in a single queueing system with Poisson arrival process and exponential service time [4, 5, 7–9, 40, 41]. In [4,5], the update frequency was optimized to improve data freshness in First-Come, First-Serve (FCFS) information-update systems. The effect of the packet management on the age was considered in [7–9]. 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 [40], 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 [41]. The analysis in [41] 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 [4, 5, 7–9, 40, 41] is whether the preemptive LCFS policy is age-optimal for exponential service times. In this chapter, 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 [42], 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.

Characterizing the age for a class of packet service time distributions that are more general than exponential was considered in [6, 10, 43]. In [6], 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 [43]. Gamma-distributed service times was considered in [10]. The studies in [10, 43] were carried out for LCFS queueing systems with and without preemption.

In this chapter, 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. Our key contributions in this chapter can be summarized as follows:

- If the packet service times are *i.i.d.* exponentially distributed, then for *arbitrary* system parameters (including *arbitrary* packet generation times, packet arrival times, number of servers, maximum replication degree[1], and buffer size), we prove that the preemptive LGFS with replication (prmp-LGFS-R) policy minimizes the age stochastic process and any non-decreasing functional

[1]Maximum replication degree is the maximum number of replicas that can be created for a packet.

of the age stochastic process among all policies in a stochastic ordering sense (Theorem 2.3.1). Note that this age penalty model is very general. Many age penalty metrics studied in the literature, such as the time-average age [4, 5, 7–10, 16, 17, 40–42], and time-average age penalty function [18, 44], 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 LGFS 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 2.3.2). Note that policy non-prmp-LGFS-R with a given maximum replication degree can be near age-optimal compared with policies with any maximum replication degree. This result was not anticipated: In [15,45,46], 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 [40], and the non-prmp-LGFS-R when the buffer size is one reduces to LCFS with preemption only in waiting for a single-source case in [40], or the "M/M/1/2*" in [7, 8]. 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 2.4.1). 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 2.4.2).

In a comparison with the existing works, our study stands as follows: For exponential service times, while [42] considered the case where 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. In addition, for NBU service time distributions, our study complements the age analysis results in [6, 10, 43] by showing that non-preemptive LGFS (and its special case non-preemptive LCFS) policies are near age-optimal. 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 [10, 43] is a special case of NBU service time distributions.

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 study that considers packet replication to minimize the age.

2.2 Model and Formulation

This section describes a single-source single-hop network model as shown in Fig. 1.2, and introduces necessary preliminary propositions, notations and definitions.

2.2.1 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 2.2.1. ***Univariate Stochastic Ordering:*** *[47] 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 2.2.2. ***Multivariate Stochastic Ordering:*** *[47] 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 2.2.3. ***Stochastic Ordering of Stochastic Processes:*** *[47] 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*

$$(X(t_1), X(t_2), \ldots, X(t_n)) \leq_{st} (Y(t_1), Y(t_2), \ldots, Y(t_n)), \tag{2.2.1}$$

where the multivariate stochastic ordering in (2.2.1) *was defined in Definition 2.2.2.*

2.2.2 Preliminary Propositions

The following propositions will be used throughout this chapter:

Proposition 2.2.1 ([47], 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*

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

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

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

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

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

Proposition 2.2.2 ([47], 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 2.2.3 ([47], 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 2.2.4 ([47], 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 2.2.5 ([47], 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

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

2.2.3 Queueing System Model

We consider a queueing system with m servers as shown in Fig. 1.2. 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 chapter, 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. 2.1, 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.

2.2.4 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 [2]. In contrast, in a **non-preemptive** policy, processing of a packet cannot be interrupted until the packet is completed or cancelled [3]; 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.

2.2.5 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 [1–4]

$$\Delta(t) = t - U(t). \tag{2.2.2}$$

[2] 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.

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

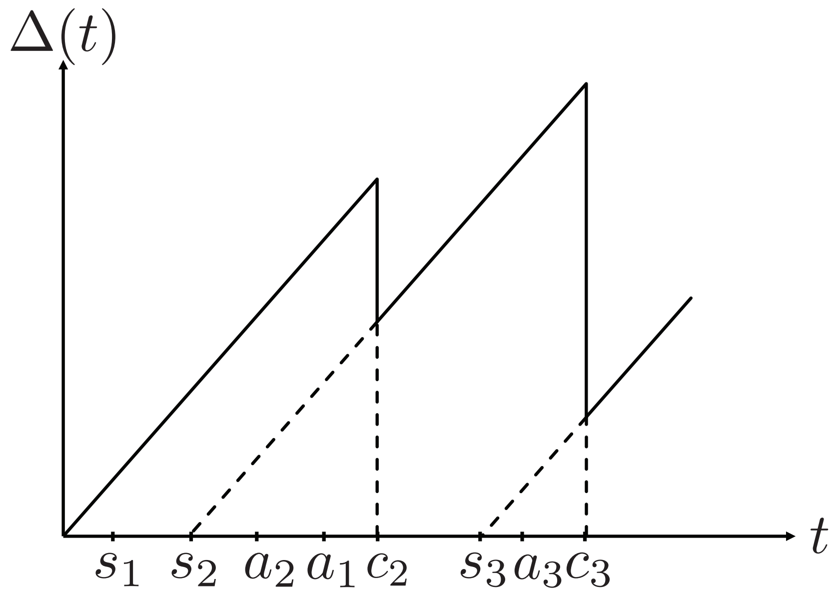


Figure 2.1: A sample path of the age process $\Delta(t)$.

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. 2.1, 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

$$\Delta = \{\Delta(t), t \in [0, \infty)\}. \tag{2.2.3}$$

In this chapter, 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 2.2.4. ***Age Penalty Functional:*** *Let* $\mathbf{V}$ *be the set of* n*-dimensional Lebesgue measurable functions, i.e.,*

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

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:*

$$g(\Delta_1) \leq g(\Delta_2), \; \textit{whenever}\, \Delta_1(t) \leq \Delta_2(t), \forall t \in [0, \infty). \tag{2.2.4}$$

The age penalty functionals used in prior studies include:

- *Time-average age [4, 5, 7–10, 16, 17, 40–42]:* The time-average age is defined as

$$g_1(\Delta) = \frac{1}{T} \int_0^T \Delta(t) dt, \tag{2.2.5}$$

- *Time-average age penalty function [18, 44]:* The average age penalty function is

$$g_3(\Delta) = \frac{1}{T} \int_0^T h(\Delta(t)) dt, \tag{2.2.6}$$

  where h: $[0, \infty) \to [0, \infty)$ can be any non-negative and non-decreasing function. As pointed out in [18], 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.

2.3 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.

2.3.1 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 2.3.1. *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 chapter, 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 [4] are depicted in Algorithm 2.1.

[4] The decision related to dropping or replacing packets in the full buffer case does not affect the

Algorithm 2.1: Prmp-LGFS-R policy when $r = 1$.

```
α := 0;                  // α is the smallest generation time of the packets under service
I := m;                  // I is the number of idle servers
Q := ∅;                  // 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 ≤ α then                    // Packet p_i is stale
                Store the packet in the queue;
            else                             // Packet p_i carries fresh information
                Find packet p_j ∈ Q with generation time α;
                Preempt packet p_j and store it back to the queue;
                Assign packet p_i to the idle server;
                Q := Q ∪ {p_i} − {p_j};
        else                                 // At least one of the servers is idle
            Assign packet p_i to an idle server;
            Q := Q ∪ {p_i};
        Update I;
        α := min{s_i : i ∈ Q};
    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 ∪ {p_h};
        Update I;
        α := min{s_i : i ∈ Q};
```

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

age performance of prmp-LGFS-R policy. Hence, we don't specify this decision under the prmp-LGFS-R policy in all related algorithms.

service may 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 2.2: 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.

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 2.3.1. *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 $\pi \in \Pi_r$*

$$[\Delta_{\textit{prmp-LGFS-R}} | \mathcal{I}] \leq_{st} [\Delta_\pi | \mathcal{I}], \tag{2.3.1}$$

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

$$\mathbb{E}[g(\Delta_{\textit{prmp-LGFS-R}}) | \mathcal{I}] = \min_{\pi \in \Pi_r} \mathbb{E}[g(\Delta_\pi) | \mathcal{I}], \tag{2.3.2}$$

provided the expectations in (2.3.2) *exist.*

Proof. See Appendix A.1. □

Theorem 2.3.1 tells us that for arbitrary sequence of packet generation times

Algorithm 2.2: Prmp-LGFS-R policy when $r \geq 1$.

$\alpha := 0$; // α 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
$k := \lfloor \frac{m}{r} \rfloor$; // k is the number of distinct packets that each one of them can be replicated on r servers
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 packet p_i in the queue;
            else // Packet p_i carries fresh information
                Find packet $p_j \in Q$ with generation time α;
                Preempt all replicas of packet p_j;
                Packet p_j is stored back to the queue;
                $Q := Q \cup \{p_i\} - \{p_j\}$;
                Update I;
        else // At least one of the servers is idle
            $Q := Q \cup \{p_i\}$;
        $\alpha := \min\{s_i : i \in Q\}$;
        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
            Preempt $(r - I)$ replicas of the packet with generation time α;
            Replicate packet p_i on r idle servers;
        else if $p_i \in Q$ ***and*** *generation time of packet* $p_i = \alpha$ **then**
            Replicate packet p_i on $\min\{r, I\}$ idle servers;
        Update I;
    if *a packet p_l is delivered* **then**
        Cancel the remaining replicas of packet p_l;
        $Q := Q - \{p_l\}$;
        if *the queue is not empty* **then**
            Pick the packet with the largest generation time in the queue p_h;
            $Q := Q \cup \{p_h\}$;
            Replicate packet p_h on $\min\{r, I\}$ idle servers;
            Update I;
        $\alpha := \min\{s_i : i \in Q\}$;

$(s_1, s_2, \ldots)$, 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, (2.3.2) tells us that the prmp-LGFS-R policy minimizes any *non-decreasing functional* of the age process, including the time-average age (2.2.5) and time-average age penalty function (2.2.6) 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 [40]. Thus, this policy can achieve age-optimality in this case.

As a result of Theorem 2.3.1, we can deduce the following corollaries:

A weaker version of Theorem 2.3.1 can be obtained as follows.

Corollary 2.3.1.1. *If the conditions of Theorem 2.3.1 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 2.3.1 and Proposition 2.2.4, the corollary follows. □

Corollary 2.3.1.2. *Under the conditions of Theorem 2.3.1, 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 2.3.1. □

It is important to recall that $\Pi_1 \subset \Pi_2 \subset \ldots \subset \Pi_m$. Therefore, Corollary 2.3.1.2 tells us that the prmp-LGFS-R policy when $r = m$ achieves age-optimality compared with all policies with any maximum replication degree.

Corollary 2.3.1.3. *If the conditions of Theorem 2.3.1 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 2.3.1.4. *Under the conditions of Theorem 2.3.1, 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* (2.3.1) *and* (2.3.2).

Proof. This corollary is a direct result of Theorem 2.3.1. □

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 $\rho = \lambda/m\mu$. [5] The queue size is B, which is a non-negative integer.

Fig. 2.2 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

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

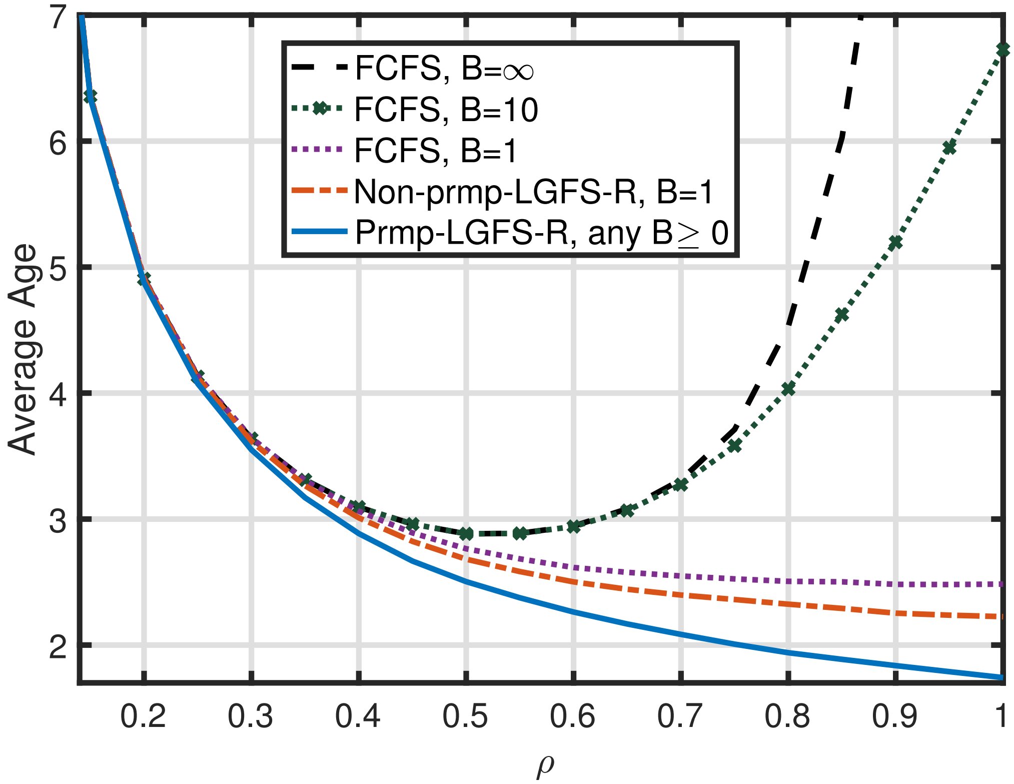


Figure 2.2: Average age versus traffic intensity ρ for an update system with $m = 1$ server, queue size B, and *i.i.d.* exponential service times.

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 [4], and the non-preemptive LGFS policy with queue size $B = 1$ which is equivalent to "M/M/1/2*" in [7, 8] 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.

2.3.2 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 2.3.2. ***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 $t, \tau \geq 0$*

$$\bar{F}(\tau + t) \leq \bar{F}(\tau)\bar{F}(t). \tag{2.3.3}$$

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 and full. The description of the non-prmp-LGFS-R policy is depicted in Algorithm 2.3: 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

Algorithm 2.3: Non-prmp-LGFS-R policy.

```
1  δ := 0;                 // δ is the smallest generation time of the packets in the queue
2  I := m;                 // I is the number of idle servers
3  k := ⌊m/r⌋;  // 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 > δ then                         // Packet p_i carries fresh information
9                      Packet p_i replaces the packet with generation time δ in the queue;
10                 else                                  // Packet p_i is stale
11                     Drop packet p_i;
12             else
13                 Store packet p_i in the queue;
14             Update δ;
15         else                                          // At least one of the servers is idle
16             Replicate packet p_i on min{r, I} idle servers;
17             Update I;
18     if a packet p_l is delivered then
19         Cancel the remaining replicas of packet p_l;
20         Update I;
21         Find packet p_j that is replicated on (m − kr) servers;
22         if the queue is empty and packet p_j exists then
23             Replicate packet p_j on extra ((k + 1)r − m) idle servers;
24         else if the queue is not empty then
25             Pick the packet with the largest generation time in the queue p_h;
26             if packet p_j exists and generation time of packet p_j > generation time of
                 packet p_h then
27                 Replicate packet p_j on extra ((k + 1)r − m) idle servers;
28                 Update I;
29             Replicate packet p_h on min{r, I} idle servers;
30         Update I;
31         Update δ;
```

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.

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.

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

$$\Delta_\pi^{\text{LB}}(t) = t - \max\{s_i : v_i(\pi) \leq t\}. \tag{2.3.4}$$

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

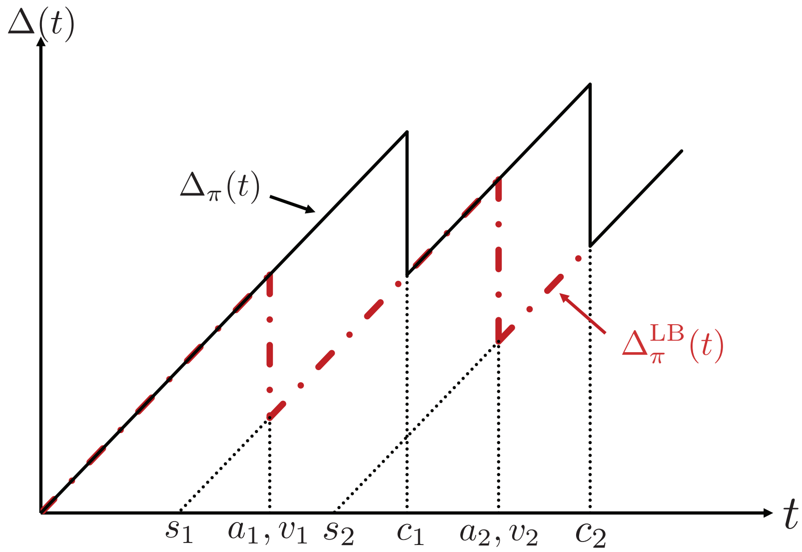


Figure 2.3: The evolution of $\Delta_\pi^{\rm LB}$ and Δ_π in a single server queue. We assume that $a_1 > s_1$ and $a_2 > c_1 > s_2$. Thus, we have $v_1 = a_1$ and $v_2 = a_2$.

process $\Delta_\pi^{\rm LB}(t)$ 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 $\Delta_\pi^{\rm LB}(t)$ 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. 2.3. The process $\Delta_{\text{non-prmp-LGFS-R}}^{\rm LB}$ is a lower bound of all policies in Π_m in the following sense.

Lemma 2.3.1. *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 $\pi \in \Pi_m$*

$$[\Delta_{\textit{non-prmp-LGFS-R}}^{LB}|\mathcal{I}] \leq_{st} [\Delta_\pi|\mathcal{I}]. \tag{2.3.5}$$

Proof. See Appendix A.2. □

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 2.3.1 to prove the following theorem.

Theorem 2.3.2. *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) We have

$$\min_{\pi \in \Pi_m} [\bar{\Delta}_\pi | \mathcal{I}] \leq [\bar{\Delta}_{\textit{non-prmp-LGFS-R}} | \mathcal{I}] \leq \min_{\pi \in \Pi_m} [\bar{\Delta}_\pi | \mathcal{I}] + \mathbb{E}[X]. \tag{2.3.6}$$

(b) If there is a positive integer a such that $m = ar$, then

$$\min_{\pi \in \Pi_m} [\bar{\Delta}_\pi | \mathcal{I}] \leq [\bar{\Delta}_{\textit{non-prmp-LGFS-R}} | \mathcal{I}] \leq \min_{\pi \in \Pi_m} [\bar{\Delta}_\pi | \mathcal{I}] + \mathbb{E}[\min_{l=1,\ldots,r} X_l], \tag{2.3.7}$$

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 A.3. □

Theorem 2.3.2 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 [40], or the "M/M/1/2*" in [7,8]. Thus, these policies can achieve near age-optimality in this case. The following corollary emphasizes this relationship.

Corollary 2.3.2.1. *Under the conditions of Theorem 2.3.2, 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* (2.3.6)*.*

Proof. This corollary is a direct result of Theorem 2.3.2. □

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.

Fig. 2.4 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 2.3.1, 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 2.3.2. This is a surprising result since it was shown in [15, 45, 46] 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,

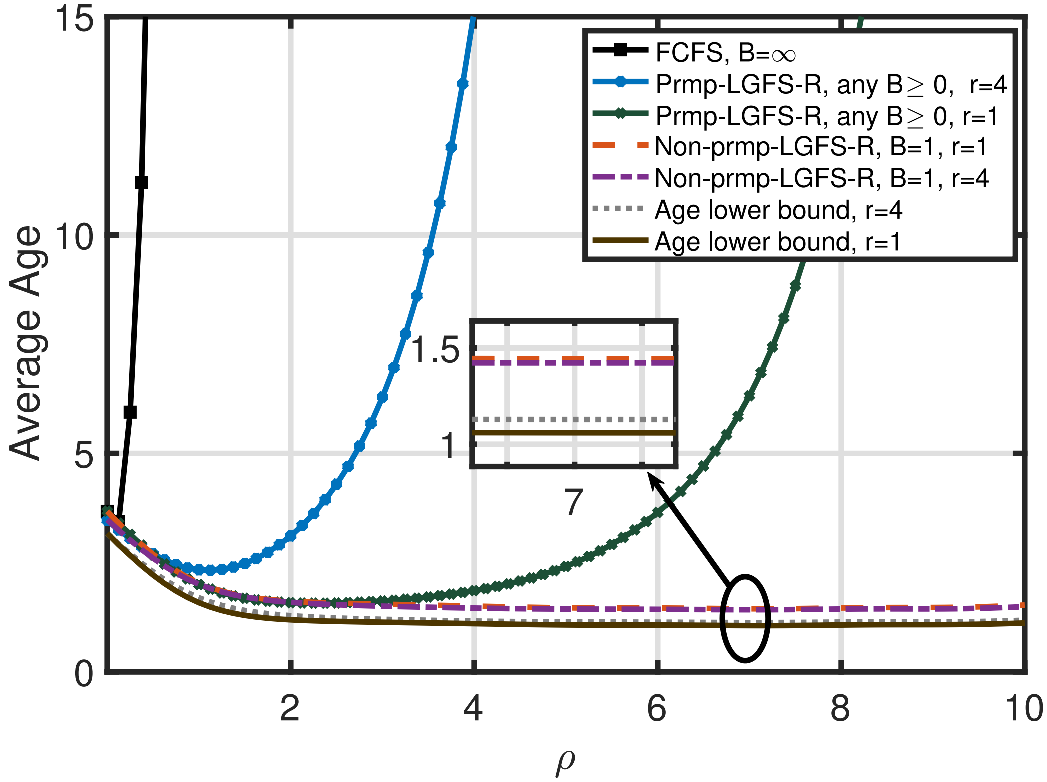


Figure 2.4: Average age versus traffic intensity ρ for an update system with $m = 4$ servers, queue size B, maximum replication degree r, and $i.i.d$ NBU service times.

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 [7].

Fig. 2.5 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

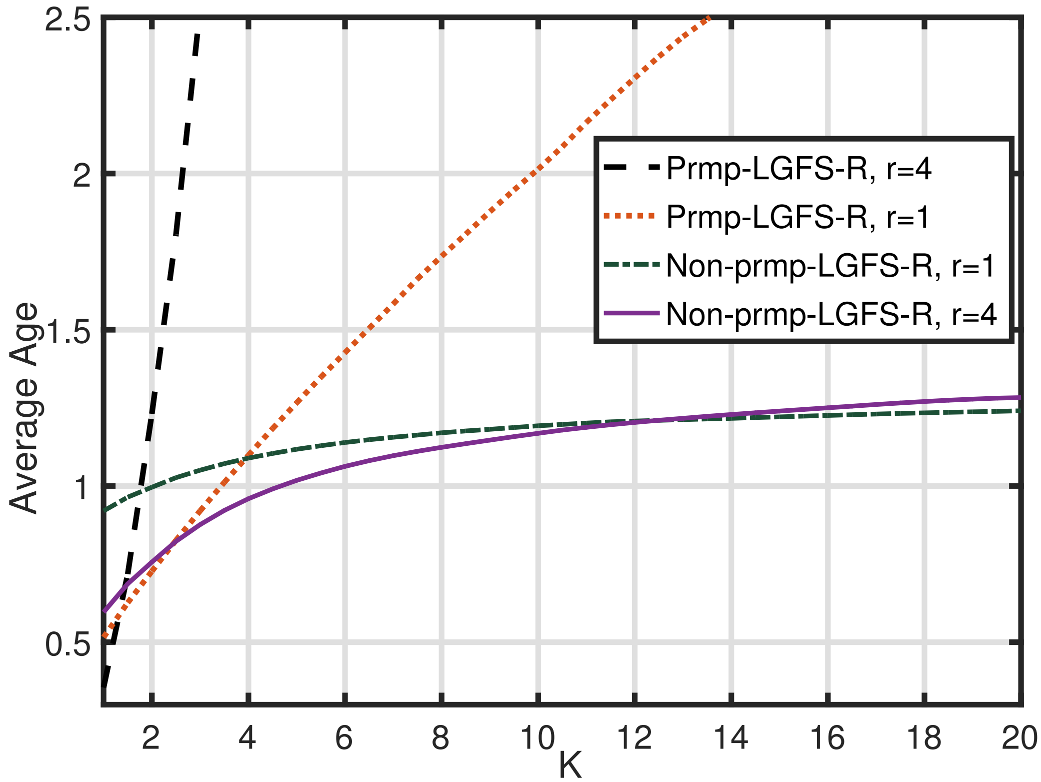


Figure 2.5: Average age under gamma service time distributions with different shape parameter K, where $m = 4$ servers, queue size $B = \infty$, and maximum replication degree r.

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 2.3.1 and Corollary 2.3.1.2. 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.

2.3.3 Discussion

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

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 2.3.1). 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. 2.5, 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.

2.5). 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. 2.5, 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 [10, 40]. For exponential service time distribution, Yates and Kaul showed in Theorem 3(a) of [40] 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 [10]. In particular, in [10, Numerical Results], the authors showed that the non-preemptive LCFS policy can achieve better average age than the preemptive LCFS policy. In this chapter, 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 [40] and [10].

Replication vs. Non-Replication

The replication technique has gained significant attention in recent years to reduce the delay in queueing systems [13–15]. However, it was shown in [15, 45, 46] 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 [47]

$$\frac{1}{\mathbb{E}[\min_{l=1,\ldots,m} X_l]} \leq \frac{m}{\mathbb{E}[X]}. \tag{2.3.8}$$

Now, if X_l represents the packet service time of server l, then the left-hand side of (2.3.8) represents the service rate when each packet is replicated to all servers; and the right-hand side of (2.3.8) 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

$$\mathbb{E}[\min_{l=1,\ldots,m} X_l] \leq \mathbb{E}[X]. \tag{2.3.9}$$

Thus, packet replication can help to reduce the age by exploiting the diversity provided by multiple servers. As shown in Fig. 2.5, 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 $K \leq 12.5$.

2.4 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 2.4.1. ***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

$$D_{\text{avg}}(\pi) = \frac{1}{n}\sum_{i=1}^{n}[c_i(\pi) - a_i], \tag{2.4.1}$$

where the delay of packet i under policy π is $c_i(\pi) - a_i$.[6]

Definition 2.4.2. ***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.

2.4.1 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:

[6] The $\limsup$ operator is enforced on the right hand side of (2.4.1) if $n \to \infty$.

Theorem 2.4.1. *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 2.4.1. 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 A.4. □

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.

2.4.2 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 2.4.2. *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 2.4.2, because it is similar to that of Theorem 2.4.1.

2.5 Conclusions

In this chapter, 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 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.

CHAPTER 3

MULTIHOP INFORMATION UPDATE SYSTEM WITH STOCHASTIC ARRIVALS

3.1 Introduction

The demand for real-time information updates in multihop networks, such as the IoT, intelligent transportation systems, and sensor networks, has gained increasing attention recently. In intelligent transportation systems [48–50], for example, a vehicle shares its information related to traffic congestion and road conditions to avoid collisions and reduce congestion. Thus, in such applications, maintaining the age at a low level at all network nodes is a crucial requirement. In some other information update applications, such as emergency alerts and sensor networks, critical information is needed to report in a timely manner, and the energy consumption of the sensor nodes must be sufficiently low to support a long battery life up to 10-15 years [23]. Because of the low traffic load in these systems, wireless interference is not the limiting factor, but rather battery life through energy consumption is. Furthermore, information updates over the Internet, cloud systems, and social networks are of significant importance. These systems are built on wireline networks or implemented based on transport layer APIs. Motivated by these applications, we investigate information updates over multihop networks that can be modeled as multihop queueing systems.

It has been observed in early studies on age of information analysis [7–10, 51] that Last-Come, First-Serve (LCFS)-type of scheduling policies can achieve a lower age than other policies. The optimality of the LCFS policy, or more generally the Last-Generated, First-Served (LGFS) policy, for minimizing the age of information in single-hop networks was established in Chapter 2. However, age-optimal scheduling in multihop networks remains an important open question.

There have been a few recent studies on the age of information in multihop networks [52–59]. The age is analyzed for specific network topologies, e.g., line or star networks, in [52]. In [53], an offline optimal sampling policy was developed to minimize the age in two-hop networks with an energy-harvesting source. A congestion control mechanism that enables timely delivery of the update packets over IP networks was considered in [54]. In [55], the author analyzed the average age in a multihop line network with Poisson arrival process and exponential service times. This analysis was later extended in [56] to include age moments and distributions. The study in this chapter and [55, 56] complement each other in the following sense: Our results (i.e., Theorem 3.3.1) show that the LCFS policy with preemption in service is age-optimal. However, we do not characterize the achieved optimal age, which was evaluated in [55, 56]. The authors of [57] addressed the problem of scheduling in wireless multihop networks with general interference model and multiple flows, assuming that all network queues are adopting an FCFS policy. A similar network model was considered in [58], where the optimal update policy was obtained for the "active sources scenario". In this scenario, each source can generate a packet at any time, and hence, each source always has a fresh packet to send. The active sources scenario in multihop networks was also considered in [59], where nodes take turns broadcasting their updates, and hence each node can act either as a source or a relay. In contrast to our study, the works in [57–59] considered a time-slotted system, where a packet is transmitted from one node to another in one time slot.

In this chapter, we minimize the age of a single information flow in interference-free multihop networks. We develop simple scheduling policies that can achieve age-optimality or near age-optimality in these networks. The following summarizes our main contributions in this chapter:

- If preemption is allowed and the packet transmission times over the network links are exponentially distributed, we prove that the preemptive LGFS policy minimizes the age processes at all nodes in the network among all causal policies in a stochastic ordering sense (Theorem 3.3.1). In other words, the preemptive LGFS policy minimizes any *non-decreasing functional of the age processes at all nodes* in a stochastic ordering sense. Note that the non-decreasing functional of the age processes at all nodes represents a very general class of age metrics in that it includes many age penalty metrics studied in the literature, such as the time-average age [4, 5, 7–10, 16, 17, 40], non-linear age functions [18, 44], and age penalty functional at single-hop network in Chapter 2.

- Although the preemptive LGFS policy can achieve age-optimality for exponential transmission times, it does not always minimize the age processes for non-exponential transmission times. When preemption is allowed, we investigate an important class of packet transmission time distributions called New-Better-than-Used (NBU) distributions, which are more general than exponential. The network topology we consider here is more restrictive in the sense that each node has one incoming link only. We show that the non-preemptive LGFS policy is within a constant age gap from the optimum average age, and that the gap is independent of the packet generation and arrival times, and buffer sizes (Theorem 3.3.2). Our numerical result (Fig. 3.3) shows that the preemptive LGFS policy can be very far from the optimum for non-exponential transmission times, while the non-preemptive LGFS policy is near age-optimal.

- If preemption is not allowed, then for arbitrary distributions of packet transmission times, we prove that the non-preemptive LGFS policy minimizes the age processes at all nodes among all work-conserving policies in the sense of stochastic ordering (Theorem 3.3.3). Age-optimality here can be achieved even if the transmission time distribution differs from one link to another, i.e., the transmission time distributions are heterogeneous.

To the best of our knowledge, these are the first optimal results on minimizing the age of information in multihop queueing networks with arbitrary packet generation and arrival processes.

3.2 Model and Formulation

This section describes a single-source multihop network model as shown in Fig. 1.1.2. Note that throughout this chapter, we will use the definitions and preliminary propositions provided in Chapter 2, Section 2.2.1.

3.2.1 Network Model

We consider a multihop network represented by a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{L}$ is the set of links, as shown in Fig. 1.1.2 [1]. The number of nodes in the network is $|\mathcal{V}| = N$. The nodes are indexed from 0 to $N-1$, where node 0 acts as a gateway node. Define $(i, j) \in \mathcal{L}$ as a link from node i to node j, where i is the origin node and j is the destination node. We assume that the links in the network can be active simultaneously, which holds in the applications mentioned in Section 3.1. The packet transmission times are independent but not necessarily

[1] For the simplicity of presentation, we focus on the network model with a single gateway node in the rest of the chapter. However, it is not hard to see that our results also hold for networks with multiple gateway nodes.

identically distributed across the links, and *i.i.d.* across time. As will be clear later on, we consider the following transmission time distributions: Exponential distribution, NBU distributions, and arbitrary distribution. In addition, we consider two types of network topology: general network topology and special network topology in which each node has one incoming link. We note that this special network topology is an extension of tandem queues. These different network settings are summarized in Table 1.1.

The system starts to operate at time $t = 0$. The update packets are generated at an external source, and are firstly forwarded to node 0, from which they are dispersed throughout the network. Thus, the update packets may arrive at node 0 some time after they are generated. The l-th update packet, called packet l, is generated at time s_l, arrives at node 0 at time a_{l0}, and is delivered to any other node j at time a_{lj} such that $0 \leq s_1 \leq s_2 \leq \ldots$ and $s_l \leq a_{l0} \leq a_{lj}$ for all $j = 1, \ldots, N-1$. Note that in this chapter, the sequences $\{s_1, s_2, \ldots\}$ and $\{a_{10}, a_{20}, \ldots\}$ are *arbitrary*. Hence, the update packets may not arrive at node 0 in the order of their generation times. For example, packet $l+1$ may arrive at node 0 earlier than packet l such that $s_l \leq s_{l+1}$ but $a_{l0} \geq a_{(l+1)0}$. We suppose that once a packet arrives at node i, it is immediately available to all the outgoing links from node i. Moreover, the update packets are time-stamped with their generation times such that each node knows the generation times of its received packets. Each link (i, j) has a queue of buffer size B_{ij} to store the incoming packets, which can be infinite, finite, or even zero. If a link has a finite queue buffer size, then the packet that arrives to a full buffer either is dropped or replaces another packet in the queue.

3.2.2 Scheduling Policy

We let π denote a scheduling policy that determines the following (at each link): i) Packet assignments to the server, ii) packet preemption if preemption is allowed, iii) packet droppings and replacements when the queue buffer is full. The sequences of packet generation times $\{s_1, s_2, \ldots\}$ and packet arrival times $\{a_{10}, a_{20}, \ldots\}$ at node 0 do not change according to the scheduling policy, while the packet arrival times at other nodes (i.e., a_{lj} for all l and $j = 1, \ldots, N-1$) are functions of the scheduling policy π. We suppose that the packet transmission times over the links are invariant of the scheduling policy and the realization of a packet transmission time at any link is unknown until its transmission over this link is completed (unless the transmission time is deterministic).

Let Π denote the set of all *causal* policies, in which scheduling decisions are made based on the history and current information of the system (system information includes the location, arrival times, and generation times of all the packets in the system, and the idle/busy state of all the servers). we define several types of policies in Π:

A policy is said to be **preemptive**, if a link can switch to send another packet at any time; the preempted packets can be stored back into the queue if there is enough buffer space and sent out at a later time when the link is available again. In contrast, in a **non-preemptive** policy, a link must complete sending the current packet before starting to send another packet. A policy is said to be **work-conserving**, if each link is busy whenever there are packets waiting in the link's queue.

3.2.3 Age Performance Metric

Let $U_j(t) = \max\{s_l : a_{lj} \leq t\}$ be the generation time of the freshest packet arrived at node j before time t. The *age of information*, or simply the *age*, at node j is defined

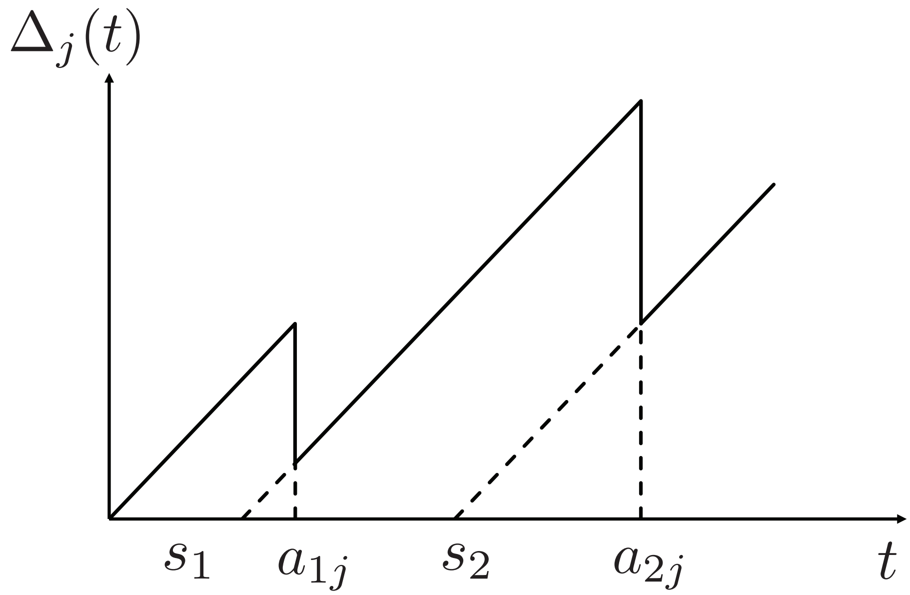


Figure 3.1: A sample path of the age process $\Delta_j(t)$ at node j.

as [1–4]

$$\Delta_j(t) = t - U_j(t). \tag{3.2.1}$$

The process of $\Delta_j(t)$ is given by $\Delta_j = \{\Delta_j(t), t \in [0, \infty)\}$. The initial state of $U_j(t)$ at time $t = 0^-$ is invariant of the scheduling policy $\pi \in \Pi$, where we assume that $U_j(0^-) = 0 = s_0$ for all $j \in \mathcal{V}$. As shown in Fig. 3.1, the age increases linearly with t but is reset to a smaller value with the arrival of a fresher packet. The age vector of all the network nodes at time t is

$$\boldsymbol{\Delta}(t) =(\Delta_0(t), \Delta_1(t), \ldots, \Delta_{N-1}(t)). \tag{3.2.2}$$

The age process of all the network nodes is given by

$$\boldsymbol{\Delta} = \{\boldsymbol{\Delta}(t), t \in [0, \infty)\}. \tag{3.2.3}$$

In this chapter, we introduce a general *age penalty functional* $g(\boldsymbol{\Delta})$ to represent the level of dissatisfaction for data staleness at all the network nodes.

Definition 3.2.1. ***Age Penalty Functional:*** *Let* $\mathbf{V}$ *be the set of* n*-dimensional functions, i.e.,*

$$\mathbf{V} = \{f : [0, \infty)^n \mapsto \mathbb{R}\}.$$

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:*

$$g(\boldsymbol{\Delta}_1) \leq g(\boldsymbol{\Delta}_2), \ \textit{whenever}\ \boldsymbol{\Delta}_1(t) \leq \boldsymbol{\Delta}_2(t), \forall t \in [0, \infty). \tag{3.2.4}$$

The age penalty functionals used in prior studies include:

- *Time-average age [4, 5, 7–10, 16, 17, 40]:* The time-average age of node j is defined as

$$g_1(\boldsymbol{\Delta}) = \frac{1}{T}\int_0^T \Delta_j(t)dt, \tag{3.2.5}$$

- *Non-linear age functions [18, 44]:* The non-linear age function of node j is in the following form

$$g_3(\boldsymbol{\Delta}) = \frac{1}{T}\int_0^T h(\Delta_j(t))dt, \tag{3.2.6}$$

where $h : [0, \infty) \to [0, \infty)$ can be any non-negative and non-decreasing function. As pointed out in [18], 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 desire for data refreshing grows 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.

- *Age penalty functional in single-hop networks in Chapter 2*: The age penalty functional in Chapter 2 is a non-decreasing functional of the age process at one node, which is a special case of that defined in Definition 3.2.1 with $n = 1$.

3.3 Main Results

In this section, we present our (near) age-optimality results for multihop networks. We prove our results using stochastic ordering.

3.3.1 Exponential Transmission Times, Preemption is Allowed

We study age-optimal packet scheduling for networks that allow for preemption and the packet transmission times are exponentially distributed, *independent* across the links and *i.i.d.* across time[2]. We consider a LGFS scheduling principle which is defined as follows.

[2] Although we consider exponential transmission times, packet transmission time distributions are not necessarily identical over the network links, i.e., different links may have different mean transmission times.

Algorithm 3.1: Preemptive Last-Generated, First-Served policy at the link (i, j).

$\alpha_{ij} := 0$; // α_{ij} is the generation time of the packet being transmitted on the link (i, j)
while *the system is ON* **do**
  if *a new packet with generation time s arrives to node i* **then**
    if *the link (i, j) is busy* **then**
      if $s \leq \alpha_{ij}$ **then**
        Store the packet in the queue;
      else // The packet carries fresher information than the packet being transmitted.
        Send the packet over the link by preempting the packet being transmitted;
        The preempted packet is stored back to the queue;
        $\alpha_{ij} = s$;
    else // The link is idle.
      The new packet is sent over the link;
  if *a packet is delivered to node j* **then**
    if *the queue is not empty* **then**
      The freshest packet in the queue is sent over the link;

Definition 3.3.1. *A scheduling policy is said to follow the **Last-Generated, First-Served** discipline, if the last generated packet is sent first among all packets in the queue.*

We consider a preemptive LGFS (prmp-LGFS) policy at each link $(i, j) \in \mathcal{L}$. The implementation details of this policy are depicted in Algorithm 3.1[3].

Define a set of parameters $\mathcal{I} = \{\mathcal{G}(\mathcal{V}, \mathcal{L}), (B_{ij}, (i, j) \in \mathcal{L}),\ s_l, a_{l0}, l = 1, 2, \ldots\}$, where $\mathcal{G}(\mathcal{V}, \mathcal{L})$ is the network graph, B_{ij} is the queue buffer size of link (i, j), s_l is the generation time of packet l, and a_{l0} is the arrival time of packet l to node 0. Let $\boldsymbol{\Delta}_\pi$

[3] The decision related to packet droppings and replacements in full buffer case (at any link) doesn't affect the age performance of prmp-LGFS policy. Hence, we don't specify this decision under the prmp-LGFS policy.

be the age processes of all nodes in the network under policy π. The age optimality of prmp-LGFS policy is provided in the following theorem.

Theorem 3.3.1. *If the packet transmission times are exponentially distributed, independent across links and i.i.d. across time, then for all $\mathcal{I}$ and $\pi \in \Pi$*

$$[\boldsymbol{\Delta}_{prmp\text{-}LGFS}|\mathcal{I}] \leq_{st} [\boldsymbol{\Delta}_{\pi}|\mathcal{I}], \tag{3.3.1}$$

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

$$\mathbb{E}[g(\boldsymbol{\Delta}_{prmp\text{-}LGFS})|\mathcal{I}] = \min_{\pi \in \Pi} \mathbb{E}[g(\boldsymbol{\Delta}_{\pi})|\mathcal{I}], \tag{3.3.2}$$

provided the expectations in (3.3.2) *exist.*

Proof sketch. We use a coupling and forward induction to prove it. We first consider the comparison between the preemptive LGFS policy and any arbitrary policy π. We couple the packet departure processes at each link of the network such that they are identical under both policies. Then, we use the forward induction over the packet delivery events at each link (using Lemma B.1.2) and the packet arrival events at node 0 (using Lemma B.1.3) to show that the generation times of the freshest packets at each node of the network are maximized under the preemptive LGFS policy. By this, the preemptive LGFS policy is age-optimal among all causal policies. For more details, see Appendix B.1. □

Theorem 3.3.1 tells us that for arbitrary sequence of packet generation times $\{s_1, s_2, \ldots\}$, sequence of arrival times $\{a_{10}, a_{20}, \ldots\}$ at node 0, network topology $\mathcal{G}(\mathcal{V}, \mathcal{L})$, and buffer sizes $(B_{ij}, (i,j) \in \mathcal{L})$, the prmp-LGFS policy achieves optimality of the joint distribution of the age processes at the network nodes within the policy space Π. In addition, (3.3.2) tells us that the prmp-LGFS policy minimizes any non-decreasing age penalty functional g, including the time-average age (3.2.5), and non-linear age functions (3.2.6).

As we mentioned before, the result of Theorem 3.3.1 still holds for the multiple-gateway model shown in Fig. 1.3(b). In particular, Lemma B.1.3 can be applied to each packet arrival event at each gateway, and hence the result follows. It is also worth pointing out that the arrival processes at the gateway nodes may be heterogeneous, and they do not change according to the scheduling policy. A weaker version of Theorem 3.3.1 can be obtained as follows.

Corollary 3.3.1.1. *If the conditions of Theorem 3.3.1 hold, then for any arbitrary packet generation and arrival processes at the external source and node 0, respectively, and for all* $\pi \in \Pi$

$$\mathbf{\Delta}_{prmp\text{-}LGFS} \leq_{st} \mathbf{\Delta}_{\pi}. \tag{3.3.3}$$

Proof. We consider a mixture over the realizations of packet generation and arrival processes (arrival process at node 0) to prove the result. In particular, by using the result of Theorem 3.3.1 and Theorem 6.B.16.(e) in [47], the corollary follows. □

3.3.2 New-Better-than-Used Transmission Times, Preemption is Allowed

Although the preemptive LGFS policy can achieve age-optimality when the transmission times are exponentially distributed, it does not always, as we will observe later, minimize the age for non-exponential transmission times. We aim to answer the question of whether for an important class of distributions that are more general than exponential, optimality or near-optimality can be achieved while preemption is allowed. We here consider the classes of New-Better-than-Used (NBU) packet transmission time distributions, which are defined as follows.

Definition 3.3.2. ***New-Better-than-Used distributions [47]:*** *Consider a non-negative random variable X with complementary cumulative distribution function (CCDF) $\bar{F}(x) = \mathbb{P}[X > x]$. Then, X is* ***New-Better-than-Used (NBU)*** *if for all $t, \tau \geq 0$*

$$\bar{F}(\tau + t) \leq \bar{F}(\tau)\bar{F}(t). \tag{3.3.4}$$

Examples of NBU [4] distributions include constant transmission time, (shifted) exponential distribution, geometric distribution, Erlang distribution, negative binomial distribution, etc. Recently, age was analyzed in single-hop networks for exponential transmission times with transmission error in [51], and for Gamma-distributed transmission times in [10]. These studies did not answer the question of which policy can be (near) age-optimal for non-exponential transmission times in single-hop networks. We provided a unified answer to identify the policy that is near age-optimal in single-hop networks in Chapter 2. Since the question has remained open for multihop networks, we here extend our investigation to answer this question in multihop networks and identify the near age-optimal policy for a more general class of transmission time distributions.

We propose a non-preemptive LGFS (non-prmp-LGFS) policy. It is important to note that under non-prmp-LGFS policy, the fresh packet replaces the oldest packet in a link's queue when the queue is already at its maximum buffer level (i.e., the queue is already full). The implementation details of non-prmp-LGFS policy are depicted in Algorithm 3.2.

---

[4] The word **better** in the terminology **New-Better-than-Used** refers to that a random variable with a long lifetime is better than that with a shorter lifetime [47]. In our case, the random variable is the transmission time, and longer transmission time is worse in terms of the age. Thus, the word **better** here does not imply an improvement in the age performance.

Algorithm 3.2: Non-preemptive Last-Generated, First-Served policy at the link (i, j).

$\delta_{ij} := 0$; // δ_{ij} is the smallest generation time of the packets in the queue (B_{ij})
while *the system is ON* **do**
    if *a new packet* p_i *with generation time* s *arrives to node* i **then**
        if *the link* (i, j) *is busy* **then**
            if *Buffer* (B_{ij}) *is full* **then**
                if $s > \delta_{ij}$ **then**
                    Packet p_i replaces the packet with generation time δ_{ij} in the queue;
                else
                    Drop packet p_i;
                Set δ_{ij} to the smallest generation time of the packets in the queue (B_{ij});
            else
                Store packet p_i in the queue;
                Set δ_{ij} to the smallest generation time of the packets in the queue (B_{ij});
        else // The link is idle.
            The new packet is sent over the link;
    if *a packet is delivered to node* j **then**
        if *the queue is not empty* **then**
            The freshest packet in the queue is sent over the link;

While we are able to consider a more general class of transmission time distributions, we are able to prove this result for a somewhat more restrictive network than the general topology $\mathcal{G}(\mathcal{V}, \mathcal{L})$. The network here is represented by a directed graph $\mathcal{G}'(\mathcal{V}, \mathcal{L})$, in which each node $j \in \mathcal{V}\backslash\{0\}$ has one incoming link. An example of this network topology is shown in Fig. 3.2. We show that the non-prmp-LGFS policy can come close to age-optimal into two steps: i) we construct an infeasible policy which provides the age lower bound, ii) we then show the near age-optimality result by identifying the gap between the constructed lower bound and our proposed policy

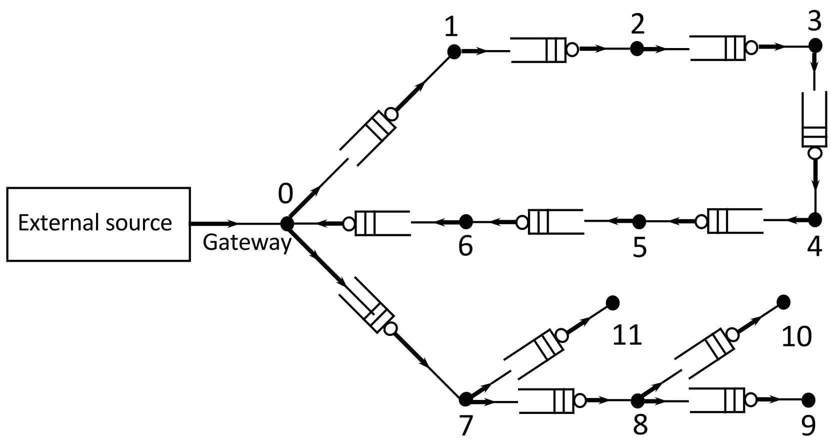


Figure 3.2: Information updates over a multihop network, where each node in the network (except the gateway) is restricted to receive data from only one node.

non-prmp-LGFS. The construction of the the infeasible policy and the lemma that explains the age lower bound are presented in Appendix B.2.

We can now proceed to characterize the age performance of policy non-prmp-LGFS among the policies in Π. Define a set of the parameters $\mathcal{I}' = \{\mathcal{G}'(\mathcal{V}, \mathcal{L}), (B_{ij}, (i,j) \in \mathcal{L}), s_l, a_{l0}, l = 1, 2, \ldots\}$, where $\mathcal{G}'(\mathcal{V}, \mathcal{L})$ is the network graph with the new restriction, B_{ij} is the queue buffer size of the link (i,j), s_l is the generation time of packet l, and a_{l0} is the arrival time of packet l to node 0. Define $\mathcal{H}_k$ as the set of nodes in the k-th hop, i.e., $\mathcal{H}_k$ is the set of nodes that are separated by k links from node 0 [5]. Let $i_{j,k}$ represent the index of the node in $\mathcal{H}_k$ that is in the path to the node j (for example, in Fig. 3.2, $i_{11,1} = 7$ and $i_{10,2} = 8$). Define X_j as the packet transmission time over the incoming link to node j. We use Lemma B.2.1 in Appendix B.2 to prove the following theorem.

Theorem 3.3.2. *Suppose that the packet transmission times are NBU, independent*

[5]Node 0 is in $\mathcal{H}_0$.

across links, and i.i.d. across time, then for all $\mathcal{I}'$ *satisfying* $B_{ij} \geq 1$ *for each* $(i,j) \in \mathcal{L}$

$$\min_{\pi \in \Pi} [\bar{\Delta}_{j,\pi} | \mathcal{I}'] \leq [\bar{\Delta}_{j,\textit{non-prmp-LGFS}} | \mathcal{I}'] \leq \min_{\pi \in \Pi} [\bar{\Delta}_{j,\pi} | \mathcal{I}'] + \mathbb{E}[X_{i_{j,1}}] + 2 \sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}], \forall j \in \mathcal{H}_k, \forall k \geq 1, \tag{3.3.5}$$

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

Proof sketch. We use the infeasible policy and the lower bound process that are constructed in Appendix B.2 to prove Theorem 3.3.2 into three steps:

Step 1: We derive an upper bound on the time differences between the arrival times (at each node) of the fresh packets under the infeasible policy and those under policy non-prmp-LGFS.

Step 2: We use the upper bound derived in Step 1 to derive an upper bound on the average gap between the constructed infeasible policy in Appendix B.2 and the non-prmp-LGFS policy.

Step 3: Finally, we use the upper bound on the average gap together with Lemma B.2.1 in Appendix B.2 to prove (3.3.5). For the full proof, see Appendix B.5. □

Theorem 3.3.2 tells us that for arbitrary sequence of packet generation times $\{s_1, s_2, \ldots\}$, sequence of arrival times $\{a_{10}, a_{20}, \ldots\}$ at node 0, and buffer sizes ($B_{ij} \geq 1, (i,j) \in \mathcal{L}$), the non-prmp-LGFS policy is within a constant age gap from the optimum average age among all policies in Π. Similar to Theorem 3.3.1, we can show that the result of Theorem 3.3.2 still holds for the multiple-gateway model shown in Fig. 1.3(b).

Remark 3.3.2.1. *The reason behind considering the restrictive network topology* $\mathcal{G}'(\mathcal{V}, \mathcal{L})$ *is as follows: In the general network topology* $\mathcal{G}(\mathcal{V}, \mathcal{L})$*, a node can receive update packets from multiple paths. As a result, the arrival time of a fresh packet* *at this node depends on the fastest path that delivers this packet to this node. This fastest path may differ from one packet to another on sample-path. Thus, it becomes challenging to establish an upper bound that is very close to the age lower bound (Steps 1 and 2 in the proof of Theorem 3.3.2) using sample-path and coupling techniques, in this case.*

3.3.3 General Transmission Times, Preemption is Not Allowed

Finally, we study age-optimal packet scheduling for networks that do not allow for preemption and for which the packet transmission times are *arbitrarily* distributed, *independent* across the links and *i.i.d.* across time. Since preemption is not allowed, we are restricted to non-preemptive policies within Π. Moreover, we consider work-conserving policies. We use $\Pi_{npwc} \subset \Pi$ to denote the set of non-preemptive work-conserving policies.

We consider the non-prmp-LGFS policy, where we show that it is age-optimal among the policies in Π_{npwc} in the following theorem.

Theorem 3.3.3. *If the packet transmission times are independent across the links and i.i.d. across time, then for all $\mathcal{I}$ and $\pi \in \Pi_{npwc}$*

$$[\mathbf{\Delta}_{\textit{non-prmp-LGFS}}|\mathcal{I}] \leq_{st} [\mathbf{\Delta}_{\pi}|\mathcal{I}], \tag{3.3.6}$$

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

$$\mathbb{E}[g(\mathbf{\Delta}_{\textit{non-prmp-LGFS}})|\mathcal{I}] = \min_{\pi \in \Pi_{npwc}} \mathbb{E}[g(\mathbf{\Delta}_{\pi})|\mathcal{I}], \tag{3.3.7}$$

provided the expectations in (3.3.7) *exist.*

Proof. The proof of Theorem 3.3.3 is similar to that of Theorem 3.3.1. The difference is that preemption is not allowed here. See Appendix B.6 for more details. □

It is interesting to note from Theorem 3.3.3 that, age-optimality can be achieved for arbitrary transmission time distributions, even if the transmission time distribution differs from a link to another. General service time distributions have been considered in some recent age analysis on single-hop networks [6, 60]. Theorem 3.3.3 explains the age-optimal policies in these scenarios. Moreover, similar to Theorem 3.3.1, the result of Theorem 3.3.3 still holds for the multiple-gateway model shown in Fig. 1.3(b).

Remark 3.3.3.1. *It is worth observing that the results in Theorem 3.3.1, Theorem 3.3.2, and Theorem 3.3.3 hold for any link buffer sizes B_{ij}'s. Hence, the buffer sizes can be chosen according to the application. In particular, in some applications, such as news and social updates, users are interested in not just the latest updates, but also past news. Thus, in such application, we may need to have queues with buffer sizes greater than one to store old packets and send them later whenever links become idle. On the other hand, there are some other applications, in which old packets become useless when the fresher packets exist. Thus, in these applications, buffer sizes can be chosen to be zero (one) when we follow the prmp-LGFS (non-prmp-LGFS) scheduling policy.*

3.4 Numerical Results

We now present numerical results that validate our theoretical findings. The inter-generation times at all setups are *i.i.d.* Erlang-2 distribution with mean $1/\lambda$.

We use Fig. 3.3 to validate the results in Section 3.3.2. We consider the network in Fig. 3.2. Fig. 3.3 illustrates the average age at node 5 under gamma transmission time

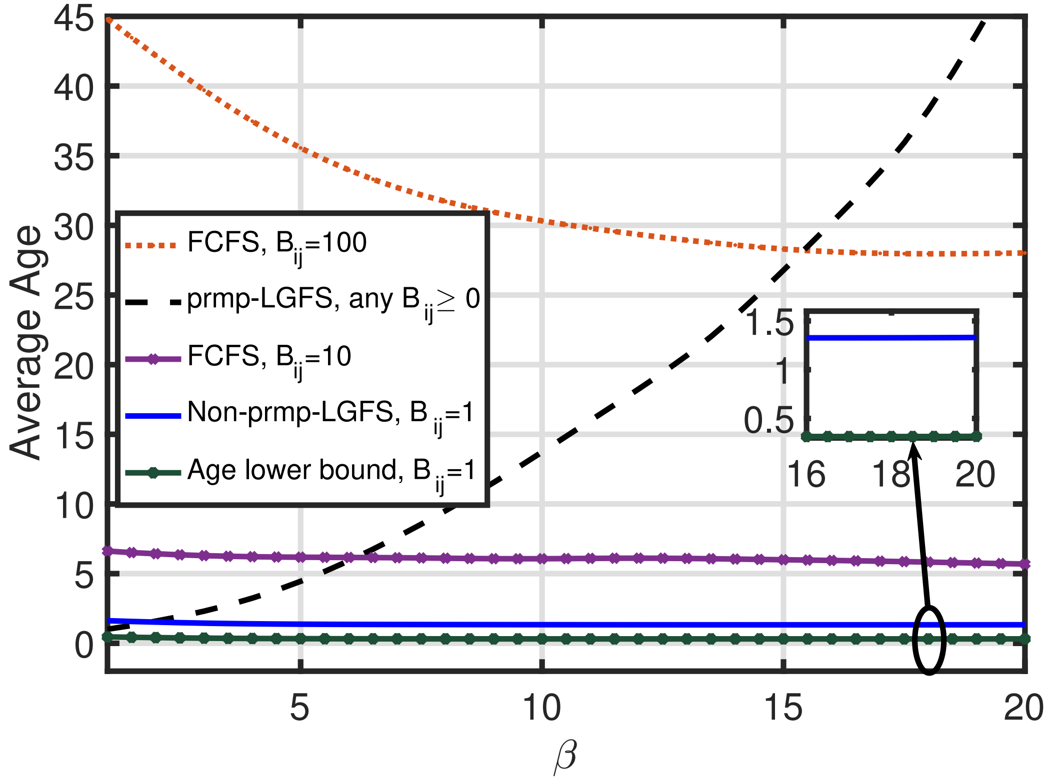


Figure 3.3: Average age at node 5 under gamma transmission time distributions at each link with different shape parameter β.

distributions at each link with different shape parameter β, where the buffer sizes are either 1, 10, or 100. The mean of the gamma transmission time distributions at each link is normalized to 0.2. The time difference $(a_{i0} - s_i)$ between packet generation and arrival to node 0 is Zero. Note that the average age of the FCFS policy with infinite buffer sizes is extremely high in this case and hence is not plotted in this figure. The "Age lower bound" curve is generated by using $\frac{\int_0^T \Delta_{5,IP}^{LB}}{T}$ when the buffer sizes are 1 which, according to Lemma B.2.1, is a lower bound of the optimum average age at node 5. We can observe that the gap between the "Age lower bound" curve and the average age of the non-prmp-LGFS policy at node 5 is no larger than $9E[X] = 1.8$, which agrees with Theorem 3.3.2. In addition, we can observe that prmp-LGFS

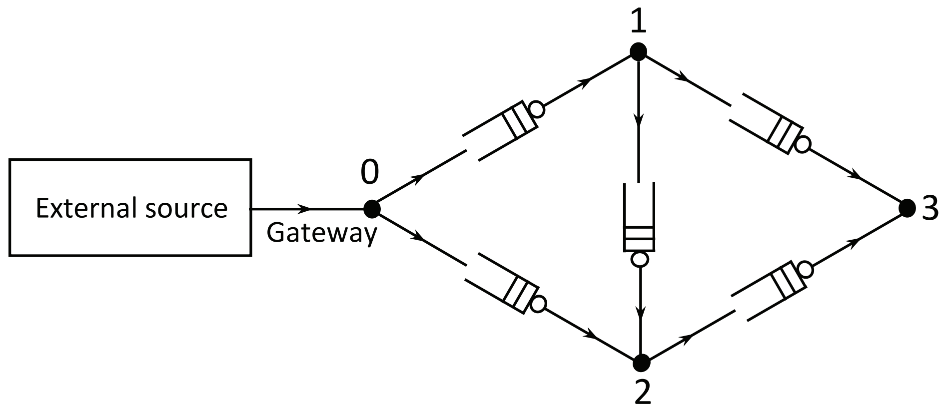


Figure 3.4: A multihop network.

policy achieves the best age performance among all plotted policies when $\beta = 1$. This is because a gamma distribution with shape parameter $\beta = 1$ is an exponential distribution. Thus, age-optimality can be achieved in this case by policy prmp-LGFS as stated in Theorem 3.3.1. However, as can be seen in the figure, the average age at node 5 of the prmp-LGFS policy blows up as the shape parameter β increases and the non-prmp-LGFS policy achieves the best age performance among all plotted policies when $\beta > 2$. The reason for this phenomenon is as follows: As β increases, the variance (variability) of normalized gamma distribution decreases. 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 generation rate is high, packet preemption happens frequently, which leads to infrequent packet delivery and increases the age. This phenomenon occurs heavily at the first link (link $(0, 1)$) which, in turn, affects the age at the subsequent nodes.

We use Fig. 3.5 to validate the result in Section 3.3.3. We consider the network in Fig. 3.4. The time difference between packet generation and arrival to node 0, i.e., $a_{i0} - s_i$, is modeled to be either 1 or 100, with equal probability. Fig. 3.5 plots the time-average age at node 3 versus the packets generation rate λ for the multihop

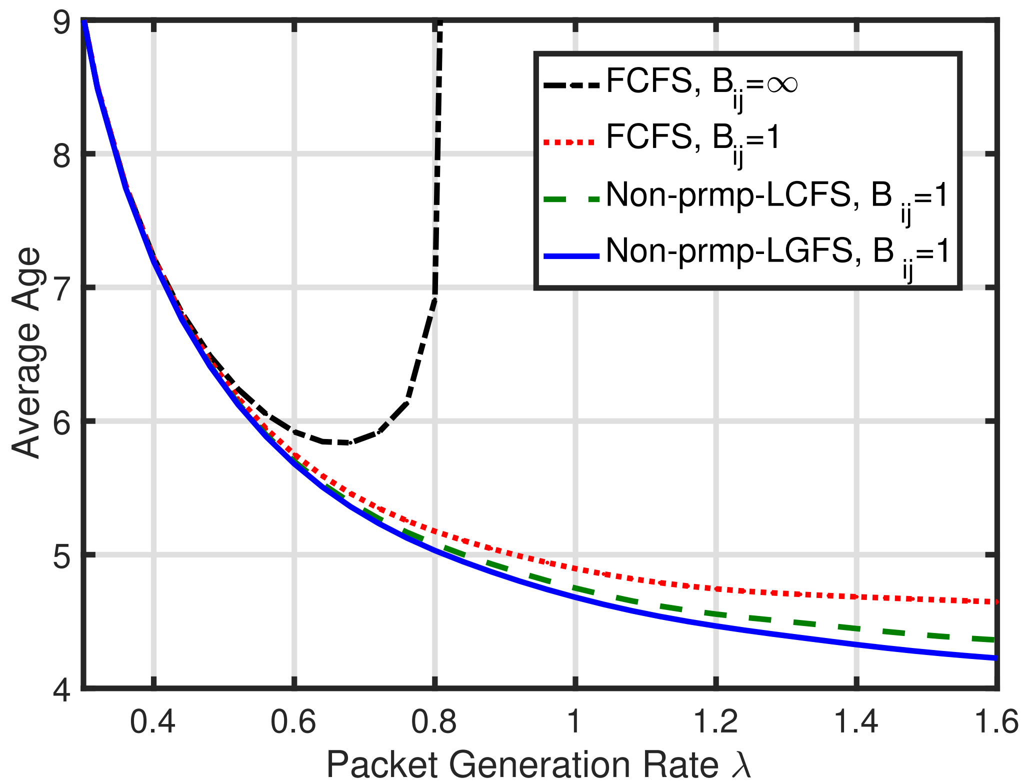


Figure 3.5: Average age at node 3 versus packets generation rate λ for general packet transmission time distributions.

network in Fig. 3.4. The plotted policies are FCFS policy, non-preemptive LCFS, and non-preemptive LGFS policy, where the buffer sizes are either 1 or infinity. The packet transmission times at links $(0,1)$ and $(1,3)$ follow a gamma distribution with mean 1. The packet transmission times at links $(0,2)$, $(1,2)$, and $(2,3)$ are distributed as the sum of a constant with value 0.5 and a value drawn from an exponential distribution with mean 0.5. We find that the non-preemptive LGFS policy achieves the best age performance among all plotted policies. By comparing the age performance of the non-preemptive LGFS and non-preemptive LCFS policies, we observe that the LGFS scheduling principle improves the age performance when the update packets arrive to node 0 out of the order of their generation times. It is important to note that the non-preemptive LGFS policy minimizes the age among the non-preemptive work-conserving policies even if the packet transmission time distributions are heterogeneous across the links. This observation agrees with Theorem 3.3.3. We also observe that the average age of FCFS policy with $B_{ij} = \infty$ 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 the FCFS policy with $B_{ij} = 1$, the average age is finite 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 $B_{ij} = \infty$.

3.5 Conclusion

In this chapter, we studied the age minimization problem in interference-free multihop networks. We considered general system settings including arbitrary network topology, packet generation and arrival times at node 0, and queue buffer sizes. A number of scheduling policies were developed and proven to be (near) age-optimal in a stochastic ordering sense for minimizing any non-decreasing functional of the age processes. In particular, we showed that age-optimality can be achieved when: i) preemption is allowed and the packet transmission times are exponentially distributed, ii) preemption is not allowed and the packet transmission times are arbitrarily distributed (among work-conserving policies). Moreover, for networks that allow for preemption and the packet transmission times are NBU, we showed that the non-preemptive LGFS policy is near age-optimal in a somewhat more restrictive network topology.

CHAPTER 4

AGE-OPTIMAL SAMPLING AND SCHEDULING IN MULTI-SOURCE SYSTEMS

4.1 Introduction

An important line of research in age of information has considered the "generate-at-will" model [16–19], in which the generation times (sampling times) of the update packets are controllable. The work in [18, 19] motivated the usage of nonlinear age functions from various real-time applications and designed sampling policies for optimizing nonlinear age functions in single source systems. Our study here extends the work in [18, 19] to a multi-source system. In this system, only one packet can be sent through the channel at a time. Therefore, a decision maker does not only consist of a sampler, but also a scheduler, which makes the problem even more challenging.

The scheduling problem for multi-source networks with different scenarios was considered in [26–29, 32–34, 36, 61–66]. In [34], the authors found that the scheduling problem for minimizing the age in wireless networks under physical interference constraints is NP-hard. Optimal scheduling for age minimization in a broadcast network was studied in [29, 32, 62–64], where a single source can be scheduled at a time. In addition, it was found that a maximum age first (MAF) service discipline is useful for reducing the age in various multi-source systems with different service time distributions in [28, 29, 36, 62, 63]. In contrast to our study, the generation of the update packets in [28, 29, 32, 34, 36, 62–64] is uncontrollable and they arrive randomly at the transmitter. Age analysis of the status updates over a multiaccess channel was considered in [26]. The studies in [27, 33, 65, 66] considered the age optimization problem in a wireless network with general interference constraints and channel uncertainty. Our result in Corollary 4.3.3.1 suggests that if the packet transmission time is fixed as in time-slotted systems [26–29, 32–34, 62–66], then it is optimal to sample as soon as the channel becomes available. However, this is not necessarily true otherwise.

In this chapter, we consider a joint sampling and scheduling problem for optimizing data freshness in multi-source systems. We use a non-decreasing age-penalty function to represent the level of dissatisfaction of data staleness, where all sources have the same age-penalty function. Sources take turns to generate update packets, and forward them to their destinations one-by-one through a shared channel with random delay. There is a scheduler, that chooses the update order of the sources, and a sampler, that determines when a source should generate a new packet in its turn. We aim to find the optimal scheduler-sampler pairs that minimize the total-average age-penalty at delivery times (Ta-APD) and the total-average age-penalty (Ta-AP), where Ta-AP is more challenging to minimize. To that end, the main contributions of this chapter are outlined as follows:

- We formulate the optimal scheduling and sampling problem to optimize data freshness in single-hop, multi-source networks. We show that our optimization problem has an important *separation principle*: For any given sampler, we show that the optimal scheduling policy is the Maximum Age First (MAF) scheduler (Proposition 4.3.1). Hence, the optimal scheduler-sampler pair can be obtained by fixing the scheduling policy to the MAF scheduler, and then optimize the sampler design separately.

- We show that the MAF scheduler and zero-wait sampler, in which a new packet is generated once the channel becomes idle, are jointly optimal for minimizing the Ta-APD (Theorem 4.3.1). We show this result by proving the optimality of the zero-wait sampler for minimizing the Ta-APD, when the scheduling policy is fixed to the MAF scheduler.

- Interestingly, we find that zero-wait sampler does not always minimize the Ta-AP, when the MAF scheduler is employed. We show that the MAF scheduler and the relative value iteration with reduced complexity (RVI-RC) sampler are jointly optimal for minimizing the Ta-AP (Theorem 4.3.2). We take several steps to prove the optimality of the RVI-RC sampler: When the scheduling policy is fixed to the MAF scheduler, we reformulate the optimal sampling problem for minimizing the Ta-AP as an equivalent semi-Markov decision problem. We use Dynamic Programming (DP) to obtain the optimal sampler. In particular, we show that there exists a stationary deterministic sampler that can achieve optimality (Proposition 4.3.2). We also show that the optimal sampler has a threshold property (Proposition 4.3.3), that helps in reducing the complexity of the relative value iteration (RVI) algorithm (by reducing the computations required for some system states). This results in the RVI-RC sampler in Algorithm 4.1.

- Finally, in Section 4.4, we devise a low-complexity threshold-type sampler via an approximate analysis of Bellman's equation whose solution is the RVI-RC sampler. In addition, for the special case of a linear age-penalty function, this threshold sampler is further simplified to the water-filling solution. The numerical results in Figs. 4.4-4.9 indicate that, when the scheduler is fixed to the MAF, the performance of these approximated samplers is almost the same as that of the RVI-RC sampler.

4.2 Model and Formulation

This section describes the multi-source network illustrated in Fig. 1.4, and introduces necessary notations.

4.2.1 Notations

We use $\mathbb{N}^+$ to represent the set of non-negative integers, $\mathbb{R}^+$ is the set of non-negative real numbers, $\mathbb{R}$ is the set of real numbers, and $\mathbb{R}^n$ is the set of n-dimensional real Euclidean space. We use t^- to denote the time instant just before t. 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 $i = 1, 2, \ldots, n$. Also, we use $x_{[i]}$ to denote the i-th largest component of vector $\mathbf{x}$.

4.2.2 System Model

We consider a status update system with m sources as shown in Fig. 1.4, where each source observes a time-varying process. An update packet is generated from a source and is then sent over an error-free delay channel to the destination, where only one packet can be sent at a time. A decision maker controls the transmission order of the sources and the generation times of the update packets for each source. This is known as the "generate-at-will" model [16–18] (i.e., the update packets can be generated at any time).

We use S_i to denote the generation time of the i-th generated packet from all sources, called packet i. Moreover, we use r_i to represent the source index from which packet i is generated. The channel is modeled as an FCFS queue with random *i.i.d.* service time Y_i, where Y_i represents the service time of packet i, $Y_i \in \mathcal{Y}$, and $\mathcal{Y} \subset \mathbb{R}^+$ is a finite and bounded set. We also assume that $0 < \mathbb{E}[Y_i] < \infty$ for all i. We suppose that the decision maker knows the idle/busy state of the server through acknowledgments (ACKs) from the destination with zero delay. If an update packet is generated while the server is busy, this packet needs to wait in the queue until its transmission opportunity, and becomes stale while waiting. Hence, there is no loss of optimality to avoid generating an update packet during the busy periods. As a result, a packet is served immediately once it is generated. Let D_i denote the delivery time of packet i, where $D_i = S_i + Y_i$. After the delivery of packet i at time D_i, the decision maker may insert a waiting time Z_i before generating a new packet (hence, $S_{i+1} = D_i + Z_i$)[1], where $Z_i \in \mathcal{Z}$, and $\mathcal{Z} \subset \mathbb{R}^+$ is a finite and bounded set[2].

At any time t, the most recently delivered packet from source l is generated at time

$$U_l(t) = \max\{S_i : r_i = l, D_i \leq t\}. \tag{4.2.1}$$

Age of information, or simply the *age*, for source l is defined as [1–4]

$$\Delta_l(t) = t - U_l(t). \tag{4.2.2}$$

As shown in Fig. 4.1, the age increases linearly with t but is reset to a smaller value with the delivery of a fresher packet. We suppose that the age $\Delta_l(t)$ is right-continuous. The age process for source l is given by $\{\Delta_l(t), t \geq 0\}$. We suppose that the initial age values $\Delta_l(0^-)$ for all l are known to the system. For notation simplicity, we use a_{li} to denote the age value of source l at time D_i, i.e., $a_{li} = \Delta_l(D_i)$[3].

For each source l, we consider an age-penalty function $g(\Delta_l(t))$ of the age $\Delta_l(t)$.

[1]We suppose that $D_0 = 0$. Thus, we have $S_1 = Z_0$.

[2]We suppose that we always have $0 \in \mathcal{Z}$.

[3]Since the age process is right-continuous, if packet i is delivered from source l, then $\Delta_l(D_i)$ is the age value of source l just after the delivery time D_i.

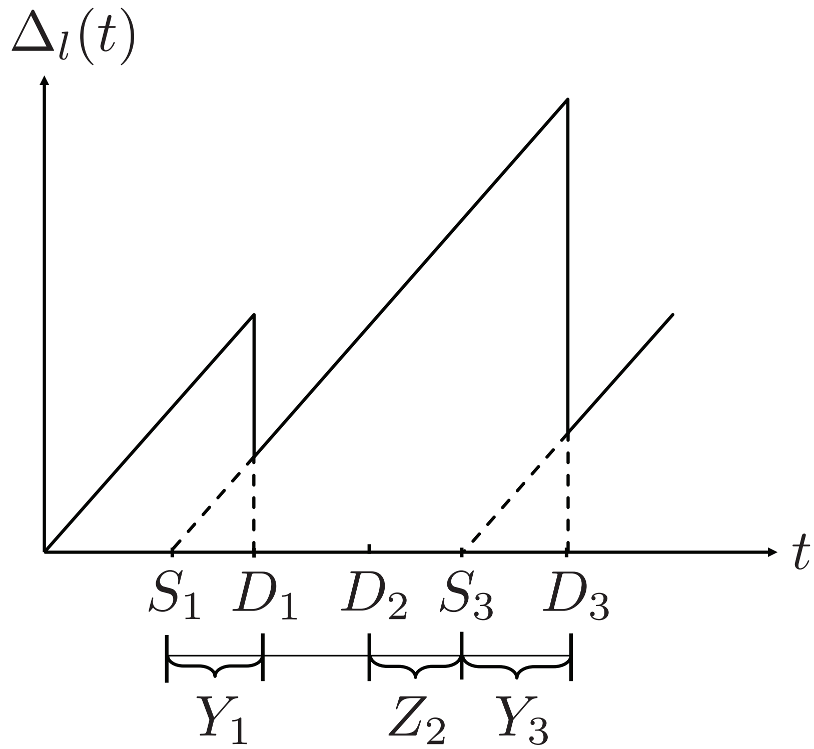


Figure 4.1: The age $\Delta_l(t)$ of source l, where we suppose that the first and third packets are generated from source l, i.e., $r_1 = r_3 = l$.

The function $g : [0, \infty) \rightarrow \mathbb{R}$ is non-decreasing and is not necessarily convex or continuous. We suppose that $\mathbb{E}[|\int_a^{a+x} g(\tau)d\tau|] < \infty$ whenever $x < \infty$. It was recently shown in [19] that, under certain conditions, information freshness metrics expressed in terms of auto-correlation functions, the estimation error of signal values, and mutual information, are monotonic functions of the age. Moreover, the age-penalty function $g(\cdot)$ can be used to represent the level of dissatisfaction of data staleness in different applications based on their demands. For instance, a stair-shape function $g(x) = \lfloor x \rfloor$ can be used to characterize the dissatisfaction for data staleness when the information of interest is checked periodically, an exponential function $g(x) = e^x$ can be utilized in online learning and control applications in which the demand for updating data increases quickly with age, and an indicator function $g(x) = \mathbb{1}(x > q)$ can be used to indicate the dissatisfaction of the violation of an age limit q.

4.2.3 Decision Policies

A decision policy, denoted by d, controls the following: i) the scheduler, denoted by π, that determines the source to be served at each transmission opportunity $\pi \triangleq (r_1, r_2, \ldots)$, ii) the sampler, denoted by f, that determines the packet generation times $f \triangleq (S_1, S_2, \ldots)$, or equivalently, the sequence of waiting times $f \triangleq (Z_0, Z_1, \ldots)$. Hence, $d = (\pi, f)$ implies that a decision policy d employs the scheduler π and the sampler f. Let $\mathcal{D}$ denote the set of causal decision policies in which decisions are made based on the history and current information of the system. Observe that $\mathcal{D}$ consists of Π and $\mathcal{F}$, where Π and $\mathcal{F}$ are the sets of causal schedulers and samplers, respectively.

After each delivery, the decision maker chooses the source to be served, and imposes a waiting time before the generation of the new packet. Next, we present our optimization problems.

4.2.4 Optimization Problem

We define two metrics to assess the long term age performance over our status update system in (4.2.3) and (4.2.4). Consider the time interval $[0, D_n]$. For any decision policy $d = (\pi, f)$, we define the total-average age-penalty at delivery times (Ta-APD) as

$$\Delta_{\text{avg-D}}(\pi, f) = \limsup_{n\to\infty} \frac{1}{n}\mathbb{E}\left[\sum_{l=1}^{m}\sum_{i=1}^{n} g\left(\Delta_l(D_i^-)\right)\right], \tag{4.2.3}$$

and the total-average age-penalty per unit time (Ta-AP) as

$$\Delta_{\text{avg}}(\pi, f) = \limsup_{n\to\infty} \frac{\mathbb{E}\left[\sum_{l=1}^{m}\int_0^{D_n} g\left(\Delta_l(t)\right) dt\right]}{\mathbb{E}\left[D_n\right]}. \tag{4.2.4}$$

In this chapter, we aim to minimize both the Ta-APD and the Ta-AP separately. In other words, we seek a decision policy $d = (\pi, f)$ that solves the following optimization problems:

$$\bar{\Delta}_{\text{avg-D-opt}} \triangleq \min_{\pi\in\Pi, f\in\mathcal{F}} \Delta_{\text{avg-D}}(\pi, f), \tag{4.2.5}$$

and

$$\bar{\Delta}_{\text{avg-opt}} \triangleq \min_{\pi\in\Pi, f\in\mathcal{F}} \Delta_{\text{avg}}(\pi, f), \tag{4.2.6}$$

where $\bar{\Delta}_{\text{avg-D-opt}}$ and $\bar{\Delta}_{\text{avg-opt}}$ are the optimum objective values of Problems (4.2.5) and (4.2.6), respectively. Due to the large decision policy space, the optimization problem is quite challenging. In particular, we need to seek the optimal decision policy that controls both the scheduler and sampler to minimize the Ta-APD and the Ta-AP. In the next section, we discuss our approach to tackle these optimization problems.

4.3 Optimal Decision Policy

We first show that our optimization problems in (4.2.5) and (4.2.6) have an important separation principle: Given the generation times of the update packets, the Maximum Age First (MAF) scheduler provides the best age performance compared to any other scheduler. What remains to be addressed is the question of finding the best sampler that solves Problems (4.2.5) and (4.2.6), given that the scheduler is fixed to the MAF. Next, we present our approach to solve our optimization problems in detail.

4.3.1 Optimal Scheduler

We start by defining the MAF scheduler as follows:

Definition 4.3.1 ([28, 29, 36, 62, 63]). *Maximum Age First scheduler: In this scheduler, the source with the maximum age is served first among all sources. Ties are broken arbitrarily.*

For simplicity, let π_{MAF} represent the MAF scheduler. The age performance of π_{MAF} scheduler is characterized in the following proposition:

Proposition 4.3.1. *For all $f \in \mathcal{F}$*

$$\Delta_{\text{avg-D}}(\pi_{MAF}, f) = \min_{\pi \in \Pi} \Delta_{\text{avg-D}}(\pi, f), \tag{4.3.1}$$

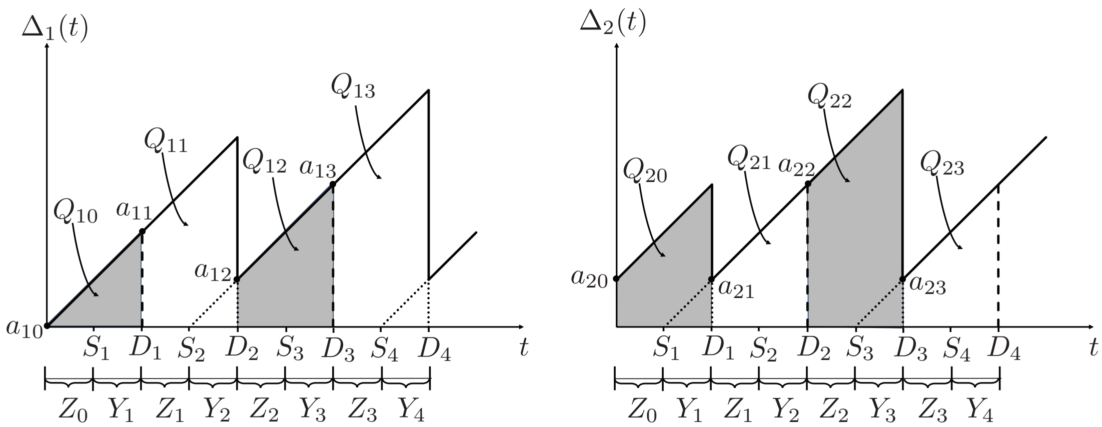
$$\Delta_{\text{avg}}(\pi_{MAF}, f) = \min_{\pi\in\Pi} \Delta_{\text{avg}}(\pi, f). \tag{4.3.2}$$

That is, the MAF scheduler minimizes both the Ta-APD and the Ta-AP in (4.2.3) *and* (4.2.4) *among all schedulers in* Π.

Proof. One of the key ideas of the proof is as follows: Given any sampler, that controls the generation times of the update packets, we only control from which source a packet is generated. We couple the policies such that the packet delivery times are fixed under all decision policies. In the MAF scheduler, a source with maximum age becomes the source with minimum age among the m sources after each delivery. Under any arbitrary scheduler, a packet can be generated from any source, which is not necessarily the one with the maximum age, and the chosen source becomes the one with minimum age among the m sources after the delivery. Since the age-penalty function, $g(\cdot)$, is non-decreasing, the MAF scheduler provides a better age performance compared to any other scheduler. For details, see Appendix C.1. □

Proposition 4.3.1 is proven by using a sample-path proof technique that was recently developed in [36]. The difference is that the authors in [36] proved the results for symmetrical packet generation and arrival processes, while we consider here that the packet generation times are controllable. We found that the same proof technique applies to both cases. Observe that, Proposition 1 holds when all sources are equally prioritized. However, for the sources with different priorities (i.e. different age-penalty functions), this result does not hold anymore. This is because the order of the age-penalty values of various sources may change with time.

Proposition 4.3.1 helps us conclude the separation principle that the optimal sampler can be optimized separately, given that the scheduling policy is fixed to the MAF



(a) The age evolution of source 1. (b) The age evolution of source 2.

Figure 4.2: The age processes evolution of the MAF scheduler in a two-sources information update system. Source 2 has a higher initial age than Source 1. Thus, Source 2 starts service and Packet 1 is generated from Source 2, which is delivered at time D_1. Then, Source 1 is served and Packet 2 is generated from Source 1, which is delivered at time D_2. The same operation is repeated over time.

scheduler. Hence, the optimization problems (4.2.5) and (4.2.6) reduce to the following:

$$\bar{\Delta}_{\text{avg-D-opt}} \triangleq \min_{f \in \mathcal{F}} \Delta_{\text{avg-D}}(\pi_{\text{MAF}}, f), \tag{4.3.3}$$

$$\bar{\Delta}_{\text{avg-opt}} \triangleq \min_{f \in \mathcal{F}} \Delta_{\text{avg}}(\pi_{\text{MAF}}, f). \tag{4.3.4}$$

By fixing the scheduling policy to the MAF scheduler, the evolution of the age processes of the sources is as follows: The sampler may impose a waiting time Z_i before generating packet $i+1$ at time $S_{i+1} = D_i + Z_i$ from the source with the maximum age at time $t = D_i$. Packet $i+1$ is delivered at time $D_{i+1} = S_{i+1} + Y_{i+1}$ and the age of the source with maximum age drops to the minimum age with the value of Y_{i+1}, while the age processes of other sources increase linearly with time without change. This operation is repeated with time and the age processes evolve accordingly. An example of age processes evolution is shown in Fig. 4.2. Next, we seek the optimal sampler for Problems (4.3.3) and (4.3.4).

4.3.2 Optimal Sampler for Problem (4.3.3)

Now, we show that the MAF scheduler and the zero-wait sampler are jointly optimal for minimizing the Ta-APD as follows:

Theorem 4.3.1. *The MAF scheduler and the zero-wait sampler form an optimal solution for Problem* (4.2.5).

Proof. We prove Theorem 4.3.1 by proving that the zero-wait sampler is optimal for Problem (4.3.3). In particular, we show that the Ta-APD is an increasing function of the packets waiting times Z_i's. For details, see Appendix C.2. □

Remark 4.3.1.1. *The results in Proposition 4.3.1 and Theorem 4.3.1 hold even if $\mathcal{Y}$ and $\mathcal{Z}$ are unbounded and uncountable sets. Indeed, the finiteness assumption of $\mathcal{Y}$ and $\mathcal{Z}$ is only needed for the utilization of the DP technique in the next subsection.*

4.3.3 Optimal Sampler for Problem (4.3.4)

Although the zero-wait sampler is the optimal sampler for minimizing the Ta-APD, it is not clear whether it also minimizes the Ta-AP. This is because the latter metric may not be a non-decreasing function of the waiting times as we will see later, which makes Problem (4.3.4) more challenging. Next, we derive the Ta-AP when the MAF scheduler is employed and reformulate Problem (4.3.4) as a semi-Markov decision problem.

Reformulation of Problem (4.3.4)

We start by analyzing the Ta-AP when the scheduling policy is fixed to the MAF scheduler. We decompose the area under each curve $g(\Delta_l(t))$ into a sum of disjoint geometric parts. Observing Fig. 4.2 [4], this area in the time interval $[0, D_n]$, where $D_n = \sum_{i=0}^{n-1} Z_i + Y_{i+1}$, can be seen as the concatenation of the areas Q_{li}, $0 \le i \le n-1$. Thus, we have

$$\int_0^{D_n} g(\Delta_l(t))dt = \sum_{i=0}^{n-1} Q_{li}, \tag{4.3.5}$$

where

$$Q_{li} = \int_{D_i}^{D_{i+1}} g(\Delta_l(t))dt = \int_{D_i}^{D_i+Z_i+Y_{i+1}} g(\Delta_l(t))dt. \tag{4.3.6}$$

For $t \in [D_i, D_{i+1})$, we have

$$\Delta_l(t) = t - U_l(t) = t - (D_i - a_{li}), \tag{4.3.7}$$

where $(D_i - a_{li})$ represents the generation time of the last delivered packet from source l before time D_{i+1}. By performing a change of variable in (4.3.6), we get

$$Q_{li} = \int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau)d\tau. \tag{4.3.8}$$

Hence, the Ta-AP can be rewritten as

$$\limsup_{n\to\infty} \frac{\sum_{i=0}^{n-1} \mathbb{E}\left[\sum_{l=1}^{m} \int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau)d\tau\right]}{\sum_{i=0}^{n-1} \mathbb{E}\left[Z_i + Y_{i+1}\right]}. \tag{4.3.9}$$

[4] Observe that a special age-penalty function is depicted in Fig. 4.2, where we choose $g(x) = x$ to simplify the illustration.

Using this, the optimal sampling problem for minimizing the Ta-AP, given that the scheduling policy is fixed to the MAF scheduler, can be cast as

$$\bar{\Delta}_{\text{avg-opt}} \triangleq \min_{f\in\mathcal{F}} \limsup_{n\to\infty} \frac{\sum_{i=0}^{n-1} \mathbb{E}\left[\sum_{l=1}^{m} \int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau)d\tau\right]}{\sum_{i=0}^{n-1} \mathbb{E}\left[Z_i + Y_{i+1}\right]}. \tag{4.3.10}$$

Since $|\int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau)d\tau| < \infty$ for all $Z_i \in \mathcal{Z}$ and $Y_i \in \mathcal{Y}$, and $\mathbb{E}[Y_i] > 0$ for all i, $\bar{\Delta}_{\text{avg-opt}}$ is bounded. Note that Problem (4.3.10) is hard to solve in the current form. Therefore, we reformulate it. We consider the following optimization problem with a parameter $\beta \geq 0$:

$$\Theta(\beta) \triangleq \min_{f\in\mathcal{F}} \limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{E}\left[\sum_{l=1}^{m} \int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau)d\tau - \beta(Z_i + Y_{i+1})\right], \tag{4.3.11}$$

where $\Theta(\beta)$ is the optimal value of (4.3.11).

Lemma 4.3.1. *The following assertions are true:*

(i). $\bar{\Delta}_{\textit{avg-opt}} \lesseqqgtr \beta$ *if and only if* $\Theta(\beta) \lesseqqgtr 0$.

(ii). If $\Theta(\beta) = 0$, *then the optimal sampling policies that solve* (4.3.10) *and* (4.3.11) *are identical.*

Proof. The proof of Lemma 4.3.1 is similar to the proof of [67, Lemma 2]. The difference is that a regenerative assumption of the inter-sampling times is used to prove the result in [67]; instead, we use the boundedness of the inter-sampling times to prove the result. For the sake of completeness, we modify the proof accordingly and provide it in Appendix C.3. □

As a result of Lemma 4.3.1, the solution to (4.3.10) can be obtained by solving (4.3.11) and seeking a $\beta = \bar{\Delta}_{\text{avg-opt}} \geq 0$ such that $\Theta(\bar{\Delta}_{\text{avg-opt}}) = 0$. Lemma 4.3.1 helps us to utilize the DP technique to obtain the optimal sampler. Note that without

Lemma 4.3.1, it would be quite difficult to use the DP technique to solve (4.3.10) optimally. Next, we illustrate our solution approach to Problem (4.3.11) in detail.

The solution of Problem (4.3.11)

Following the methodology proposed in [68], when $\beta = \bar{\Delta}_{\text{avg-opt}}$, Problem (4.3.11) is equivalent to an average cost per stage problem. According to [68], we describe the components of this problem in detail below.

- **States:** At stage[5] i, the system state is specified by

$$\mathbf{s}(i) = (a_{[1]i}, \ldots, a_{[m]i}), \tag{4.3.12}$$

  where $a_{[l]i}$ is the l-th largest age of the sources at stage i, i.e., it is the l-th largest component of the vector $(a_{1i}, \ldots, a_{mi})$. We use $\mathcal{S}$ to denote the state-space including all possible states. Notice that $\mathcal{S}$ is finite and bounded because $\mathcal{Z}$ and $\mathcal{Y}$ are finite and bounded.

- **Control action:** At stage i, the action that is taken by the sampler is $Z_i \in \mathcal{Z}$.

- **Random disturbance:** In our model, the random disturbance occurring at stage i is Y_{i+1}, which is independent of the system state and the control action.

- **Transition probabilities:** If the control $Z_i = z$ is applied at stage i and the service time of packet $i+1$ is $Y_{i+1} = y$, then the evolution of the system state from $\mathbf{s}(i)$ to $\mathbf{s}(i+1)$ is as follows:

$$\begin{aligned} &a_{[m]i+1} = y, \\ &a_{[l]i+1} = a_{[l+1]i} + z + y,\ l = 1, \ldots, m-1. \end{aligned} \tag{4.3.13}$$

[5]From henceforward, we assume that the duration of stage i is $[D_i, D_{i+1})$.

We let $\mathbb{P}_{\mathbf{ss}'}(z)$ denote the transition probabilities

$$\mathbb{P}_{\mathbf{ss}'}(z)=\mathbb{P}(\mathbf{s}(i+1)=\mathbf{s}'|\mathbf{s}(i)=\mathbf{s}, Z_i=z),\ \mathbf{s},\mathbf{s}'\in\mathcal{S}. \tag{4.3.14}$$

When $\mathbf{s} = (a_{[1]}, \ldots, a_{[m]})$ and $\mathbf{s}' = (a'_{[1]}, \ldots, a'_{[m]})$, the law of the transition probability is given by

$$\mathbb{P}_{\mathbf{ss}'}(z) = \begin{cases} \mathbb{P}(Y_{i+1} = y) & \text{if } a'_{[m]}=y \text{ and} \\ & a'_{[l]}=a_{[l+1]}+z+y \text{ for } l\neq m; \\ 0 & \text{else.} \end{cases} \tag{4.3.15}$$

- **Cost Function:** Each time the system is in stage i and control Z_i is applied, we incur a cost

$$C(\mathbf{s}(i), Z_i, Y_{i+1}) = \sum_{l=1}^{m} \int_{a_{[l]i}}^{a_{[l]i}+Z_i+Y_{i+1}} g(\tau)d\tau - \bar{\Delta}_{\text{avg-opt}}(Z_i + Y_{i+1}). \tag{4.3.16}$$

To simplify notation, we use the expected cost $C(\mathbf{s}(i), Z_i)$ as the cost per stage, i.e.,

$$C(\mathbf{s}(i), Z_i) = \mathbb{E}_{Y_{i+1}}\left[C(\mathbf{s}(i), Z_i, Y_{i+1})\right], \tag{4.3.17}$$

where $\mathbb{E}_{Y_{i+1}}$ is the expectation with respect to Y_{i+1}, which is independent of $\mathbf{s}(i)$ and Z_i. It is important to note that there exists $c \in \mathbb{R}^+$ such that $|C(\mathbf{s}(i), Z_i)| \leq c$ for all $\mathbf{s}(i) \in \mathcal{S}$ and $Z_i \in \mathcal{Z}$. This is because $\mathcal{Z}$, $\mathcal{Y}$, $\mathcal{S}$, and $\bar{\Delta}_{\text{avg-opt}}$ are bounded.

In general, the average cost per stage under a sampling policy $f \in \mathcal{F}$ is given by

$$\limsup_{n\to\infty} \frac{1}{n}\mathbb{E}\left[\sum_{i=0}^{n-1} C(\mathbf{s}(i), Z_i)\right]. \tag{4.3.18}$$

We say that a sampling policy $f \in \mathcal{F}$ is *average-optimal* if it minimizes the average cost per stage in (4.3.18). Our objective is to find the average-optimal sampling policy. A policy f is called a stationary deterministic policy if $Z_i = q(\mathbf{s}(i))$ for all $i = 0, 1, \ldots$, where $q : \mathbb{R}^{m+} \to \mathcal{Z}$ is a deterministic function. In the next proposition, we show that there is a stationary deterministic policy that is average-optimal.

Proposition 4.3.2. *There exist a scalar λ and a function h that satisfy the following Bellman's equation:*

$$\lambda + h(\mathbf{s}) = \min_{z \in \mathcal{Z}} \left(C(\mathbf{s}, z) + \sum_{\mathbf{s}' \in \mathcal{S}} \mathbb{P}_{\mathbf{s}\mathbf{s}'}(z) h(\mathbf{s}') \right), \tag{4.3.19}$$

where λ is the optimal average cost per stage that is independent of the initial state $\mathbf{s}(0)$ and satisfies

$$\lambda = \lim_{\alpha \to 1} (1 - \alpha) J_\alpha(\mathbf{s}), \forall \mathbf{s} \in \mathcal{S}, \tag{4.3.20}$$

and $h(\mathbf{s})$ is the relative cost function that, for any state $\mathbf{o}$, satisfies

$$h(\mathbf{s}) = \lim_{\alpha \to 1} (J_\alpha(\mathbf{s}) - J_\alpha(\mathbf{o})), \forall \mathbf{s} \in \mathcal{S}, \tag{4.3.21}$$

where $J_\alpha(\mathbf{s})$ is the optimal total expected α-discounted cost function, which is defined by

$$J_\alpha(\mathbf{s}) = \min_{f \in \mathcal{F}} \limsup_{n \to \infty} \mathbb{E} \left[\sum_{i=0}^{n-1} \alpha^i C(\mathbf{s}(i), Z_i) \right], \mathbf{s}(0) = \mathbf{s} \in \mathcal{S}, \tag{4.3.22}$$

where $0 < \alpha < 1$ is the discount factor. Furthermore, there exists a stationary deterministic policy that attains the minimum in (4.3.19) for each $\mathbf{s} \in \mathcal{S}$ and is average-optimal.

Proof. According to [68, Proposition 4.2.1 and Proposition 4.2.6], it is enough to show that for every two states $\mathbf{s}$ and $\mathbf{s}'$, there exists a stationary deterministic policy f such that for some k, we have $\mathbb{P}\left[\mathbf{s}(k) = \mathbf{s}'|\mathbf{s}(0) = \mathbf{s}, f\right] > 0$, i.e., we have a communicating Markov decision process (MDP). Observe that the proof idea of this proposition is different from those used in literature such as [63, 64], where they have used the discounted cost problem to show their results and then connect it to the average cost problem. For details, see Appendix C.4. □

We can deduce from Proposition 4.3.2 that the optimal waiting time is a fixed function of the state $\mathbf{s}$. Next, we use the RVI algorithm to obtain the optimal sampler for minimizing the Ta-AP, and then exploit the structure of our problem to reduce its complexity.

Optimal Sampler Structure: The RVI algorithm [69, Section 9.5.3], [70, Page 171] can be used to solve Bellman's equation (4.3.19). Starting with an arbitrary state $\mathbf{o}$, a single iteration for the RVI algorithm is given as follows:

$$\begin{aligned} Q_{n+1}(\mathbf{s}, z) &= C(\mathbf{s}, z) + \sum_{\mathbf{s}' \in \mathcal{S}} \mathbb{P}_{\mathbf{s}\mathbf{s}'}(z) h_n(\mathbf{s}'), \\ J_{n+1}(\mathbf{s}) &= \min_{z \in \mathcal{Z}} (Q_{n+1}(\mathbf{s}, z)), \\ h_{n+1}(\mathbf{s}) &= J_{n+1}(\mathbf{s}) - J_{n+1}(\mathbf{o}), \end{aligned} \tag{4.3.23}$$

where $Q_{n+1}(\mathbf{s}, z)$, $J_n(\mathbf{s})$, and $h_n(\mathbf{s})$ denote the state action value function, value function, and relative value function for iteration n, respectively. In the beginning, we set $J_0(\mathbf{s}) = 0$ for all $\mathbf{s} \in \mathcal{S}$, and then we repeat the iteration of the RVI algorithm as described before[6].

[6] According to [69, 70], a sufficient condition for the convergence of the RVI algorithm is the aperiodicity of the transition matrices of stationary deterministic optimal policies. In our case, these transition matrices depend on the service times. This condition can always be achieved

The complexity of the RVI algorithm is high due to many sources (i.e., the curse of dimensionality [71]). Thus, we need to simplify the RVI algorithm. To that end, we show that the optimal sampler has a threshold property that can reduce the complexity of the RVI algorithm. Define $z_{\mathbf{s}}^{\star}$ as the optimal waiting time for state $\mathbf{s}$, and Y as a random variable that has the same distribution as Y_i. The threshold property in the optimal sampler is manifested in the following proposition:

Proposition 4.3.3. *If the state* $\mathbf{s} = (a_{[1]}, \ldots, a_{[m]})$ *satisfies* $\mathbb{E}_Y \left[\sum_{l=1}^{m} g(a_{[l]} + Y) \right] \geq \bar{\Delta}_{avg\text{-}opt}$*, then we have* $z_{\mathbf{s}}^{\star} = 0$.

Proof. See Appendix C.5. □

We can exploit the threshold test in Proposition 4.3.3 to reduce the complexity of the RVI algorithm as follows: The optimal waiting time for any state $\mathbf{s}$ that satisfies $\mathbb{E}_Y \left[\sum_{l=1}^{m} g(a_{[l]} + Y) \right] \geq \bar{\Delta}_{\text{avg-opt}}$ is zero. Thus, we need to solve (4.3.23) only for the states that fail this threshold test. As a result, we reduce the number of computations required along the system state space, which reduces the complexity of the RVI algorithm. Note that $\bar{\Delta}_{\text{avg-opt}}$ can be obtained using the bisection method or any other one-dimensional search method. Combining this with the result of Proposition 4.3.3 and the RVI algorithm, we propose the "RVI with reduced complexity (RVI-RC) sampler" in Algorithm 4.1. In the outer layer of Algorithm 4.1, bisection is employed to obtain $\bar{\Delta}_{\text{avg-opt}}$, where β converges to $\bar{\Delta}_{\text{avg-opt}}$.

Note that, according to [69, 70], $J(\mathbf{o})$ in Algorithm 4.1 converges to the optimal average cost per stage. Moreover, the value of u in Algorithm 4.1 can be initialized to the value of the Ta-AP of the zero-wait sampler (as the Ta-AP of the zero-wait sampler provides an upper bound on the optimal Ta-AP), which can be easily calculated.

---

by applying the aperiodicity transformation as explained in [69, Section 8.5.4], which is a simple transformation. However, This is not always necessary to be done.

Algorithm 4.1: RVI algorithm with reduced complexity.

given $l = 0$, sufficiently large u, tolerance $\epsilon_1 > 0$, tolerance $\epsilon_2 > 0$;
while $u - l > \epsilon_1$ **do**
  $\beta = \frac{l+u}{2}$;
  $J(\mathbf{s}) = 0$, $h(\mathbf{s}) = 0$, $h_{\text{last}}(\mathbf{s}) = 10\epsilon_2$ for all states $\mathbf{s} \in \mathcal{S}$;
  while $\max_{\mathbf{s}\in\mathcal{S}} |h(\mathbf{s}) - h_{last}(\mathbf{s})| > \epsilon_2$ **do**
    for *each* $\mathbf{s} \in \mathcal{S}$ **do**
      if $\mathbb{E}_Y \left[\sum_{l=1}^{m} g(a_{[l]} + Y) \right] \geq \beta$ **then**
        $z_s^\star = 0$;
      else
        $z_s^\star = \text{argmin}_{z\in\mathcal{Z}} C(\mathbf{s}, z) + \sum_{\mathbf{s}'\in\mathcal{S}} \mathbb{P}_{\mathbf{ss}'}(z) h(\mathbf{s}')$;
      $J(\mathbf{s}) = C(\mathbf{s}, z_s^\star) + \sum_{\mathbf{s}'\in\mathcal{S}} \mathbb{P}_{\mathbf{ss}'}(z_s^\star) h(\mathbf{s}')$;
    $h_{\text{last}}(\mathbf{s}) = h(\mathbf{s})$;
    $h(\mathbf{s}) = J(\mathbf{s}) - J(\mathbf{o})$;
  if $J(\mathbf{o}) \geq 0$ **then**
    $u = \beta$;
  else
    $l = \beta$;

The RVI algorithm and Whittle's methodology have been used in literature to obtain the optimal age scheduler in time-slotted multi-source networks (e.g., [63, 64]). Since they considered a time-slotted system, their model belongs to the class of Markov decision problems. In contrast, we consider random discrete transmission times that can be more than one time slot. Thus, our model belongs to the class of semi-Markov decision problems, and hence is different from those in [63, 64].

In conclusion, an optimal solution for Problem (4.2.6) is manifested in the following theorem:

Theorem 4.3.2. *The MAF scheduler and the RVI-RC sampler form an optimal solution for Problem* (4.2.6)*.*

Proof. The theorem follows directly from Proposition 4.3.1, Proposition 4.3.2, and Proposition 4.3.3. □

Special Case of $g(x) = x$

Now we consider the case of $g(x) = x$ and obtain some useful insights. Define $A_s = \sum_{l=1}^{m} a_{[l]}$ as the sum of the age values of state $\mathbf{s}$. The threshold test in Proposition 4.3.3 is simplified as follows:

Proposition 4.3.4. *If the state* $\mathbf{s} = (a_{[1]}, \dots, a_{[m]})$ *satisfies* $A_s \geq (\bar{\Delta}_{avg\text{-}opt} - m\mathbb{E}[Y])$*, then we have* $z_s^{\star} = 0$*.*

Proof. The proposition follows directly by substituting $g(x) = x$ into the threshold test in Proposition 4.3.3. □

Hence, the only change in Algorithm 4.1 is to replace the threshold test in Step 7 by $A_s \geq (\bar{\Delta}_{\text{avg-opt}} - m\mathbb{E}[Y])$. Let $y_{\text{inf}} = \inf\{y \in \mathcal{Y} : \mathbb{P}[Y = y] > 0\}$, i.e., y_{inf} is the smallest possible transmission time in $\mathcal{Y}$. As a result of Proposition 4.3.4, we obtain the following sufficient condition for the optimality of the zero-wait sampler for minimizing the Ta-AP when $g(x) = x$:

Theorem 4.3.3. *If*

$$y_{inf} \geq \frac{(m-1)\mathbb{E}[Y]^2 + \mathbb{E}[Y^2]}{(m+1)\mathbb{E}[Y]}, \tag{4.3.24}$$

then the zero-wait sampler is optimal for Problem (4.3.11)*.*

Proof. See Appendix C.6 □

From Theorem 4.3.3, it immediately follows that:

Corollary 4.3.3.1. *If the transmission times are positive and constant (i.e.,* $Y_i = const > 0$ *for all* i*), then the zero-wait sampler is optimal for Problem* (4.3.11)*.*

Proof. The corollary follows directly from Theorem 4.3.3 by showing that (4.3.24) always holds in this case. □

Corollary 4.3.3.1 suggests that the designed schedulers in [26, 27, 29, 32–34, 62–66] are indeed optimal in time-slotted systems. However, if there is a variation in the transmission times, these schedulers alone may not be optimal anymore, and we need to optimize the sampling times as well.

4.4 Low-complexity Sampler Design via Bellman's Equation Approximation

In this section, we try to obtain low-complexity samplers via an approximate analysis for Bellman's equation in (4.3.19). The obtained low-complexity samplers in this section will be shown to have near optimal age performance in our numerical results in Section 4.5. For a given state $\mathbf{s}$, we denote the next state given z and y by $\mathbf{s}'(z, y)$. We can observe that the transition probability in (4.3.15) depends only on the distribution of the packet service time which is independent of the system state and the control action. Hence, the second term in Bellman's equation in (4.3.19) can be rewritten as

$$\sum_{\mathbf{s}' \in \mathcal{S}} \mathbb{P}_{\mathbf{s}\mathbf{s}'}(z) h(\mathbf{s}'(z, y)) = \sum_{y \in \mathcal{Y}} \mathbb{P}(Y = y) h(\mathbf{s}'(z, y)). \tag{4.4.1}$$

As a result, Bellman's equation in (4.3.19) can be rewritten as

$$\lambda = \min_{z}\left(C(\mathbf{s}, z) + \sum_{y\in\mathcal{Y}} \mathbb{P}(Y = y)(h(\mathbf{s}'(z, y)) - h(\mathbf{s}))\right). \tag{4.4.2}$$

Although $h(\mathbf{s})$ is discrete, we can interpolate the value of $h(\mathbf{s})$ between the discrete values so that it is differentiable by following the same approach in [72] and [73]. Let $\mathbf{s} = (a_{[1]}, \ldots, a_{[m]})$, then using the first order Taylor approximation around a state $\mathbf{v} = (a^v_{[1]}, \ldots, a^v_{[m]})$ (some fixed state), we get

$$h(\mathbf{s}) \approx h(\mathbf{v}) + \sum_{l=1}^{m}(a_{[l]} - a^v_{[l]})\frac{\partial h(\mathbf{v})}{\partial a_{[l]}}. \tag{4.4.3}$$

Again, we use the first order Taylor approximation around the state $\mathbf{v}$, together with the state evolution in (4.3.13), to get

$$h(\mathbf{s}'(z, y)) \approx h(\mathbf{v}) + (y - a^v_{[m]})\frac{\partial h(\mathbf{v})}{\partial a_{[m]}} + \sum_{l=1}^{m-1}(a_{[l+1]} - a^v_{[l]} + z + y)\frac{\partial h(\mathbf{v})}{\partial a_{[l]}}. \tag{4.4.4}$$

From (4.4.3) and (4.4.4), we get

$$h(\mathbf{s}'(z, y)) - h(\mathbf{s}) \approx (y - a_{[m]})\frac{\partial h(\mathbf{v})}{\partial a_{[m]}} + \sum_{l=1}^{m-1}(a_{[l+1]} - a_{[l]} + z + y)\frac{\partial h(\mathbf{v})}{\partial a_{[l]}}. \tag{4.4.5}$$

This implies that

$$\begin{aligned}\sum_{y\in\mathcal{Y}} \mathbb{P}(Y = y)(h(\mathbf{s}'(z, y)) - h(\mathbf{s})) \approx &(\mathbb{E}[Y] - a_{[m]})\frac{\partial h(\mathbf{v})}{\partial a_{[m]}} + \\ &\sum_{l=1}^{m-1}(a_{[l+1]} - a_{[l]} + z + \mathbb{E}[Y])\frac{\partial h(\mathbf{v})}{\partial a_{[l]}}.\end{aligned} \tag{4.4.6}$$

Using (4.4.2) with (4.4.6), we can get the following approximated Bellman's equation:

$$\lambda \approx \min_z \left(C(\mathbf{s}, z) + (\mathbb{E}[Y] - a_{[m]}) \frac{\partial h(\mathbf{v})}{\partial a_{[m]}} + \sum_{l=1}^{m-1} (a_{[l+1]} - a_{[l]} + z + \mathbb{E}[Y]) \frac{\partial h(\mathbf{v})}{\partial a_{[l]}} \right). \quad (4.4.7)$$

By following the same steps as in Appendix C.5 to get the optimal z that minimizes the objective function in (4.4.7), we get the following condition: The optimal z, for a given state $\mathbf{s}$, must satisfy

$$\mathbb{E}_Y \left[\sum_{l=1}^{m} g(a_{[l]} + t + Y) \right] - \bar{\Delta}_{\text{avg-opt}} + \sum_{l=1}^{m-1} \frac{\partial h(\mathbf{v})}{\partial a_{[l]}} \geq 0 \quad (4.4.8)$$

for all $t > z$, and

$$\mathbb{E}_Y \left[\sum_{l=1}^{m} g(a_{[l]} + t + Y) \right] - \bar{\Delta}_{\text{avg-opt}} + \sum_{l=1}^{m-1} \frac{\partial h(\mathbf{v})}{\partial a_{[l]}} \leq 0 \quad (4.4.9)$$

for all $t < z$. The smallest z that satisfies (4.4.8)-(4.4.9) is

$$\hat{z}_s^\star = \inf \left\{ t \geq 0 : \mathbb{E}_Y \left[\sum_{l=1}^{m} g(a_{[l]} + t + Y) \right] \geq \bar{\Delta}_{\text{avg-opt}} - \sum_{l=1}^{m-1} \frac{\partial h(\mathbf{v})}{\partial a_{[l]}} \right\}, \quad (4.4.10)$$

where $\hat{z}_s^*$ is the optimal solution of the approximated Bellman's equation for state $\mathbf{s}$. Note that the term $\sum_{i=1}^{m-1} \frac{\partial h(\mathbf{v})}{\partial a_{[i]}}$ is constant. Hence, (4.4.10) can be rewritten as

$$\hat{z}_s^\star = \inf \left\{ t \geq 0 : \mathbb{E}_Y \left[\sum_{l=1}^{m} g(a_{[l]} + t + Y) \right] \geq T \right\}. \quad (4.4.11)$$

This simple threshold sampler can approximate the optimal sampler for the original Bellman's equation in (4.3.19). The optimal threshold (T) in (4.4.11) can be obtained using a golden-section method [74]. Moreover, for a given state $\mathbf{s}$ and the threshold

(T), (4.4.11) can be solved using the bisection method or any other one-dimensional search method.

Low-complexity Water-filling Sampler: Consider the case that $g(x) = x$, the solution in (4.4.11) can be further simplified. Substituting $g(x) = x$ into (4.4.10), where the equality holds in this case, we get the following condition: The optimal z in this case, for a given state $\mathbf{s}$, must satisfy

$$A_s - \bar{\Delta}_{\text{avg-opt}} + mz + m\mathbb{E}[Y] + \sum_{l=1}^{m-1} \frac{\partial h(\mathbf{v})}{\partial a_{[l]}} = 0, \tag{4.4.12}$$

where A_s is the sum of the age values of state $\mathbf{s}$. Rearranging (4.4.12), we get

$$\hat{z}_s^* = \left[\frac{\bar{\Delta}_{\text{avg-opt}} - m\mathbb{E}[Y] - \sum_{l=1}^{m-1} \frac{\partial h(\mathbf{v})}{\partial a_{[l]}}}{m} - \frac{A_s}{m} \right]^+ . \tag{4.4.13}$$

By observing that the term $\displaystyle\sum_{i=1}^{m-1} \frac{\partial h(\mathbf{v})}{\partial a_{[i]}}$ is constant, (4.4.13) can be rewritten as

$$\hat{z}_s^* = \left[T - \frac{A_s}{m} \right]^+ , \tag{4.4.14}$$

The solution in (4.4.14) is in the form of the water-filling solution as we compare a fixed threshold (T) with the average age of a state $\mathbf{s}$. The solution in (4.4.14) suggests that this simple water-filling sampler can approximate the optimal solution of the original Bellman's equation in (4.3.19) when $g(x) = x$. Similar to the general case, the optimal threshold (T) in (4.4.14) can be obtained using a golden-section method. We evaluate the performance of the approximated samplers in the next section.

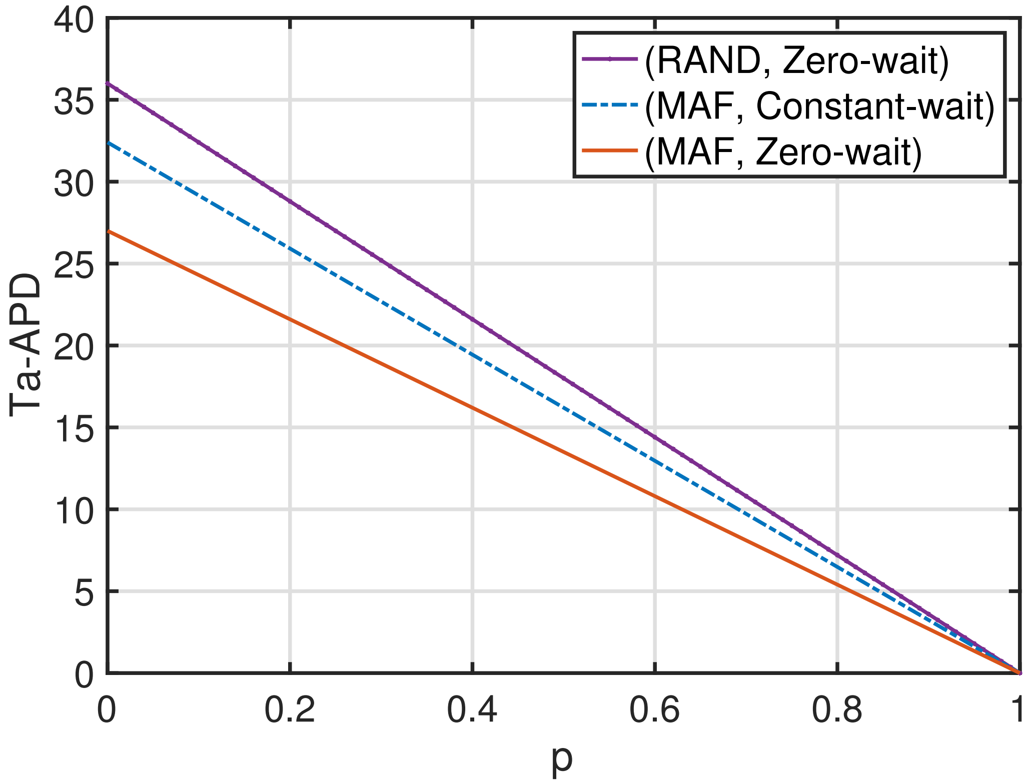


Figure 4.3: Ta-APD versus the probability p for an update system with $m = 3$ sources, where $g(x) = x$.

4.5 Numerical Results

We present numerical results to evaluate our proposed solutions. We consider an information update system with $m = 3$ sources. We use "RAND" to represent a random scheduler, where sources are chosen to be served with equal probability. By "Constant-wait", we refer to the sampler that imposes a constant waiting time after each delivery with $Z_i = 0.3\mathbb{E}[Y], \sim \forall i$. Moreover, we use "Threshold" and "Water-filling" to denote the proposed samplers in (4.4.11) and (4.4.14), respectively.

We set the transmission times to be either 0 or 3 with probability p and $1 - p$, respectively. Fig. 4.3 illustrates the Ta-APD versus the probability p, where we have $g(x) = x$. As we can observe, with fixing the sampler to the zero-wait one, the MAF

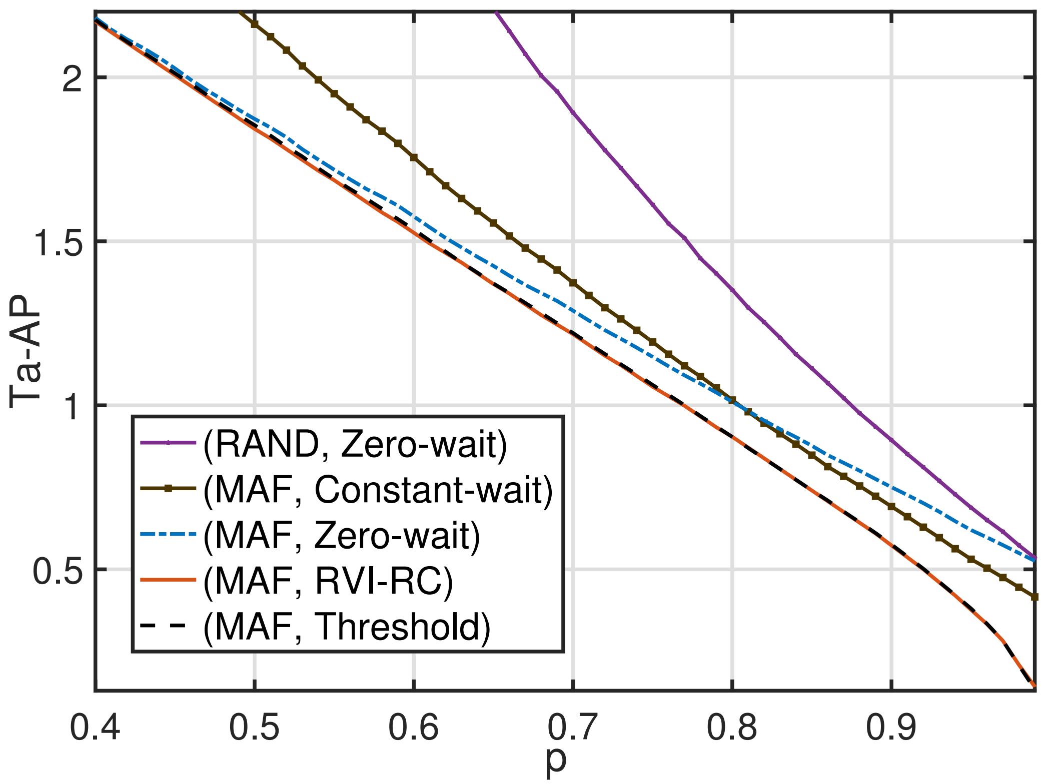


Figure 4.4: Ta-AP versus the probability p for an update system with $m = 3$ sources, where $g(x) = e^{0.1x} - 1$.

scheduler provides a lower Ta-APD than that of the RAND scheduler. Moreover, with fixing the scheduling policy to the MAF scheduler, the zero-wait sampler provides a lower Ta-APD compared to the constant-wait sampler. These observations agree with Theorem 4.3.1. However, as we will see later, zero-wait sampler does not always minimize the Ta-AP.

We now evaluate the performance of our proposed solutions for minimizing the Ta-AP. We set the transmission times to be either 0 or 3 with probability p and $1-p$, respectively. Figs. 4.4, 4.5, and 4.6 illustrate the Ta-AP versus the probability p, where we set the age-penalty function $g(x)$ to be $e^{0.1x} - 1$, $x^{0.1}$, and x, respectively. The range of the probability p is $[0.4, 0.99]$ in Figs. 4.4, 4.5, and 4.6. When $p = 1$,

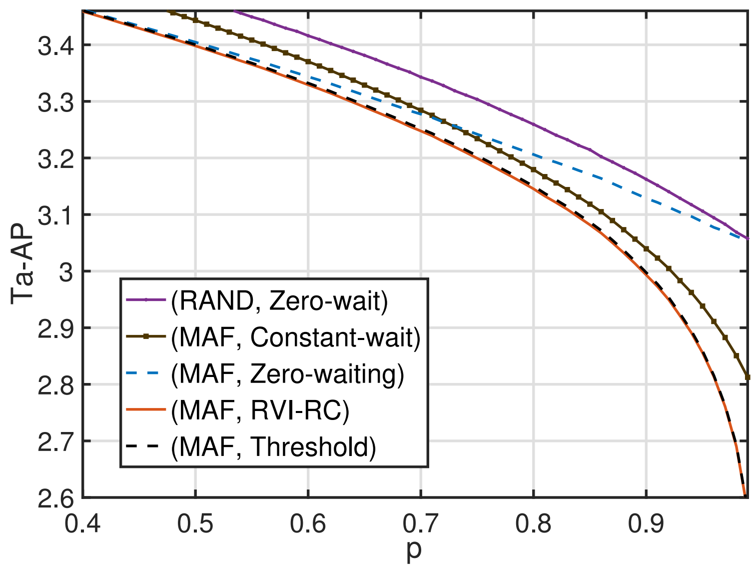


Figure 4.5: Ta-AP versus the probability p for an update system with $m = 3$ sources, where $g(x) = x^{0.1}$.

$\mathbb{E}[Y] = \mathbb{E}[Y^2] = 0$ and hence the Ta-AP of the zero-wait sampler (for any scheduler) at $p = 1$ is undefined. Therefore, the point $p = 1$ is not plotted in Figs. 4.4, 4.5, and 4.6. For the zero-wait sampler, we find that the MAF scheduler provides a lower Ta-AP than that of the RAND scheduler. This agrees with Proposition 4.3.1. Moreover, when the scheduling policy is fixed to the MAF scheduler, we find that the Ta-AP resulting from the RVI-RC sampler is lower than those resulting from the zero-wait sampler and the constant-wait sampler. This observation suggests the following: i) The zero-wait sampler does not necessarily minimize the Ta-AP, ii) optimizing the scheduling policy only is not enough to minimize the Ta-AP, but we have to optimize both the scheduling policy and the sampling policy together to minimize the Ta-AP. In addition, as we can observe, the Ta-AP resulting from the threshold sampler in Figs. 4.4 and 4.5, and the water-filling sampler in Fig. 4.6 almost coincides with the Ta-AP resulting from the RVI-RC sampler.

We then set the transmission times to be either 0 or $Y_{\max}$ with probability 0.9 and

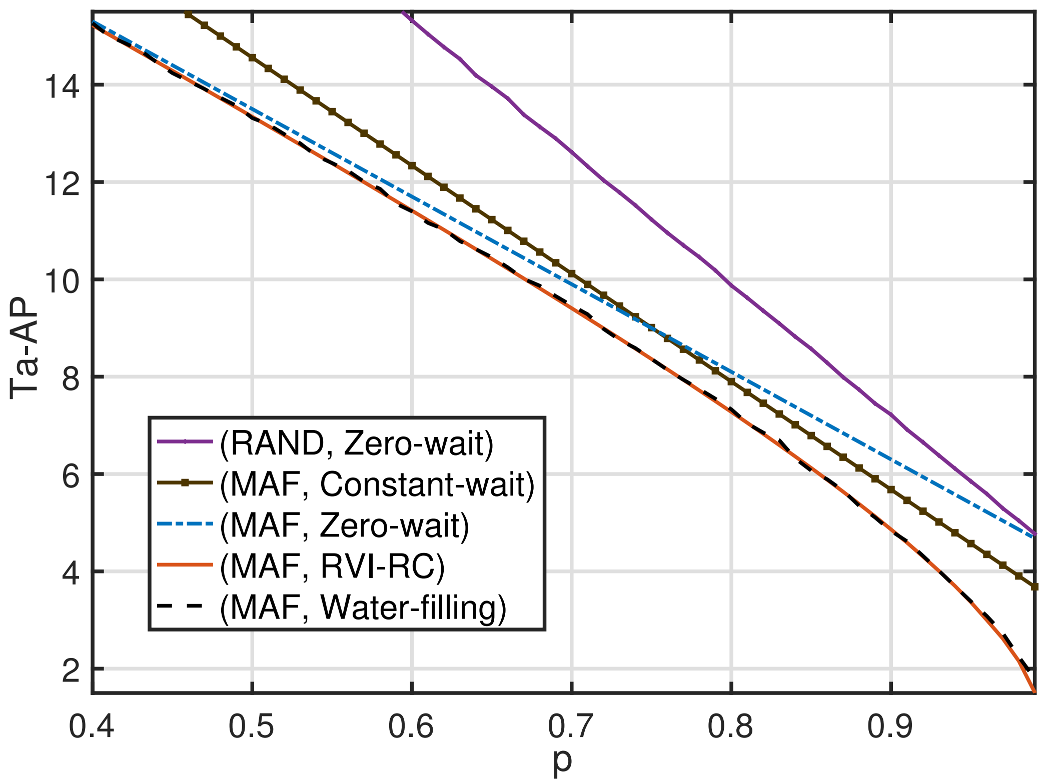


Figure 4.6: Ta-AP versus the probability p for an update system with $m = 3$ sources, where $g(x) = x$.

0.1, respectively. We vary the maximum transmission time $Y_{\max}$ and plot the Ta-AP in Figs. 4.7, 4.8, and 4.9, where $g(x)$ is set to be $e^{0.1x} - 1$, $x^{0.1}$, and x, respectively. The scheduling policy is fixed to the MAF scheduler in all plotted curves. We can observe in all figures that the Ta-AP resulting from the RVI-RC sampler is lower than those resulting from the zero-wait sampler and the constant-wait sampler, and the gap between them increases as the variability (variance) of the transmission times increases. This suggests that when the transmission times have a big variation, we have to optimize the scheduler and the sampler together to minimize the Ta-AP. We also can observe that the Ta-AP of the threshold sampler in Figs. 4.7 and 4.8, and the water-filling sampler in Fig. 4.9 almost coincides with that of the RVI-RC sampler.

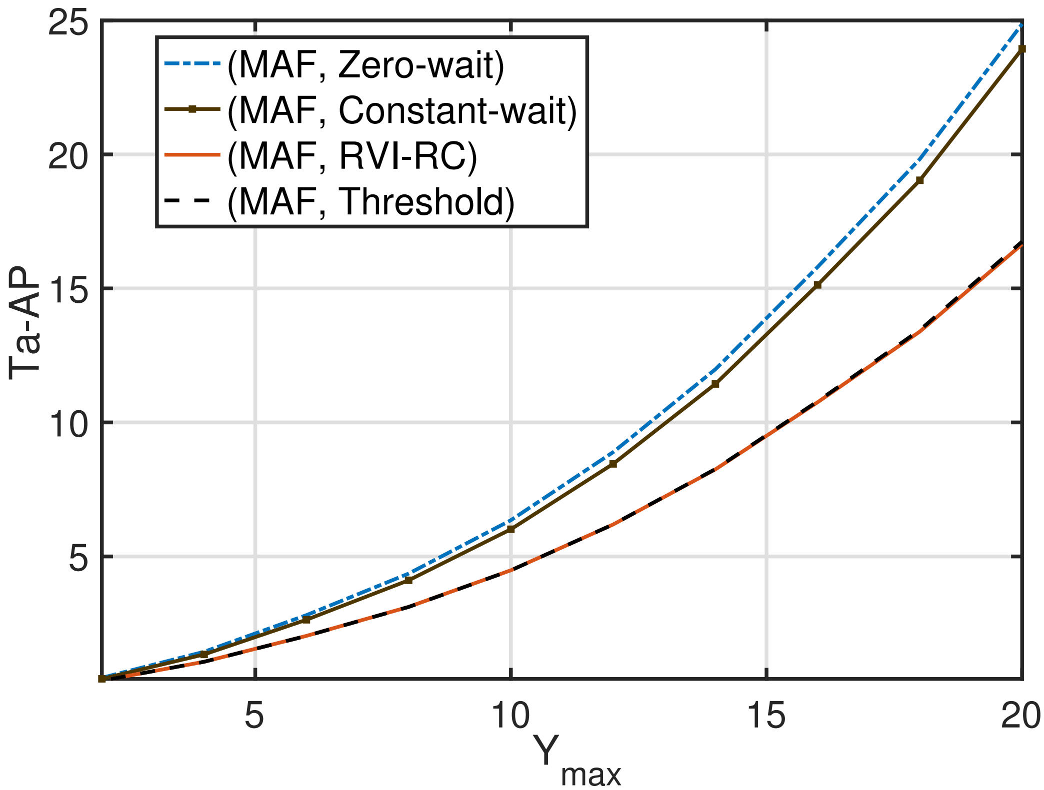


Figure 4.7: Ta-AP versus the maximum service time $Y_{\max}$ for an update system with $m = 3$ sources, where $g(x) = e^{0.1x} - 1$.

Finally, we consider a larger scale update system with $m = 10$. We model the transmission time as a discrete Markov chain with a probability mass function $\mathbb{P}[Y_i = 1] = 0.9$ and $\mathbb{P}[Y_i = 30] = 0.1$, and a transition matrix

$$\begin{bmatrix} \frac{8}{9} + \frac{\sigma}{9} & 1 - \frac{8}{9} - \frac{\sigma}{9} \\ 1 - \sigma & \sigma \end{bmatrix}. \tag{4.5.1}$$

Fig. 4.10 illustrates the Ta-AP versus the transition matrix parameter σ, where $g(x) = x$. As we can observe, the (MAF, water-filling) policy provides the lowest Ta-AP compared to all plotted policies. Also, when $\sigma = 1$, the transmission time reduces to be a constant time. We can observe that, when the scheduling policy is

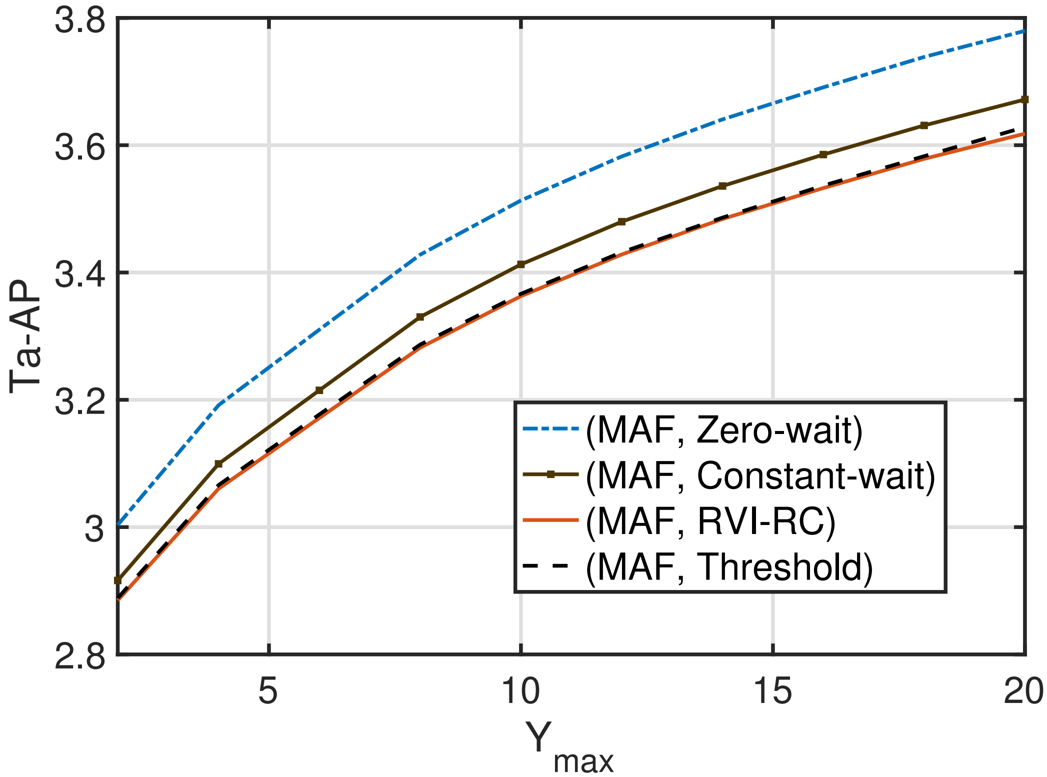


Figure 4.8: Ta-AP versus the maximum service time $Y_{\max}$ for an update system with $m = 3$ sources, where $g(x) = x^{0.1}$.

the MAF, the Ta-APs achieved by the zero-wait and water-filling samplers are equal. This agrees with Corollary 4.3.3.1.

4.6 Conclusion

In this chapter, we studied the problem of finding the optimal decision policy that controls the packet generation times and transmission order of the sources to minimize the Ta-APD and Ta-AP in a multi-source information update system. We showed that the MAF scheduler and the zero-wait sampler are jointly optimal for minimizing the Ta-APD. Moreover, we showed that the MAF scheduler and the RVI-RC sampler, which results from reducing the computation complexity of the RVI algorithm,

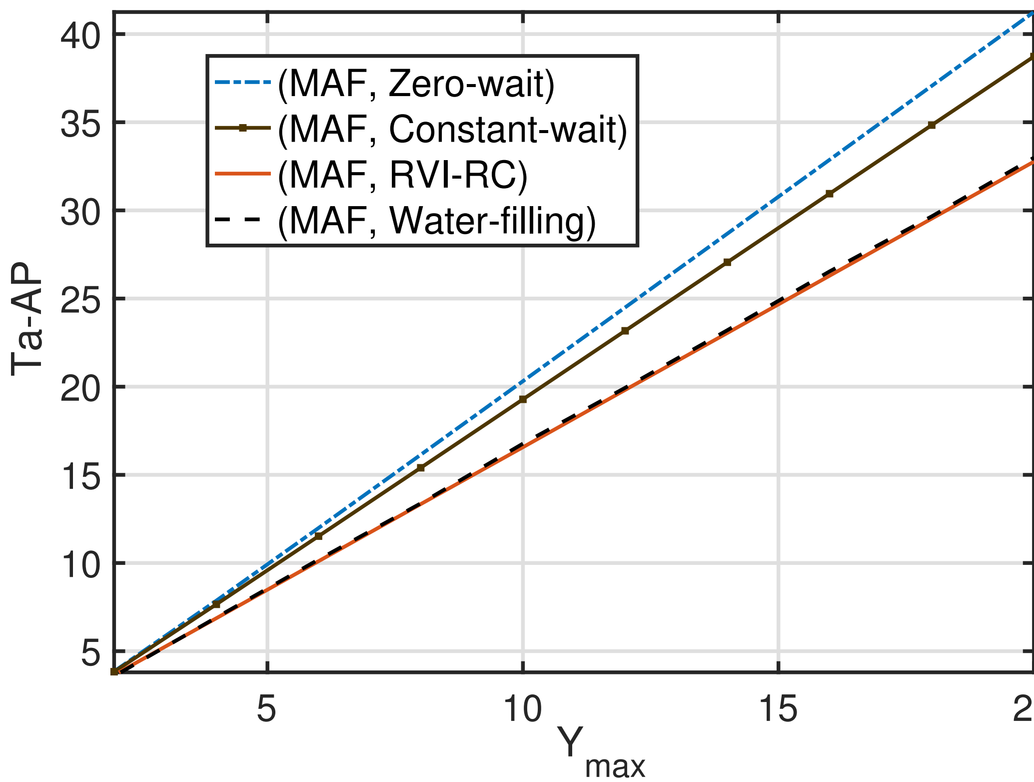


Figure 4.9: Ta-AP versus the maximum service time $Y_{\max}$ for an update system with $m = 3$ sources, where $g(x) = x$.

are jointly optimal for minimizing the Ta-AP. Finally, we devised a low-complexity threshold sampler via an approximate analysis of Bellman's equation. This threshold sampler is further simplified to a simple water-filling sampler in the special case of linear age-penalty function. The numerical results showed that the performance of these approximated samplers is almost the same as that of the RVI-RC sampler.

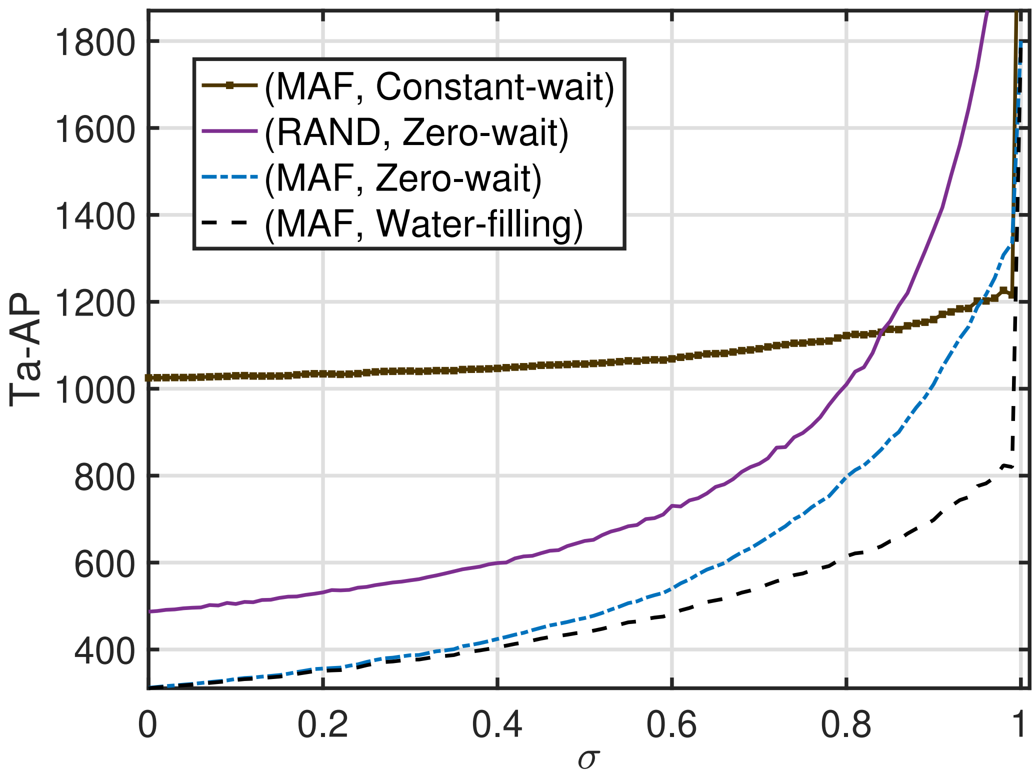


Figure 4.10: Ta-AP versus the parameter σ of the transmission time Markov chain for an update system with $m = 10$ sources, where $g(x) = x$.

CHAPTER 5

LOW-POWER STATUS UPDATES VIA SLEEP-WAKE SCHEDULING

5.1 Introduction

In this chapter, we consider the problem of optimizing the freshness of status updates that are sent from a large number of low-power source nodes to a common access point. The source nodes utilize carrier sensing to reduce collisions and adopt an asynchronized sleep-wake scheduling strategy to achieve an extended battery lifetime (e.g., 10-15 years). We use age of information (AoI) to measure the freshness of status updates, and design the sleep-wake parameters for minimizing the weighted-sum peak AoI of the sources, subject to per-source battery lifetime constraints.

Designing scheduling policies for minimizing AoI in multi-source networks has recently received increasing attention, e.g., [26–35]. Of particular interest are those pertaining to designing distributed scheduling policies [26–31]. The work in [26] considered a slotted ALOHA-like random access scheme in which each node accesses the channel with a certain access probability. These probabilities were then optimized in order to minimize the AoI. However, the model of [26] allows multiple interfering users to gain channel access simultaneously, and hence allows for the collision. The authors in [27] generalized the work in [26] to a wireless network in which the interference is described by a general interference model. The Round Robin or Maximum

Age First policy was shown to be (near) age-optimal for different system models, e.g., in [28–31, 36]. The aforementioned studies focused exclusively on minimizing the AoI and overlooked the need to reduce power consumption. This motivates us to derive scheduling algorithms that achieve a trade-off between the competing tasks of minimizing AoI and reducing the energy consumption in multi-source networks.

Carrier sensing distributed medium access mechanisms, e.g., Carrier Sense Multiple Access (CSMA), have been widely adopted in many wireless networks; see [75] for a recent survey. There has been an interest in designing CSMA-based scheduling schemes that optimize the AoI [76, 77]. In [76], the authors designed an idealized CSMA (similar to that in [78]) to minimize the AoI with an exponentially distributed packet transmission times. In [77], the authors designed a slotted Carrier Sense Multiple Access/Collision-Avoidance (CSMA/CA) (similar to that in [79]) to minimize the broadcast age of information, which is defined, from a sender's perspective, as the age of the freshest successfully broadcasted packet. Contrary to these works, the sleep-wake scheduling scheme proposed by us emphasizes on reducing the cumulative energy consumption in multi-source networks in addition to minimizing the total weighted AoI. Moreover, in our study, transmission times are not necessarily random variables with some commonly used parametric density [76], or deterministic [77], but can be any generally distributed random variables with finite mean. Our key contributions in this chapter are summarized as follows:

- In our model, sources utilize an asynchronized sleep-wake scheduling strategy to achieve an extended battery lifetime. We aim at designing the mean sleeping period of each source, which controls its channel access probability, in order to minimize the total weighted average peak age of the sources while simultaneously meeting per-source battery lifetime constraints. Although, the aforementioned optimization problem is non-convex, we devise a solution. In the regime for which the sensing time is negligible compared to the packet transmission time, the proposed solution is near-optimal (Theorem 5.3.1 and Theorem 5.3.2). Our near-optimality results hold for general distributions of the packet transmission times.

- We propose an algorithm that can be easily implemented in many practical control systems. In particular, our solution requires the knowledge of only two variables in its implementation. These two variables are functions of the network parameters. An implementation procedure to compute these two variables is provided.

- As the ratio between the sensing time and the packet transmission time reduces to zero, we show that the age performance of our proposed algorithm is as good as that of the optimal synchronized scheduler (e.g., for time-slotted systems).

5.2 Model and Formulation

This section describes the sleep-wake scheduling model under consideration as well as the energy model.

5.2.1 Network Model and Sleep-wake Scheduling

Consider a wireless network composed of M source nodes, each observing a time-varying signal. The sources generate update packets of the observed signals and send the packets to an access point (AP) over a shared spectrum band. If multiple sources transmit packets simultaneously, a packet collision occurs and these packet transmissions fail.

The sources use a sleep-wake scheduling scheme to access the shared spectrum, where each source switches between a sleep mode and a transmission mode over

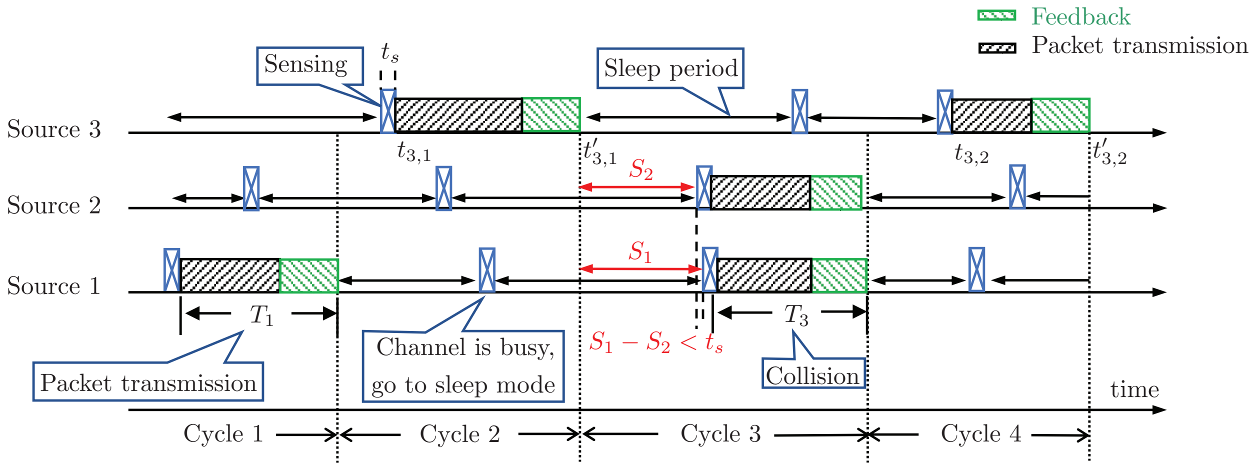


Figure 5.1: Illustration of the sleep-wake cycles. In Cycles 1-2, we have successful packet transmissions. Let S_1 and S_2 represent the remaining sleeping times of Sources 1 and 2, respectively, after a successful transmission. Then, a collision occurs in Cycle 3 because the difference between wake-up times of Sources 1 and 2 is less than t_s, i.e., $S_1 - S_2 < t_s$. As we can observe, each cycle consists of an idle period before a transmission/collision event.

time, according the following rules: Upon waking from the sleep mode, a source first performs carrier sensing to check whether the channel is occupied by another source, as illustrated in Figure 5.1. The time duration of carrier sensing is denoted as t_s, which is sufficiently long to ensure a high sensing accuracy. If the channel is sensed to be busy, the source enters the sleep mode directly; otherwise, the source generates an update packet and sends it over the channel. The source hereafter goes back to the sleep mode.

In the above sleep-wake scheduling scheme, if two sources start transmitting within a time duration of t_s, then their sensing periods are overlapping and they may not be able to detect the transmission of each other. In order to obtain a robust system design, we consider that they cannot detect each other's transmission and a collision occurs. Upon completing a packet transmission, sources switch to the reception mode and wait for an acknowledgement (ACK) that indicates the outcome of their transmissions (successful transmission or collision). They then go back to the sleep mode.

A *sleep-wake cycle*, or simply a *cycle*, is defined as the time period between the ends of two successive packet transmission or collision events. Each cycle consists of an idle period and a transmission/collision period[1]. As depicted in Figure 5.1, the packet transmissions in Cycles 1-2 are successful, but a collision occurs in Cycle 3 because Sources 1 and 2 wake up within a short duration t_s.

We use $T_j, j \in \{1, 2, \ldots\}$ to represent the time incurred by the j-th packet transmission or collision event, which includes transmission/collision time and feedback delays. For example, in Figure 5.1, T_1 is the time duration of the packet transmission event by Source 1, while T_3 is the time duration of the collision event between Source 1 and 2. We assume that the T_j's are *i.i.d.* for all transmission and collision events, with a general distribution. This assumption does not hold in practice. Nonetheless, NS-3 simulation results in Section 5.5.2 show that this assumption has a negligible impact on the performance of the proposed algorithm. When there is no confusion, we omit the subscript j of T_j for simplicity, and use T to denote the transmission/collision time, which is assumed to have a finite mean, i.e., $E[T] < \infty$. The sleep periods of source l are exponentially distributed random variables with mean value $\mathbb{E}[T]/r_l$ and are independent across sources and *i.i.d.* across time. Notice that, the sleep period parameter $r_l > 0$ has been normalized by the mean transmission time $\mathbb{E}[T]$. Let $\mathbf{r} = (r_1, \ldots, r_M)$ be the vector comprising of these sleep period parameters.

[1] To make the sleep-wake scheduling problem solvable analytically, we make several approximations. For example, in 802.11b frame structure, there exists a Short Inter-frame Space (SIFS) between the packet transmission frame and the ACK frame (i.e., the CTS frame). If another source wakes up during the SIFS, then it may not detect the transmission/ACK frames, leading to unexpected collisions. In our analytical model, such collision events are omitted. In other words, we suppose that each cycle must start with an idle period, where all sources are in the sleep mode, followed by a transmission/collision period. NS-3 simulation results will be provided in Section 5.5.2 to show that these approximations have a negligible impact on the age performance of our solution.

5.2.2 Total Weighted Average Peak Age

Let $U_l(t)$ represent the generation time of the most recently delivered packet from source l by time t. Then, the *age of information*, or simply the *age*, of source l is defined as [4]

$$\Delta_l(t) = t - U_l(t), \tag{5.2.1}$$

where $\Delta_l(t)$ is right-continuous. As shown in Figure 5.2, the age increases linearly with t, but is reset to a smaller value upon the delivery of a fresher packet. Observe that a small age $\Delta_l(t)$ indicates that the AP has a fresh status update packet that was generated at source l recently. Hence, it is desirable to keep $\Delta_l(t)$ small for all the sources.

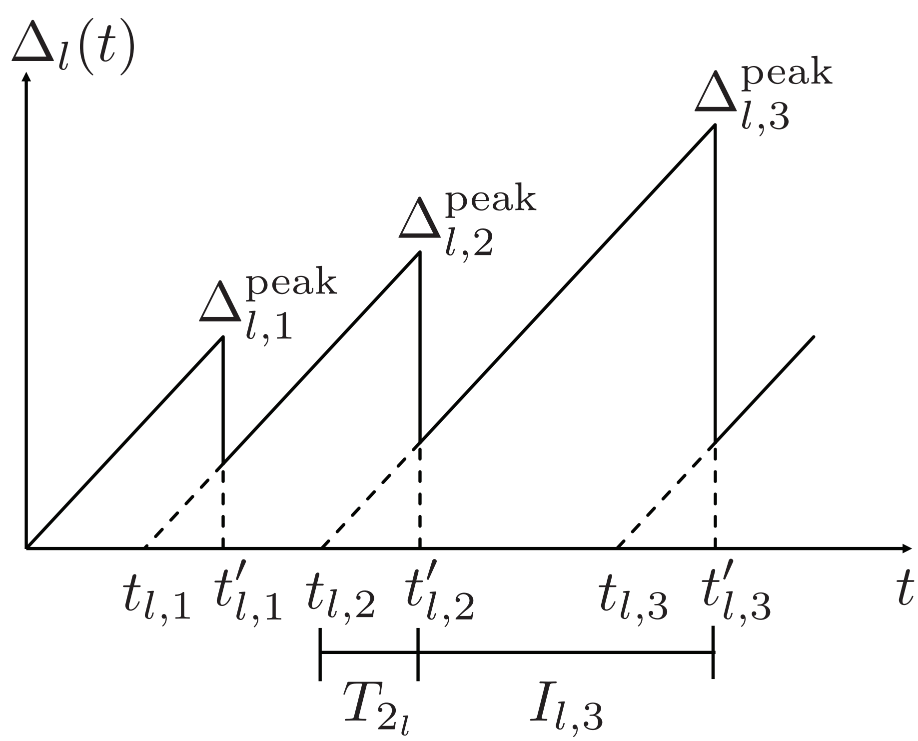


Figure 5.2: The age $\Delta_l(t)$ of source l.

Let us introduce some notations and definitions. Let i_l be the index of the i-th delivered packet from source l. We use $t_{l,i}$ and $t'_{l,i}$ to denote the generation and delivery times, respectively, of the i-th delivered packet from source l, such that $t'_{l,i} - t_{l,i} = T_{i_l}$.[2] Let $I_{l,i} = t'_{l,i} - t'_{l,i-1}$ denote the i-th inter-departure time of source l, which satisfies $\mathbb{E}[I_{l,i}] = \mathbb{E}[I_l]$ for all i. The i-th peak age of source l, denoted by $\Delta^{\text{peak}}_{l,i}$, is defined as the AoI of source l right before the i-th packet delivery from source l. As shown in Figure 5.2, i.e., we have

$$\Delta^{\text{peak}}_{l,i} = \Delta_l(t'^{-}_{l,i}), \tag{5.2.2}$$

where $t'^{-}_{l,i}$ is the time instant just before the delivery time $t'_{l,i}$. One can observe from Figure 5.2 that the peak age is [7]

$$\Delta^{\text{peak}}_{l,i} = T_{(i-1)_l} + I_{l,i}. \tag{5.2.3}$$

Hence, the average peak age of source l is given by

$$\mathbb{E}[\Delta^{\text{peak}}_{l}] = \mathbb{E}[T] + \mathbb{E}[I_l], \tag{5.2.4}$$

where we omit the subscripts i and i_l as $I_{l,i}$'s and T_{i_l}'s are *i.i.d.* across time. The average peak age metric provides information regarding the worst case age, with the advantage of having a simpler formulation than the average age metric [7]. Thus, it is suitable for applications that have an upper bound restriction on AoI.

We now derive an expression for $\mathbb{E}[I_l]$. Let α_l be the probability of the event that the source l obtains channel access and successfully transmits a packet within a sleep-wake cycle. As shown in [39], one can utilize the memoryless property of exponential

[2] A packet of a particular source is deemed delivered when the source receives the feedback.

distributed sleep periods to get

$$\alpha_l = \frac{r_l e^{r_l \frac{t_s}{\mathbb{E}[T]}}}{e^{\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}} \sum_{i=1}^{M} r_i}. \tag{5.2.5}$$

To keep the chapter self-contained, we provide the derivation of (5.2.5) in Appendix D.1. Let N_l denote the total number of sleep-wake cycles between two subsequent successful transmissions of source l. Because the probability that source l obtains channel access and transmits successfully in a given cycle is α_l, N_l is geometrically distributed with mean $\frac{1}{\alpha_l}$. By this and (5.2.5), we get

$$\mathbb{E}[N_l] = \frac{e^{\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}} \sum_{i=1}^{M} r_i}{r_l e^{r_l \frac{t_s}{\mathbb{E}[T]}}}. \tag{5.2.6}$$

An inter-departure time duration of source l is composed of N_l consecutive sleep-wake cycles. With a slight abuse of notation, let $\mathbf{cycle}_{l,k}$ denote the duration of the k-th sleep-wake cycle after a successful transmission of source l. Hence,

$$\mathbb{E}[I_l] = \mathbb{E}\left[\sum_{k=1}^{N_l} \mathbf{cycle}_{l,k}\right]. \tag{5.2.7}$$

Note that $\mathbf{cycle}_{l,k}$'s are *i.i.d.* across time. Moreover, since the event $(N_l = n)$ depends only on the history, N_l is a stopping time [80]. Hence, it follows from Wald's identity [81] that

$$\mathbb{E}[I_l] = \mathbb{E}[N_l]\mathbb{E}[\mathbf{cycle}], \tag{5.2.8}$$

where $\mathbb{E}[\mathbf{cycle}]$ is the mean duration of a sleep-wake cycle. Each cycle consists of an idle period and a transmission/collision time, see Figure 5.1. Using the memoryless property of exponential distribution, we observe that the idle period is the minimum of *i.i.d.* exponential random variables. Thus, it can be shown that the idle period in each cycle is exponentially distributed with mean value equal to $\mathbb{E}[T]/\sum_{i=1}^{M} r_i$, where $\mathbb{E}[T]/r_l$ is the mean of sleep periods of source l. Hence, we have

$$\mathbb{E}[\mathbf{cycle}] = \frac{\mathbb{E}[T]}{\sum_{i=1}^{M} r_i} + \mathbb{E}[T]. \tag{5.2.9}$$

Substituting the expressions for $\mathbb{E}[N_l]$ and $\mathbb{E}[\mathbf{cycle}]$ from (5.2.6) and (5.2.9), respectively, into (5.2.8), and (5.2.4), we obtain

$$\mathbb{E}[\Delta_l^{\text{peak}}] = \frac{e^{-r_l \frac{t_s}{\mathbb{E}[T]}}\mathbb{E}[T]}{r_l} e^{\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}} \left(1 + \sum_{i=1}^{M} r_i\right) + \mathbb{E}[T]. \tag{5.2.10}$$

In this chapter, we aim to minimize the total weighted average peak age, which is given by

$$\sum_{l=1}^{M} w_l \mathbb{E}[\Delta_l^{\text{peak}}] = \sum_{l=1}^{M} \frac{w_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}}\mathbb{E}[T]}{r_l} e^{\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}} \left(1 + \sum_{i=1}^{M} r_i\right) + \sum_{l=1}^{M} w_l \mathbb{E}[T], \tag{5.2.11}$$

where $w_l > 0$ is the weight of source l. These weights enable us to prioritize the sources according to their relative importance [27, 33].

5.2.3 Energy Constraint

Each source is equipped with a battery that can possibly be recharged by a renewable energy source, such as solar. In typical wireless sensor networks, sources have a much smaller power consumption in the sleep mode than in the transmission mode. For example, if the sensor is equipped with the radio unit TR 1000 from RF Monolithic [82, 83], the power consumption in the sleep mode is 15 μW while the power consumption in the transmission mode is 24.75 mW. Motivated by this, we assume that the energy dissipation during the sleep mode is negligible as compared to the power consumption in the transmission mode. Moreover, we assume that the sensing time duration t_s is much shorter than the transmission time and hence neglect the energy consumed during channel sensing. In Section 5.5.2, we show that these assumptions have a negligible effect on the performance of the proposed sleep-wake scheduling algorithm. Under these assumptions, the amount of energy used by a source is equal to the amount of energy consumed in packet transmissions and feedback receptions.

The energy constraint on source l is described by the following parameters: a) Initial battery level B_l, which denotes the initial amount of energy stored in the battery, b) Target lifetime D_l, which is the minimum time-duration that the source l should be active before its battery is depleted, c) Average energy replenishment rate[3] R_l, which is the rate at which the battery of source l receives energy from its energy source. If source l does not have access to an energy source, then we have $R_l = 0$. Define $P_{\max,l}$ for source l as

$$P_{\max,l} = \frac{B_l}{D_l} + R_l, \ \forall l, \tag{5.2.12}$$

where $P_{\max,l}$ is the maximum allowable power consumption of source l such that the target lifetime D_l is met.

For the sleep-wake scheduling mechanism under consideration, it has been shown in [39] that the fraction of time in which source l is in the transmission mode is given by

$$\sigma_l = \frac{[1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}] \sum_{i=1}^{M} r_i + r_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}}}{\sum_{i=1}^{M} r_i + 1}. \tag{5.2.13}$$

[3]It is assumed that R_l is either known, or it can be estimated accurately.

For the sake of completeness, the derivation of σ_l is provided in Appendix D.2. Let $P_{\text{avg},l}$ denote the average power consumption of source l in the transmission mode. Then the actual power consumption of source l, denoted by $P_{\text{act},l}$, is given by

$$P_{\text{act},l} = \sigma_l P_{\text{avg},l}, \ \forall l. \tag{5.2.14}$$

For source l to achieve its target lifetime D_l, we must have

$$P_{\text{act},l} \leq P_{\max,l}, \ \forall l. \tag{5.2.15}$$

Define $b_l \triangleq P_{\max,l}/P_{\text{avg},l}$ as the target power efficiency of source l. By using (5.2.13)-(5.2.14), the constraints in (5.2.15) can be rewritten as

$$\sigma_l = \frac{[1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}] \sum_{i=1}^{M} r_i + r_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}}}{\sum_{i=1}^{M} r_i + 1} \leq b_l, \ \forall l. \tag{5.2.16}$$

Because $\sigma_l \leq 1$, if $b_l \geq 1$, then constraint (5.2.16) is always satisfied.

5.2.4 Problem Formulation

Our goal is to find the optimal sleep-wake parameters $\mathbf{r}$ that minimizes the total weighted average peak age in (5.2.11), while simultaneously ensuring the energy constraints (5.2.16) for all sources. Dividing the objective function (5.2.11) by $\mathbb{E}[T]$, we obtain the following optimization problem: (Problem **1**)

$$\begin{aligned} \bar{\Delta}_{\text{opt}}^{\text{w-peak}} \triangleq \min_{r_l > 0} \sum_{l=1}^{M} \frac{w_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}}}{r_l} e^{\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}} \left(1 + \sum_{i=1}^{M} r_i\right) + \sum_{l=1}^{M} w_l \\ \textbf{s.t.} \ \frac{[1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}] \sum_{i=1}^{M} r_i + r_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}}}{\sum_{i=1}^{M} r_i + 1} \leq b_l, \forall l, \end{aligned} \tag{5.2.17}$$

where $\bar{\Delta}_{\text{opt}}^{\text{w-peak}}$ is the optimal objective value of Problem **1**. We will use $\bar{\Delta}^{\text{w-peak}}(\mathbf{r})$ to denote the objective value for given sleeping period parameters $\mathbf{r}$. One can notice from (5.2.17) that the optimal sleeping period parameters depend on the sensing time t_s and the mean transmission time $\mathbb{E}[T]$ only through their ratio $t_s/\mathbb{E}[T]$. This insight plays a crucial role in subsequent analysis of Problem **1**.

5.3 Main Results

When $t_s = 0$, although Problem **1** is non-convex, it can be solved by defining an auxiliary variable $y = \sum_{i=1}^{M} r_i + 1$ and applying a nested optimization algorithm: In the inner layer, we optimize r_l for a given y. Then, we write the optimized objective as a function of y. In the outer layer, we optimize y. It happens that the inner and outer layer optimization problems are both convex. The details can be found in Section 5.3.3.

However, this method does not work for positive sensing times $t_s > 0$ and Problem **1** becomes non-convex. Hence, it is challenging to optimize $\mathbf{r}$ for positive t_s. In this section, we develop a low-complexity closed-form solution which is shown to be near-optimal if the sensing time t_s is short as compared with the mean transmission time $\mathbb{E}[T]$. Our solution is developed by considering the following two regimes separately: (i) *Energy-adequate regime* denoted as $\sum_{i=1}^{M} b_i \geq 1$, where the condition $\sum_{i=1}^{M} b_i \geq 1$ means that the sources have a sufficient amount of total energy to ensure that at least one source is awake at any time, (ii) *Energy-scarce regime* represented by $\sum_{i=1}^{M} b_i < 1$, which indicates that the sources have to sleep for some time to meet the sources' energy constraints.

5.3.1 Energy-adequate Regime

In the energy-adequate regime $\sum_{i=1}^{M} b_i \geq 1$, our solution $\mathbf{r}^\star := (r_1^\star, \ldots, r_M^\star)$ is given as

$$r_l^\star = \min\{b_l, \beta^\star \sqrt{w_l}\} x^\star, \forall l, \tag{5.3.1}$$

where $x^\star$ and $\beta^\star$ are expressed in terms of the parameters $\{b_i, w_i\}_{i=1}^M, t_s/\mathbb{E}[T]$ as follows:

$$x^\star = \frac{-1}{2} + \sqrt{\frac{1}{4} + \frac{\mathbb{E}[T]}{t_s}}, \tag{5.3.2}$$

and $\beta^\star$ is the unique root of

$$\sum_{i=1}^{M} \min\{b_i, \beta^\star \sqrt{w_i}\} \ = 1. \tag{5.3.3}$$

The performance of the above solution $\mathbf{r}^\star$ is manifested in the following theorem:

Theorem 5.3.1 (Near-optimality). *If* $\sum_{i=1}^{M} b_i \geq 1$*, then the solution* $\mathbf{r}^\star$ (5.3.1) - (5.3.3) *is near-optimal for solving* (5.2.17) *when* $t_s/E[T]$ *is sufficiently small, in the following sense:*[4]

$$\left| \bar{\Delta}^{\textit{w-peak}}(\mathbf{r}^\star) - \bar{\Delta}_{\textit{opt}}^{\textit{w-peak}} \right| \leq 2\sqrt{\frac{t_s}{\mathbb{E}[T]}} C_1 + o\left(\sqrt{\frac{t_s}{\mathbb{E}[T]}}\right), \tag{5.3.4}$$

where

$$C_1 = \sum_{i=1}^{M} \frac{w_i}{\min\{b_i, \beta^\star \sqrt{w_i}\}}. \tag{5.3.5}$$

[4]We use the standard order notation: $f(h) = O(g(h))$ means $z_1 \leq \lim_{h\to 0} f(h)/g(h) \leq z_2$ for some constants $z_1 > 0$ and $z_2 > 0$, while $f(h) = o(g(h))$ means $\lim_{h\to 0} f(h)/g(h) = 0$.

Proof. See Section 5.4.1. □

From Theorem 5.3.1, we can obtain the following corollary:

Corollary 5.3.1.1 (Asymptotic optimality). *If* $\sum_{i=1}^{M} b_i \geq 1$*, then the solution* $\mathbf{r}^\star$ (5.3.1) - (5.3.3) *is asymptotically optimal for Problem* ***1*** *in* (5.2.17) *as* $t_s/\mathbb{E}[T] \to 0$*, i.e.,*

$$\lim_{\frac{t_s}{\mathbb{E}[T]} \to 0} \left| \bar{\Delta}^{\text{w-peak}}(\mathbf{r}^\star) - \bar{\Delta}^{\text{w-peak}}_{opt} \right| = 0. \tag{5.3.6}$$

Moreover, the asymptotic optimal objective value of Problem ***1*** *as* $t_s/\mathbb{E}[T] \to 0$ *is*[5]

$$\lim_{\frac{t_s}{\mathbb{E}[T]} \to 0} \bar{\Delta}^{\text{w-peak}}_{opt} = \sum_{i=1}^{M} \left[\frac{w_i}{\min\{b_i, \beta^\star \sqrt{w_i}\}} + w_i \right]. \tag{5.3.7}$$

Proof. See Section 5.4.1. □

5.3.2 Energy-scarce Regime

Now, we present a solution to Problem **1** in the energy-scarce regime $\sum_{i=1}^{M} b_i < 1$, and show it is near-optimal. The solution $\mathbf{r}^\star$ of the energy-scarce regime is again given by (5.3.1), where $x^\star$ and $\beta^\star$ are

$$x^\star = \frac{\min_l c_l}{1 - \sum_{i=1}^{M} b_i}, \; \beta^\star = \sum_{i=1}^{M} \frac{1}{\sqrt{w_i}}, \tag{5.3.8}$$

[5] Observe that, according to (5.3.7), the asymptotic optimal average peak age of source l is $(1/\min\{b_l, \beta^\star\sqrt{w_l}\} + 1)$ which decreases with the weight w_l. The weighted average peak age is $w_l(1/\min\{b_l, \beta^\star\sqrt{w_l}\} + 1)$ which increases with w_l. This phenomenon is reasonable and agrees with our expectation.

and

$$
\begin{aligned}
c_l =& \frac{2b_l\left(1-\sum_{i=1}^{M} b_i\right)^2}{Q_l}, && (5.3.9)\\
Q_l =& b_l\left(1-\sum_{i=1}^{M} b_i\right)^2 \\
&+ \sqrt{b_l^2\left(1-\sum_{i=1}^{M} b_i\right)^4 + 4b_l^2\left(1-\sum_{i=1}^{M} b_i\right)^2\left(\sum_{i=1}^{M} b_i - b_l\right)\frac{t_s}{\mathbb{E}[T]}}. && (5.3.10)
\end{aligned}
$$

The near-optimality of the proposed solution (i.e., $\mathbf{r}^\star$) in the energy scarce regime is explained in the following theorem:

Theorem 5.3.2 (**Near-optimality**). *If* $\sum_{i=1}^{M} b_i < 1$, *then the solution* $\mathbf{r}^\star$ (5.3.1) *and* (5.3.8) - (5.3.10) *is near-optimal for solving* (5.2.17) *when* $t_s/\mathbb{E}[T]$ *is sufficiently small, in the following sense:*

$$
\left|\bar{\Delta}^{\textit{w-peak}}(\mathbf{r}^\star) - \bar{\Delta}_{opt}^{\textit{w-peak}}\right| \le \frac{t_s}{\mathbb{E}[T]}C_2 + o\left(\frac{t_s}{\mathbb{E}[T]}\right), \tag{5.3.11}
$$

where

$$
C_2 = \sum_{l=1}^{M} \frac{w_l}{b_l(1-\sum_{i=1}^{M} b_i)}\left(3\sum_{i=1}^{M} b_i - \min_j b_j\right). \tag{5.3.12}
$$

Proof. See Section 5.4.2. □

We obtain the following corollary from Theorem 5.3.2.

Corollary 5.3.2.1 (**Asymptotic optimality**). *If* $\sum_{i=1}^{M} b_i < 1$, *then* (5.3.6) *holds for the solution* $\mathbf{r}^\star$ (5.3.1) *and* (5.3.8) - (5.3.10)*. In other words, our proposed solution* *is asymptotically optimal for Problem **1** in* (5.2.17) *as* $t_s/\mathbb{E}[T] \to 0$*. Moreover, the asymptotic optimal objective value of Problem **1** as* $t_s/\mathbb{E}[T] \to 0$ *is*

$$\begin{aligned}\lim_{\frac{t_s}{\mathbb{E}[T]}\to 0} \bar{\Delta}^{w\text{-}peak}_{opt} &= \sum_{i=1}^{M}\left[\frac{w_i}{\min\{b_i, \beta^\star\sqrt{w_i}\}} + w_i\right] \\ &= \sum_{i=1}^{M}\left[\frac{w_i}{b_i} + w_i\right].\end{aligned} \tag{5.3.13}$$

Proof. See Section 5.4.2. □

Interestingly, the asymptotic optimal objective values of Problem **1** in both regimes, given by (5.3.7) and (5.3.13), are of an identical expression. However, in the energy-scarce regime, we can observe that $\beta^\star$, which is defined in (5.3.8), always satisfies $\min\{b_l, \beta^\star\sqrt{w_l}\} = b_l$ for all l.

Remark 5.3.2.1. *We would like to point out that the condition* $t_s/\mathbb{E}[T] \approx 0$ *is satisfied in many practical applications. For instance, in a wireless sensor network that is equipped with low-power UHF transceivers [84], the carrier sensing time is* $t_s = 40$ μs*, while the transmission time is around* 5 *ms. Hence,* $t_s/\mathbb{E}[T] \approx 0.008$*.*

5.3.3 Discussion

In this subsection, we present a simple implementation of our proposed solution, discuss the nested convex optimization method that can be used to solve Problem **1** when $t_s = 0$, provide some useful insights about our proposed solution at the limit point $t_s/\mathbb{E}[T] \to 0$, and provide a comparison with synchronized schedulers performance.

Implementation of Sleep-wake Scheduling

We devise a simple algorithm to compute our solution $\mathbf{r}^\star$, which is provided in Algorithm 5.1. Notice that $\mathbf{r}^\star$ has the same expression (5.3.1) in the energy-adequate and energy-scarce regimes. We exploit this fact to simplify the implementation of sleep-wake scheduling. In particular, the sources report w_l and b_l to the AP, which computes $\beta^\star$ and $x^\star$, and broadcasts them back to the sources. After receiving $\beta^\star$ and $x^\star$, source l computes $r_l^\star$ based on (5.3.1). In practical wireless sensor networks, e.g., smart city networks and industrial control sensor networks [85,86], the sensors report their measurements via an access point (AP). Hence, it is reasonable to employ the AP in implementing the sleep-wake scheduler.

Algorithm 5.1: Implementation of sleep-wake scheduler.

The AP gathers the parameters $\{(w_i, b_i)_{i=1}^M, t_s/\mathbb{E}[T]\}$;
if $\sum_{i=1}^{M} b_i \geq 1$ **then**
  The AP computes $x^\star, \beta^\star$ from (5.3.2) and (5.3.3);
else
  The AP computes $x^\star, \beta^\star$ from (5.3.8) - (5.3.10);
The AP broadcasts $x^\star, \beta^\star$ to all the M sources;
Upon hearing $x^\star, \beta^\star$, source l compute $r_l^\star$ from (5.3.1);

In the above implementation procedure, the sources do not need to know if the overall network is in the energy-adequate or energy-scarce regime; only the AP knows about it. Further, the amount of downlink signaling overhead is small, because only two parameters $\beta^\star$ and $x^\star$ are broadcasted to the sources. Moreover, when the node density is high, the scalability of the network is a crucial concern and reporting w_l and b_l for each source is impractical. In this case, the AP can compute $\beta^\star$ and $x^\star$ by estimating the distribution of w_l and b_l, as well as the number of source nodes, which reduces the uplink signaling overhead. Finally, when sources are not in the hearing range of each other, hidden/exposed source problems arise. These problems are challenging to solve analytically. However, this can be solved by designing practical heuristic solutions based on the theoretical solutions. One design method was given in [39].

The Nested Convex Optimization Method for $t_s = 0$

If $t_s = 0$, Problem **1** reduces to the following optimization problem:

$$\begin{aligned}\bar{\Delta}_{\text{opt}}^{\text{w-peak}} \triangleq \min_{r_l>0} \sum_{l=1}^{M} & \frac{w_l\left(1+\sum_{i=1}^{M} r_i\right)}{r_l} + \sum_{l=1}^{M} w_l \\ \textbf{s.t. } & r_l \leq b_l(\sum_{i=1}^{M} r_i + 1), \forall l.\end{aligned} \tag{5.3.14}$$

Observe that the optimization problem in (5.3.14) is non-convex. To bypass this difficulty, we use an auxiliary variable $y = \sum_{i=1}^{M} r_i + 1$. Hence, we obtain the following optimization problem for given y:

$$\min_{r_i>0} \sum_{i=1}^{M} \left[\frac{w_i y}{r_i} + w_i\right] \tag{5.3.15}$$

$$\textbf{s.t. } r_l \leq b_l y, \forall l, \tag{5.3.16}$$

$$\sum_{i=1}^{M} r_i + 1 = y. \tag{5.3.17}$$

The objective function in (5.3.15) is a convex function. Moreover, the constraints in (5.3.16) and (5.3.17) are affine. Hence, Problem (5.3.15) is convex. Exploiting (5.3.15), we solve (5.3.14) by using a two-layer nested convex optimization method: In the inner layer, we optimize $\mathbf{r}$ for given y. After solving $\mathbf{r}$, we will optimize y in the outer layer. This technique is used in the proof of Lemma 5.4.3 in Appendix D.4, where the reader can find the detailed solution.

Asymptotic Behavior of The Optimal Solution

In the energy-adequate regime, the sleeping period parameter $r_l^\star$ of source l tends to infinity as $t_s/\mathbb{E}[T] \to 0$, while the ratio $r_l^\star/r_i^\star$ between source l and source i is kept as a constant for all l and i. Hence, the sleeping time of the sources tends to zero. Meanwhile, since $t_s/\mathbb{E}[T] \to 0$, the sensing time becomes negligible. The channel access probability of source l in this limit can be computed as

$$\lim_{\frac{t_s}{\mathbb{E}[T]} \to 0} \sigma_l^\star = \min\{b_l, \beta^\star \sqrt{w_l}\}. \tag{5.3.18}$$

Because of (5.3.3), $\lim_{t_s/\mathbb{E}[T]\to 0} \sum_{i=1}^{M} \sigma_i^\star = 1$. Hence, the channel is occupied by the sources at all time, without any time overhead spent on sensing and sleeping.

On the other hand, in the energy-scarce regime, the sleeping period parameter $r_l^\star$ of source l converges to a constant value when $t_s/\mathbb{E}[T] \to 0$, i.e., we have

$$\lim_{\frac{t_s}{\mathbb{E}[T]} \to 0} r_l^\star = \frac{b_l}{1 - \sum_{i=1}^{M} b_i}. \tag{5.3.19}$$

Since the cumulative energy is scarce, the sources necessarily need to stay idle for some time in order to meet their target lifetime. Hence, sleep periods are imposed for achieving the optimal trade-off between minimizing AoI and energy consumption.

Comparison with Synchronized Schedulers Performance

We would like to show that the performance of our proposed algorithm is asymptotically no worse than any synchronized (e.g., centralized) scheduler. Consider a scheduler in which the fraction of time during which source l transmits update packets is equal to a_l, where we have $\mathbf{a} = \{a_l\}_{l=1}^{M}$ and $\sum_{i=1}^{M} a_i \leq 1$. In this scheduler, only one source is allowed to access the channel at a time, i.e., there is no collision (this can be achieved either by a deterministic scheduler or by assigning a channel access probability a_l for each source l after each packet transmission)[6]. We can perform an analysis similar to that of Section 5.2.2, and show that the total weighted average peak age of a synchronized scheduler is given by

$$\sum_{i=1}^{M} \left[\frac{w_i \mathbb{E}[T]}{a_i} + w_i \, \mathbb{E}[T] \right]. \tag{5.3.20}$$

Hence, the problem of designing an optimal synchronized scheduler that minimizes the total weighted average peak age under energy constraints can be cast as

$$\bar{\Delta}^{\text{w-peak}}_{\text{opt-s}} \triangleq \min_{a_i > 0} \sum_{i=1}^{M} \left[\frac{w_i}{a_i} + w_i \right] \tag{5.3.21}$$

$$\textbf{s.t. } a_l \leq b_l, \ \forall l, \tag{5.3.22}$$

$$\sum_{i=1}^{M} a_i \leq 1, \tag{5.3.23}$$

where we have divided the objective function by $\mathbb{E}[T]$. Next, we show that the performance of our proposed algorithm converges to that of the optimal synchronized scheduler when $t_s/\mathbb{E}[T] \to 0$.

[6]Note that if $\sum_{i=1}^{M} a_i < 1$, then it is possible that the scheduler decides not to serve any source after the transmission of some packet. In this case, the scheduler waits for a random time that has the same distribution as the transmission time T before deciding to serve another source.

Corollary 5.3.2.2. *For any* $(w_i, b_i)_{i=1}^M$*, we have*

$$\lim_{\frac{t_s}{\mathbb{E}[T]} \to 0} \bar{\Delta}_{opt}^{w\text{-}peak} = \bar{\Delta}_{opt\text{-}s}^{w\text{-}peak}. \tag{5.3.24}$$

Proof. The proof is provided in Appendix D.7 which is listed at the end as it requires some results from precedent appendixes. □

Synchronized schedulers were recently studied in [33] for the case without energy constraints, i.e., $b_l \geq 1$ for all l. According to Corollary 5.3.2.2, the channel access probability of the synchronized scheduler in [33] is a special case of our solution (5.3.18) where $b_l \geq 1$ for all l.

5.4 Proofs of the Main Results

In this section, we provide the proofs of Theorem 5.3.1, Corollary 5.3.1.1, Theorem 5.3.2, and Corollary 5.3.2.1.

5.4.1 The Proofs of Theorem 5.3.1 and Corollary 5.3.1.1

We prove Theorem 5.3.1 and Corollary 5.3.1.1 in three steps:

Step 1: We show that our solution $\mathbf{r}^\star$ (5.3.1) - (5.3.3) is feasible for Problem **1**.

Lemma 5.4.1. *If* $\sum_{i=1}^M b_i \geq 1$*, then the solution* $\mathbf{r}^\star$ (5.3.1) - (5.3.3) *is feasible for Problem **1**.*

Proof. See Appendix D.3. □

Hence, by substituting this solution $\mathbf{r}^\star$ into the objective function of Problem **1** in (5.2.17), we get an upper bound on the optimal value $\bar{\Delta}_{\text{opt}}^{\text{w-peak}}$, which is expressed in the following lemma:

Lemma 5.4.2. *If* $\sum_{i=1}^{M} b_i \geq 1$*, then*

$$\bar{\Delta}_{opt}^{w\text{-}peak} \leq \bar{\Delta}^{w\text{-}peak}(\mathbf{r}^\star) \leq \sum_{i=1}^{M} \left[\frac{w_i e^{x^\star \frac{t_s}{\mathbb{E}[T]}} \left(1 + \frac{1}{x^\star}\right)}{\min\{b_i, \beta^\star \sqrt{w_i}\}} + w_i \right], \tag{5.4.1}$$

where $x^\star$*,* $\beta^\star$ *are defined in* (5.3.2), (5.3.3)*.*

Proof. In Lemma 5.4.1, we showed that our proposed solution $\mathbf{r}^\star$ (5.3.1) - (5.3.3) is feasible for Problem **1**. Hence, we substitute this solution into Problem **1** to obtain the following upper bound:

$$\sum_{i=1}^{M} \left[\frac{w_i e^{x^\star \frac{t_s}{\mathbb{E}[T]}} \left(1 + \frac{1}{x^\star}\right) e^{-\min\{b_i, \beta^\star \sqrt{w_i}\} x^\star \frac{t_s}{\mathbb{E}[T]}}}{\min\{b_i, \beta^\star \sqrt{w_i}\}} + w_i \right]. \tag{5.4.2}$$

Next, we replace $e^{-\min\{b_i, \beta^\star \sqrt{w_i}\} x^\star (t_s/\mathbb{E}[T])}$ by 1 to derive another upper bound with a simple expression, which is given by (5.4.1). This completes the proof. □

Step 2: We now construct a lower bound on the optimal value of Problem **1**. Suppose that $\mathbf{r} = (r_1, \ldots, r_M)$ is a feasible solution to Problem **1**, such that $r_l > 0$ and

$$\frac{[1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}] \sum_{i=1}^{M} r_i + r_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}}}{\sum_{i=1}^{M} r_i + 1} \leq b_l, \forall l. \tag{5.4.3}$$

Because $[1 - e^{-r_l (t_s/\mathbb{E}[T])}] \sum_{i=1}^{M} r_i + r_l e^{-r_l (t_s/\mathbb{E}[T])} > r_l$ for all l, $\mathbf{r}$ satisfies $r_l / (\sum_{i=1}^{M} r_i + 1) \leq b_l$. Hence, the following Problem **2** has a larger feasible set than Problem **1**: (Problem

2)

$$\bar{\Delta}_{\text{opt},2}^{\text{w-peak}} \triangleq \min_{r_l>0} \sum_{l=1}^{M} \frac{w_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}}}{r_l} e^{\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}} \left(1 + \sum_{i=1}^{M} r_i\right) + \sum_{l=1}^{M} w_l \tag{5.4.4}$$

$$\textbf{s.t. } r_l \leq b_l \left(\sum_{i=1}^{M} r_i + 1\right), \forall l, \tag{5.4.5}$$

where $\bar{\Delta}_{\text{opt},2}^{\text{w-peak}}$ is the optimal value of Problem **2**. The optimal objective value of Problem **2** is a lower bound of that of Problem **1**. We note that the constraint set corresponding to Problem **2** is convex. Thus, this relaxation converts the constraint set of Problem **1** to a convex one, and hence enables us to obtain a lower bound for the optimal value of Problem **1**, which is expressed in the following lemma:

Lemma 5.4.3. *If* $\sum_{i=1}^{M} b_i \geq 1$*, then*

$$\bar{\Delta}_{\textit{opt}}^{\textit{w-peak}} \geq \bar{\Delta}_{\textit{opt},2}^{\textit{w-peak}} \geq \sum_{i=1}^{M} \left[\frac{w_i}{\min\{b_i, \beta^\star \sqrt{w_i}\}} + w_i\right], \tag{5.4.6}$$

where $\beta^\star$ *is the root of* (5.3.3).

Proof. See Appendix D.4. □

Step 3: After the upper and lower bounds of $\bar{\Delta}_{\text{opt}}^{\text{w-peak}}$ were derived in Steps 1-2, we are ready to analysis their gap. By combining (5.4.1) and (5.4.6), the sub-optimality gap of the solution $\mathbf{r}^\star$ (5.3.1) - (5.3.3) is upper bounded by

$$\left|\bar{\Delta}^{\text{w-peak}}(\mathbf{r}^\star) - \bar{\Delta}_{\text{opt}}^{\text{w-peak}}\right| \leq \sum_{i=1}^{M} \frac{w_i \left(e^{x^\star \frac{t_s}{\mathbb{E}[T]}}(1+\frac{1}{x^\star})-1\right)}{\min\{b_i, \beta^\star \sqrt{w_i}\}}, \tag{5.4.7}$$

where $x^\star$, $\beta^\star$ are defined in (5.3.2), (5.3.3). Next, we characterize the right-hand-side (RHS) of (5.4.7) by Taylor expansion. For simplicity, let $\epsilon = \dfrac{t_s}{\mathbb{E}[T]}$. Using the expression for $x^\star$ from (5.3.2), we have

$$x^\star \epsilon = -\frac{\epsilon}{2} + \sqrt{\frac{\epsilon^2}{4} + \epsilon} = \frac{\epsilon}{\frac{\epsilon}{2} + \sqrt{\frac{\epsilon^2}{4} + \epsilon}} = \sqrt{\epsilon} + o(\sqrt{\epsilon}). \tag{5.4.8}$$

Moreover,

$$x^\star = -\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{\epsilon}} = \frac{\frac{1}{\epsilon}}{\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{\epsilon}}} = \frac{1}{\sqrt{\epsilon}} + o\left(\frac{1}{\sqrt{\epsilon}}\right). \tag{5.4.9}$$

Substituting (5.4.8) and (5.4.9) in (5.4.7), we obtain

$$\begin{aligned}
\left|\bar{\Delta}^{\text{w-peak}}(\mathbf{r}^\star) - \bar{\Delta}^{\text{w-peak}}_{\text{opt}}\right| &\le \sum_{i=1}^{M} \frac{w_i[e^{\sqrt{\epsilon}+o(\sqrt{\epsilon})}(1+\sqrt{\epsilon}+o(\sqrt{\epsilon}))-1]}{\min\{b_i, \beta^\star\sqrt{w_i}\}} \\
&= \sum_{i=1}^{M} \frac{w_i[(1+\sqrt{\epsilon}+o(\sqrt{\epsilon}))(1+\sqrt{\epsilon}+o(\sqrt{\epsilon}))-1]}{\min\{b_i, \beta^\star\sqrt{w_i}\}} \\
&= 2\sqrt{\epsilon}\sum_{i=1}^{M} \frac{w_i}{\min\{b_i, \beta^\star\sqrt{w_i}\}} + o(\sqrt{\epsilon}),
\end{aligned} \tag{5.4.10}$$

where the second inequality involves the use of Taylor expansion. This proves Theorem 5.3.1.

We can observe that the gap $\left|\bar{\Delta}^{\text{w-peak}}(\mathbf{r}^\star) - \bar{\Delta}^{\text{w-peak}}_{\text{opt}}\right|$ in the energy-adequate regime converges to zero at a speed of $O(\sqrt{\epsilon})$, as $\epsilon \to 0$. Further, both the upper and lower bounds (5.4.1), (5.4.6), converge to $\sum_{i=1}^{M}[(w_i/\min\{b_i, \beta^\star\sqrt{w_i}\}) + w_i]$ as $t_s/\mathbb{E}[T] \to 0$. Thus, this value is the asymptotic optimal objective value of Problem **1**. This proves Corollary 5.3.1.1.

5.4.2 The Proofs of Theorem 5.3.2 and Corollary 5.3.2.1

Similar to Section 5.4.1, we prove Theorem 5.3.2 and Corollary 5.3.2.1 also in three steps:

Step 1: We show that the proposed solution $\mathbf{r}^\star$ (5.3.1) and (5.3.8) - (5.3.10) is a feasible solution for Problem **1**.

Lemma 5.4.4. *If* $\sum_{i=1}^{M} b_i < 1$, *then the solution* $\mathbf{r}^\star$ (5.3.1) *and* (5.3.8) - (5.3.10) *is feasible for Problem* ***1***.

Proof. See Appendix D.5. □

Now, we construct an upper bound on the optimal value of Problem **1** using our proposed solution as follows:

Lemma 5.4.5. *If* $\sum_{i=1}^{M} b_i < 1$, *then*

$$\begin{aligned}\bar{\Delta}_{opt}^{w\text{-}peak} \leq \bar{\Delta}^{w\text{-}peak}(\mathbf{r}^\star) \leq \sum_{l=1}^{M} \frac{w_l}{b_l} e^{\sum_{i=1}^{M} b_i x^\star \frac{t_s}{\mathbb{E}[T]}} \left(\frac{1}{x^\star} + \sum_{i=1}^{M} b_i \right) \\ + \sum_{l=1}^{M} w_l,\end{aligned} \tag{5.4.11}$$

where $x^\star$ *is defined in* (5.3.8).

Proof. In Lemma 5.4.4, we showed that our proposed solution $\mathbf{r}^\star$ (5.3.1) and (5.3.8) - (5.3.10) is feasible for Problem **1**. Hence, we substitute this solution into Problem **1** to obtain the following upper bound:

$$\sum_{l=1}^{M} \frac{w_l e^{-b_l x^\star \frac{t_s}{\mathbb{E}[T]}}}{b_l} e^{\sum_{i=1}^{M} b_i x^\star \frac{t_s}{\mathbb{E}[T]}} \left(\frac{1}{x^\star} + \sum_{i=1}^{M} b_i \right) + \sum_{l=1}^{M} w_l. \tag{5.4.12}$$

Next, we replace $e^{-b_l x^\star \frac{t_s}{\mathbb{E}[T]}}$ by 1 to derive another upper bound with a simple expression, which is given by (5.4.11). This completes the proof. □

Step 2: Similar to the proof in Section 5.4.1, we use the relaxed problem, Problem **2**, to construct a lower bound as follows:

Lemma 5.4.6. *If* $\sum_{i=1}^{M} b_i < 1$, *then*

$$\bar{\Delta}_{opt}^{w\text{-}peak} \geq \bar{\Delta}_{opt,2}^{w\text{-}peak} \geq \sum_{l=1}^{M} \frac{w_l}{b_l} e^{\frac{-\sum_{i=1}^{M} b_i}{1-\sum_{i=1}^{M} b_i} \frac{t_s}{\mathbb{E}[T]}} + \sum_{l=1}^{M} w_l. \tag{5.4.13}$$

Proof. See Appendix D.6. □

Step 3: We now characterize the sub-optimality gap by analyzing the upper and lower bounds constructed above. By combining (5.4.11) and (5.4.13), the sub-optimality gap of the solution $\mathbf{r}^\star$ (5.3.1) and (5.3.8) - (5.3.10) is upper bounded by

$$\begin{aligned}
&\left|\bar{\Delta}^{\text{w-peak}}(\mathbf{r}^\star) - \bar{\Delta}_{\text{opt}}^{\text{w-peak}}\right| \\
&\leq \sum_{l=1}^{M} \frac{w_l}{b_l} \left[e^{\sum_{i=1}^{M} b_i x^\star \frac{t_s}{\mathbb{E}[T]}} \left(\frac{1}{x^\star} + \sum_{i=1}^{M} b_i \right) - e^{\frac{-\sum_{i=1}^{M} b_i}{1-\sum_{i=1}^{M} b_i} \frac{t_s}{\mathbb{E}[T]}} \right].
\end{aligned} \tag{5.4.14}$$

where $x^\star$ is defined in (5.3.8). Next, we characterize the RHS of (5.4.14) by Taylor expansion. For simplicity, let $\epsilon = t_s/\mathbb{E}[T]$, $Z = (\sum_{i=1}^{M} b_i)/(1 - \sum_{i=1}^{M} b_i)$, and $k_l = (\sum_{i=1}^{M} b_i - b_l)/(1 - \sum_{i=1}^{M} b_i)^2$. Using Taylor expansion, we are able to obtain the following:

$$\min_l c_l = 1 + \left(\min_l k_l \right) \epsilon + o(\epsilon), \tag{5.4.15}$$

$$\frac{1}{\min_l c_l} = \max_l \frac{1}{c_l} = 1 + \left(\max_l k_l \right) \epsilon + o(\epsilon). \tag{5.4.16}$$

Using (5.4.15), (5.4.16), $x^\star$ from (5.3.8), and Taylor expansion again, we get

$$
\begin{aligned}
e^{\sum_{i=1}^M b_i x^\star \epsilon} &= 1 + Z\left(1 + \left(\min_l k_l\right)\epsilon + o(\epsilon)\right)\epsilon + o(\epsilon) \\
&= 1 + Z\epsilon + o(\epsilon),
\end{aligned} \tag{5.4.17}
$$

$$
\begin{aligned}
\frac{1}{x^\star} + \sum_{i=1}^M b_i &= \frac{1 - \sum_{i=1}^M b_i}{\min_l c_l} + \sum_{i=1}^M b_i \\
&= 1 + \left(\max_l k_l\right)\left(1 - \sum_{i=1}^M b_i\right)\epsilon + o(\epsilon),
\end{aligned} \tag{5.4.18}
$$

$$
e^{-Z\epsilon} = 1 - Z\epsilon + o(\epsilon). \tag{5.4.19}
$$

Substituting (5.4.17) - (5.4.19) into (5.4.14), we get (5.3.11). This proves Theorem (5.3.2).

Moreover, we observe that the gap $\left|\bar{\Delta}^{\text{w-peak}}(\mathbf{r}^\star) - \bar{\Delta}^{\text{w-peak}}_{\text{opt}}\right|$ in the energy-scarce regime converges to zero at a speed of $O(\epsilon)$, as $\epsilon \to 0$. Further, both the upper and lower bounds (5.4.11), (5.4.13), converge to $\sum_{i=1}^M [(w_i/b_i) + w_i]$ as $t_s/\mathbb{E}[T] \to 0$. Thus, this value is the asymptotic optimal objective value of Problem **1**. This proves Corollary 5.3.2.1.

5.5 Numerical and Simulation Results

We use Matlab and NS-3 to evaluate the performance of our algorithm. We use "age-optimal scheduler" to denote the sleep-wake scheduler with the sleep period paramters $r_l^\star$'s as in (5.3.1), which was shown to be near-optimal in Theorem 5.3.1 and Theorem 5.3.2. By "throughput-optimal scheduler", we refer to the sleep-wake algorithm of [39] that is known to achieve the optimal trade-off between the throughput and energy consumption reduction. Moreover, we use "fixed sleep-rate scheduler" to denote the sleep-wake scheduler in which the sleep period parameters r_l's are equal for all the sources, i.e., $r_l = k$ for all l, where the parameter k has been chosen so as to satisfy the energy constraints of Problem **1**. We also let $\bar{\Delta}^{\text{w-peak}}_{\text{un}}(\mathbf{r})$ denote the unnormalized total weighted average peak age in (5.2.11). Finally, we would like to mention that we do not compare the performance of our proposed algorithm with the CSMA algorithms of [76,77] where the goal was solely to minimize the age. Since they do not incorporate energy constraints, it is not fair to compare the performance of our algorithm with them.

Unless stated otherwise, our set up is as follows: The average transmission time is $\mathbb{E}[T] = 5$ ms. The weights w_l's attached to different sources are generated by sampling from a uniform distribution in the interval $[0, 10]$. The target power efficiencies b_l's are randomly generated according to a uniform distribution in the range $[0, 1]$.

5.5.1 Numerical Evaluations

Fig. 5.3 plots the total weighted average peak age $\bar{\Delta}^{\text{w-peak}}_{\text{un}}(\mathbf{r})$ in (5.2.11) as a function of the ratio $\frac{t_s}{\mathbb{E}[T]}$, where the number of sources is $M = 10$. The age-optimal scheduler is seen to outperform the throughput-optimal and Fixed sleep-rate schedulers. This implies that what minimizes the throughput does not necessarily minimize AoI and vice versa. Moreover, we observe that the total weighted average peak age of all schedulers increases as the sensing time increases. This is expected since an increase in the sensing time leads to an increase in the probability of packet collisions, which in turn deteriorates the age performance of these schedulers.

We then scale the number of sources M, and plot $\bar{\Delta}^{\text{w-peak}}_{\text{un}}(\mathbf{r})$ in (5.2.11) as a function of M in Fig. 5.4. While plotting, we normalize the performance by the number of sources M. The sensing time t_s is fixed at $t_s = 40 \sim \mu$s. The weights w_l's corresponding to different sources are randomly generated uniformly within the range $[0, 2]$. The age-optimal scheduler is shown to outperform other schedulers uniformly

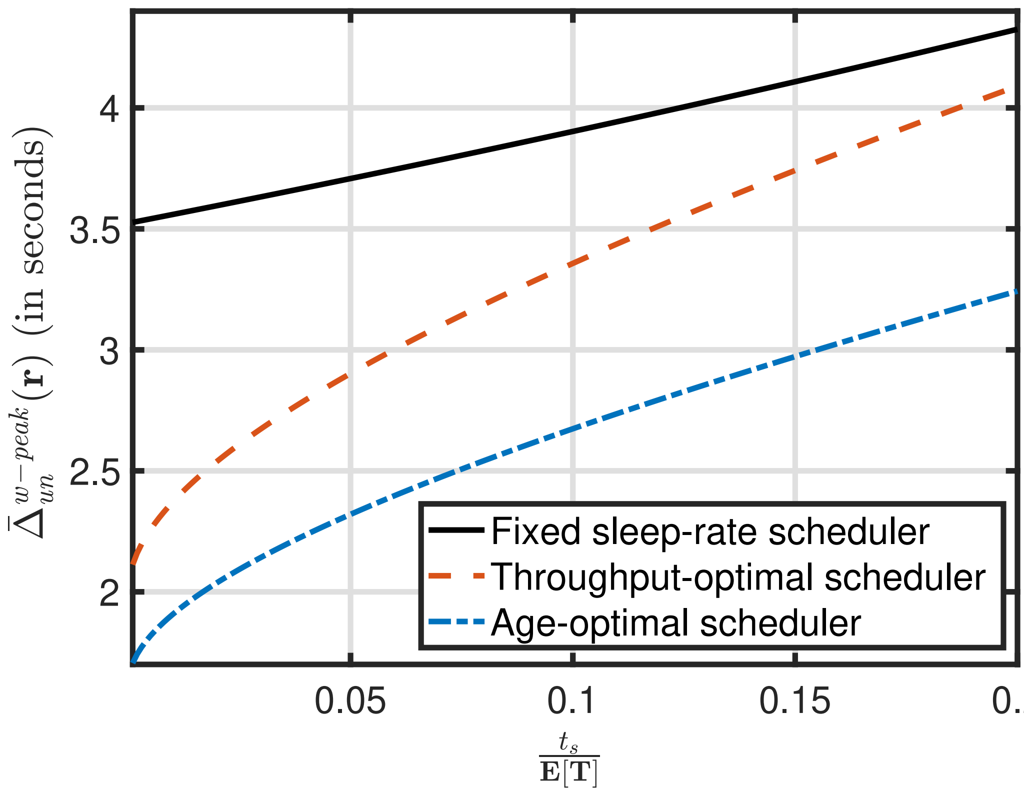


Figure 5.3: Total weighted average peak age $\bar{\Delta}_{\text{un}}^{\text{w-peak}}(\mathbf{r})$ in (5.2.11) versus the ratio $\frac{t_s}{\mathbb{E}[T]}$ for $M = 10$ sources.

for all values of M. Moreover, as we can observe, the average peak age of the sources under age-optimal scheduler increases up to around 0.55 seconds only, while the number of sources rises from 1 to 100. This indicates the robustness of our algorithm to changes in the number of sources in a network.

In Fig. 5.5, we fix the value of M as 100 and the target power efficiencies at the same value for all the sources, i.e., $b_l = b$ for all l. We then vary the parameter b and plot the resulting performance. While plotting, we normalize the performance by the number of sources M. We exclude the simulation of the throughput-optimal scheduler for $b < 0.01$ since the sleeping period parameters that are proposed in [39]

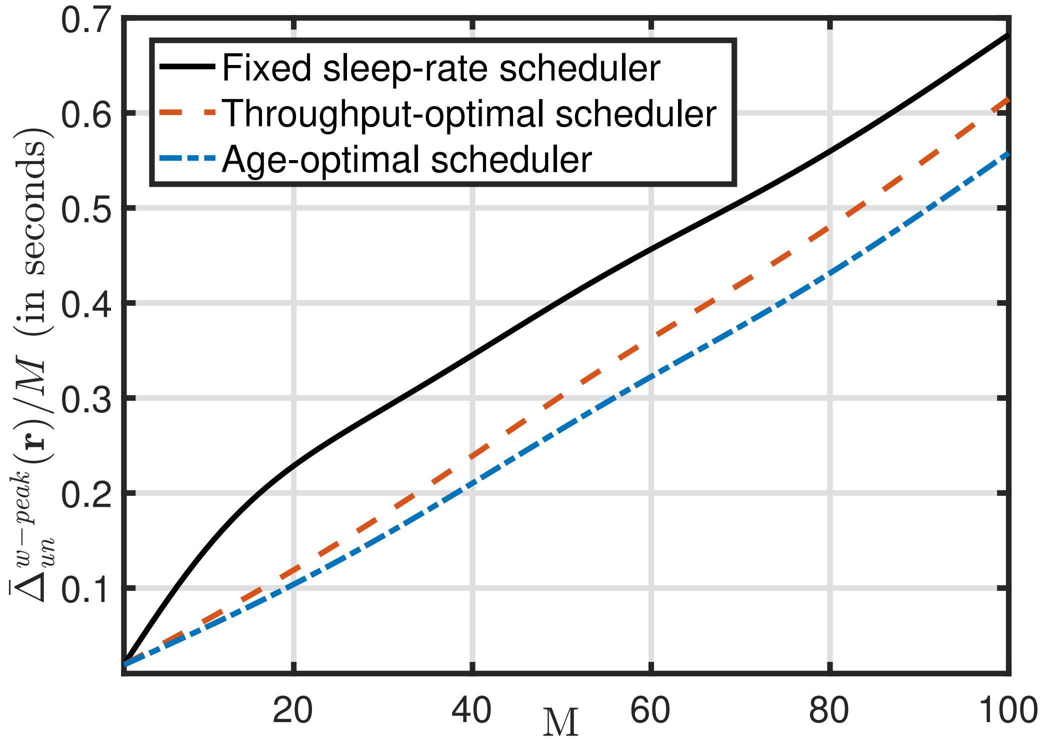


Figure 5.4: Total weighted average peak age $\bar{\Delta}_{\text{un}}^{\text{w-peak}}(\mathbf{r})$ in (5.2.11) versus the number of sources M, where $\bar{\Delta}_{\text{un}}^{\text{w-peak}}(\mathbf{r})$ has been normalized by M while plotting.

are not feasible for Problem **1** in the energy-scarce regime, i.e., when $\sum_{i=1}^{M} b_i < 1$. The age-optimal scheduler outperforms the other schedulers. Moreover, its performance is a decreasing function of b, and then settles at a constant value. This occurs because our proposed solution in (5.3.1) is a function solely of the weights w_l's and $\beta^\star$ when b exceeds some value. Thus, the performance of the proposed scheduler saturates after this value of b.

We now show the effectiveness of the proposed scheduler when deployed in "dense networks" [37, 38]. Dense networks are characterized by a large number of sources connected to a single AP. We fix M at 10^5 sources, and take the target lifetimes of the sources to be equal, i.e., $D_l = D$ for all l. The weights w_l's corresponding to different

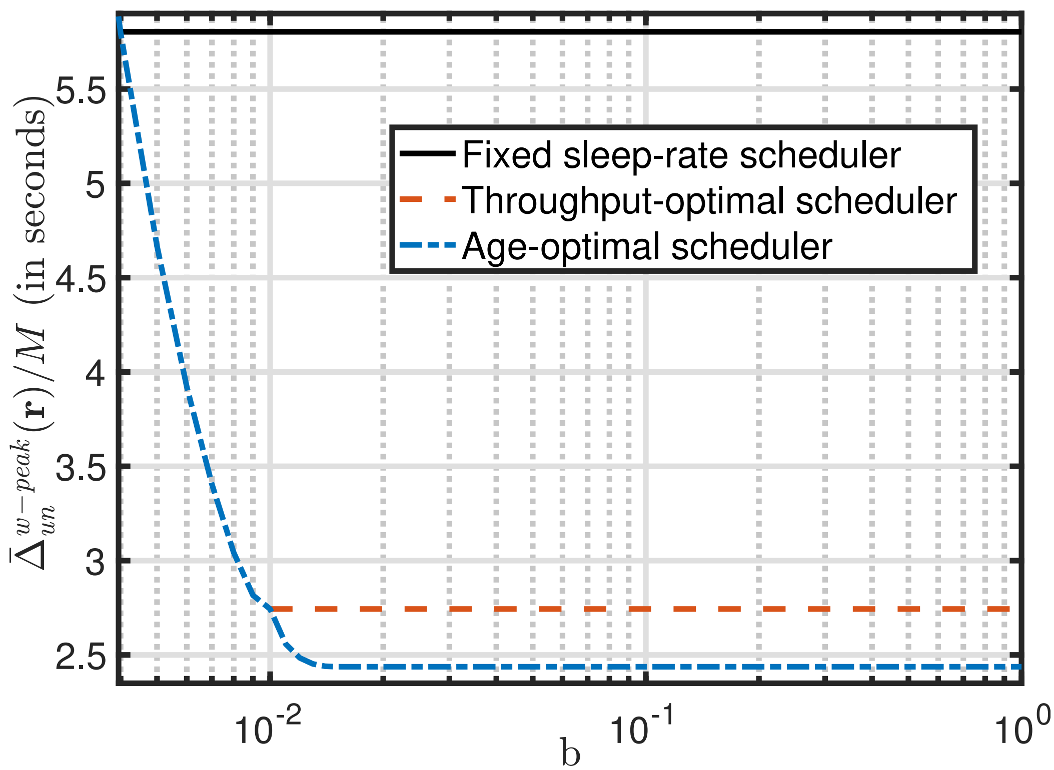


Figure 5.5: Total weighted average peak age $\bar{\Delta}_{\text{un}}^{\text{w-peak}}(\mathbf{r})$ in (5.2.11) versus the target power efficiency b for $M = 100$ sources, where $\bar{\Delta}_{\text{un}}^{\text{w-peak}}(\mathbf{r})$ has been normalized by M while plotting.

sources are generated randomly by sampling from the uniform distribution in the range $[0, 2]$. We let the initial battery level $B_l = 8$ mAh for all l and the output voltage is 5 Volt. We also let the energy consumption in a transmission mode to be 24.75 mW for all sources. We vary the parameter D and plot the resulting performance in Fig. 5.6. While plotting, we normalize the performance by the number of sources M. We exclude simulations for the throughput-optimal scheduler for values of D for which the scheduler is infeasible, i.e., its cumulative energy consumption exceeds the total allowable energy consumption. The age-optimal scheduler is seen to outperform the others. As observed in Fig. 5.6, under the age-optimal scheduler, sources can be

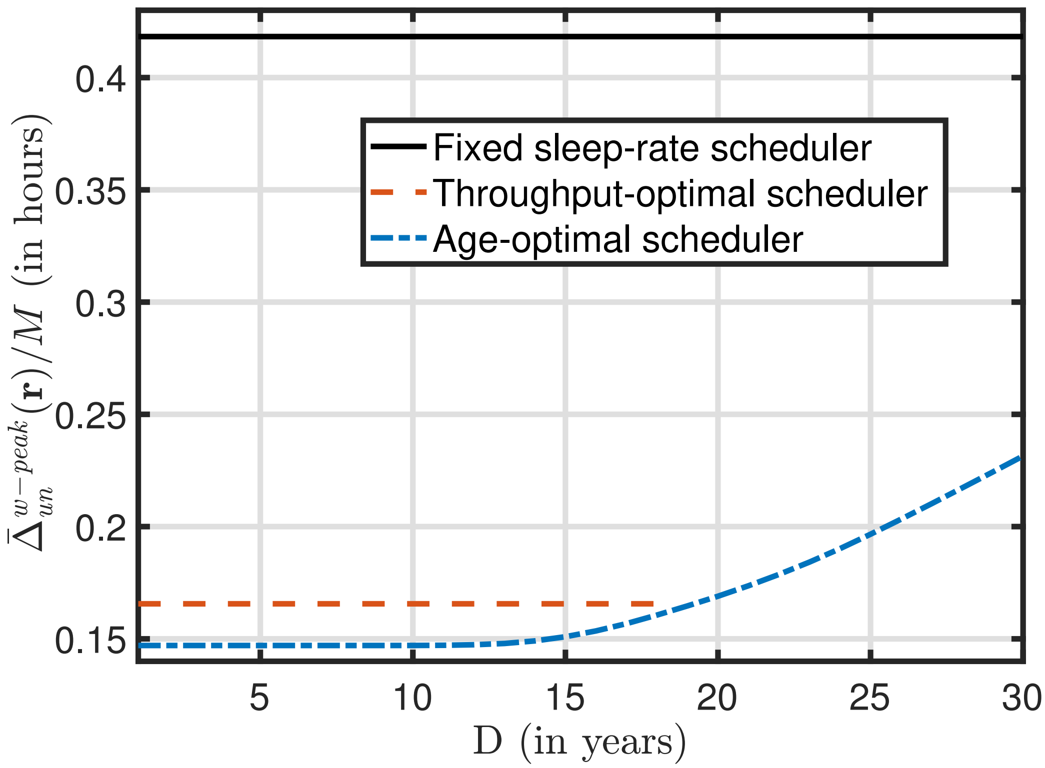


Figure 5.6: Total weighted average peak age $\bar{\Delta}_{\text{un}}^{\text{w-peak}}(\mathbf{r})$ in (5.2.11) versus the target lifetime D for a dense network with $M = 10^5$ sources, where $\bar{\Delta}_{\text{un}}^{\text{w-peak}}(\mathbf{r})$ has been normalized by M while plotting. Since the throughput–optimal scheduler is infeasible for values of D greater than 18 years, we do not plot its performance for these values.

active for up to 25 years, while simultaneously achieving a decent average peak age of around .2 hour, i.e., 12 minutes. This makes it suitable for dense networks, where it is crucial that the sources are necessarily active for many years.

5.5.2 NS-3 Simulation

We use NS-3 [87] to investigate the effect of our model assumptions on the performance of age-optimal scheduler in a more practical situation. We simulate the age-optimal scheduler by using IEEE 802.11b while disabling the RTS-CTS and modifying the back-off times to be exponentially distributed in the MAC layer. Our simulation

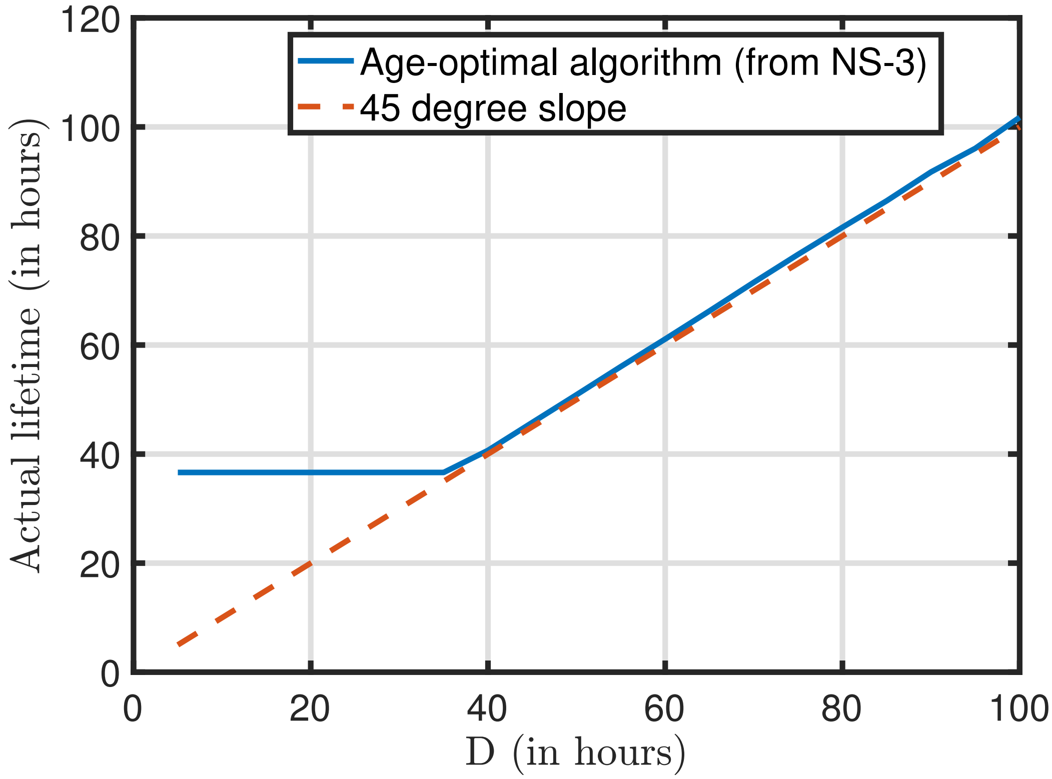


Figure 5.7: The average actual lifetime versus the target lifetime D.

results are averaged over 5 system realizations. The UDP saturation conditions are satisfied such that the source nodes always have packets to send.

Our simulation consists of a WiFi network with 1 AP and 3 associated source nodes in a field of size 50m $\times$ 50m. We set the sensing threshold to -100 dBm which covers a range of 110m. Thus, all sources can hear each other. The initial battery level of each source is 60 mAh, where the output voltage is 5 Volt. For each source, the power consumption in the transmission mode is 24.75 mW, and the power consumption in the sleep mode is 15 μW. Moreover, all weights are set to unity, i.e., $w_l = 1$ for all l.

Fig. 5.7 plots the average actual lifetime of the sources versus the target lifetime, where we take the target lifetimes of all sources to be equal, i.e., $D_l = D$ for all l.

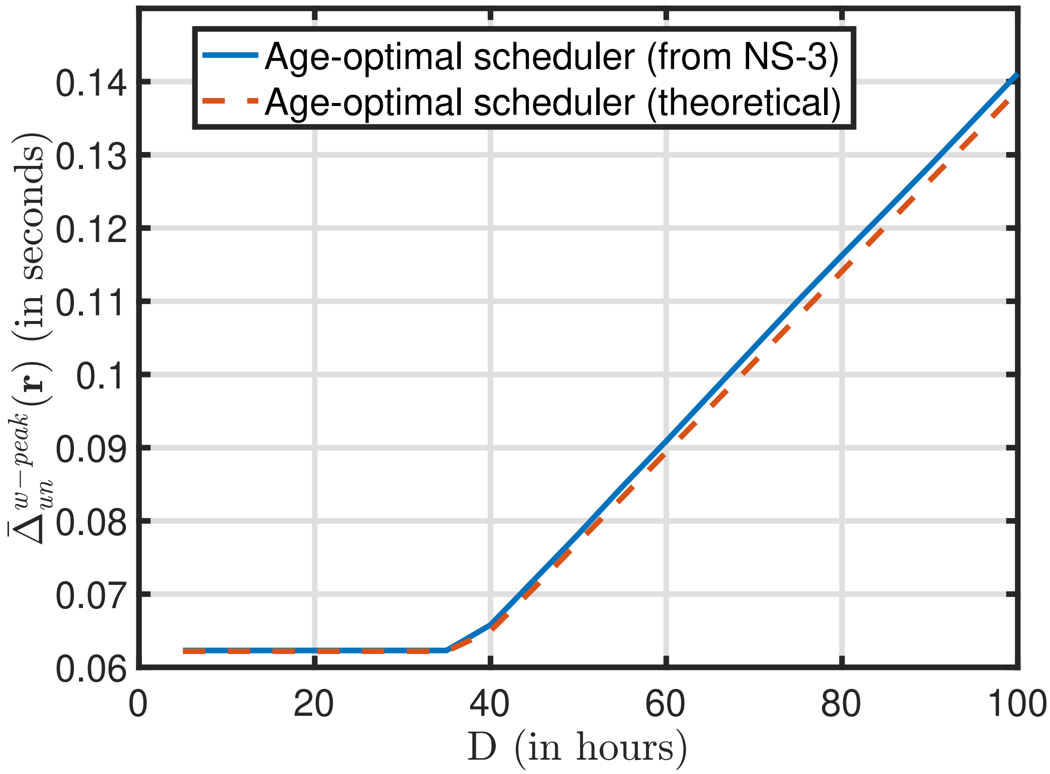


Figure 5.8: Total weighted average peak age $\bar{\Delta}_{\mathrm{un}}^{\text{w-peak}}(\mathbf{r})$ versus the target lifetime D.

As we can observe, the actual lifetime of the age-optimal scheduler always achieves the target lifetime. This suggests that our assumptions (i.e., (i) omitting the power dissipation in the sleep mode and in the sensing times, (ii) the average transmission times and collision times are equal to each other) do not affect the performance of the algorithm which reaches its target lifetime.

Fig. 5.8 plots the total weighted average peak age versus the target lifetime, where again we take the target lifetimes of all sources to be equal, i.e., $D_l = D$ for all l. The age-optimal scheduler (theoretical) curve is obtained using (5.2.11), while the age-optimal scheduler (from NS-3) curve is obtained using the NS-3 simulator. As we can observe, the difference between the plotted curves does not exceed 2% of the age-optimal scheduler (theoretical) performance. This emphasizes the negligible impact of our assumptions on the performance of our proposed algorithm.

5.6 Conclusions

We designed an efficient sleep-wake scheduling algorithm for wireless networks that attains the optimal trade-off between minimizing the AoI and energy consumption. Since the associated optimization problem is non-convex, in general we could not hope to solve it for all values of the system parameters. However, in the regime when the carrier sensing time t_s is negligible as compared to the average transmission time $\mathbb{E}[T]$, we were able to provide a near-optimal solution. Moreover, the proposed solution is in a simple form that allowed us to design an easy-to-implement algorithm to obtain the solution. Finally, we showed that the performance of our proposed algorithm is asymptotically no worse than that of the optimal synchronized scheduler, as $t_s/\mathbb{E}[T] \to 0$.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

In this dissertation, we investigated how we can improve data freshness in information update systems. Our over-arching goal has been to design low-complexity schedulers that can optimize data freshness in a variety of information update systems. To achieve this goal, we have made the following progress.

The first two chapters have focused on networks with stochastic packet arrivals. We began with single-hop single-source information update systems. We showed that LGFS-type of scheduling policies can achieve age optimality when the transmission times are exponentially distributed. We also showed that LGFS-type of policies can be near age-optimal for NBU transmission times. Interestingly, we allowed for packet replication and, unlike throughput and delay, we showed that packet replication can indeed improve the age. We then extended these optimality results to single-source multihop networks.

The second part of this dissertation have focused on multi-source networks in which packet generation is controllable. We started by designing optimal sampler-scheduler pairs that achieve age-optimality in these networks. Despite the complexity that is usually inherited in joint optimization problems, we are able to obtain the optimal sampler-scheduler pair. The key idea was to show the separation principle of the considered problem. In particular, we showed that the scheduler and the sampler can be designed independently. We find that it is optimal to first serve the source with the highest age, and it is not always optimal to generate packets as soon as the channel becomes available. We also derived low-complexity threshold samplers via an approximate analysis of Bellman's equation. We then shifted our focus to design a low-power asynchronous sleep-wake scheduler that can optimize the data freshness in large scale networks. It turned out that the associated optimization problem is non-convex. Nonetheless, we were able to devise a low-complexity solution to solve this problem and prove that, for practical sensing times that are short, the solution is within a small gap from the optimum age performance. Our numerical and NS-3 simulation results show that our solution can indeed elongate the batteries lifetime of information sources, while providing a competitive age performance.

6.0.1 Near-Term Future Research

Based on our preliminary research in Chapter 5, we can extend our study by relaxing its main assumptions. In particular, we can consider the following extensions:

General Age Metrics

So far, we consider the average peak age as a performance metric in our study in Chapter 5. However, the performance metric must capture the variation of the stale information's harmful impact from one application to another. Here comes the importance of using penalty functions of the age. For example, it was recently shown in [19] that, under certain conditions, information freshness metrics expressed in terms of auto-correlation functions, the estimation error of signal values, and mutual information, are monotonic functions of AoI. Moreover, the age-penalty function can be used to represent the level of dissatisfaction of data staleness in different applications based on their demands. For instance, a stair-shape function ($g(x) = \lfloor x \rfloor$) can be used to characterize the dissatisfaction of data staleness when the information of interest is checked periodically, and an exponential function can be utilized in online learning and control applications in which the demand for updating data increases quickly with age. It is known from the literature that what minimizes the average peak age does not necessarily minimize the average penalty function of age and vise versa. Thus, this extension is not straightforward. The difficulty lies in the fact that the average penalty function of age has a more complex relationship with the control action (sleeping period parameters $\mathbf{r}$) than the peak age does. One possible approach to handle this extension is to begin with our peak age near-optimal solution and quantify its performance in terms of the age-penalty function $g(x) = x^k$. Then, through Taylor series expansion, this would quantify the performance of the age metric under consideration.

General Interference Model

In Chapter 5, we assume that all nodes can hear each other. However, in practical applications, we may have general interference model [88] with general interference graph. We assume that the interference model is known in advance. In this model, for each link l, there is a subset of links $N(l)$ that interfere with link l, i.e., $N(l)$ is the set of links that cannot attempt transmission simultaneously with link l, and we may have $N(i) \neq N(j)$. This extension is not straightforward. The difficulty lies in the fact that multiple links can be active simultaneously and not only one. One approach to handle this is to consider a time-slotted system and employ the following modified CSMA (MCSMA) algorithm: A time-slot is divided into a *mini-slot* and a *data-slot*, where the former is dedicated for making channel access decision, and the latter is dedicated for update packet transmission. At the beginning of each mini-slot, each link l generates an exponentially distributed random number with a mean $\frac{1}{r_l}$. Then, in the mini-slot, links take turns to broadcast their generated numbers. The channel access decision at each link is taken as follows. At the end of the mini-slot, each link l gains access to the channel if its generated number is less than the received generated numbers of its neighbors. Since the generated numbers are exponentially distributed, the probability of two numbers to equal each other is zero, and hence the probability of collision is zero. By this, we can handle a general interference model and our target aims to control the parameters r_l's to minimize the average age-penalty function.

6.0.2 Long-Term Open Questions:

Many future applications will extensively rely on sharing real-time information in networks with larger geographic areas. For example, in vehicle-to-vehicle (V2V) networks, vehicles exchange real-time information about the speed and positions of surrounding vehicles to help in avoiding collisions and reducing traffic congestion. Due to the large geographic scale of such networks, some nodes may not be directly connected, and hence can be modeled as multihop networks. Moreover, due to the constant change in the geographic locations of the nodes, the interference model cannot be known ahead of time. This raises the question of how to optimize the data freshness in multihop networks with a time-varying graph? In particular, many challenges will arise in such networks such as hidden terminal problems and relay mechanisms. In the hidden terminal problem, some nodes may not hear the transmission of each other, and hence collision rate increases. Moreover, nodes need to decide when to relay the received messages or send their own packets. This adds another level of difficulty to the design problem and new tools need to be used to solve them. Indeed, a dynamic solution that can adapt to the network's parameters (e.g., mean transmission times, network graph, channel model, etc.) is needed. Learning algorithms can be used to come up with such a solution. In fact, using machine learning to optimize data freshness in such networks is a promising direction in this research area.

BIBLIOGRAPHY

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

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

[3] L. Golab, T. Johnson, and V. Shkapenyuk, "Scheduling updates in a real-time stream warehouse," in *Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE)*, pp. 1207–1210, 2009.

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

[5] R. D. Yates and S. Kaul, "Real-time status updating: Multiple sources," in *Proc. IEEE ISIT*, pp. 2666–2670, 2012.

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

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

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

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

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

[11] 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*, pp. 2589–2597, 2013.

[12] 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.

[13] 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*, pp. 1042–1050, 2014.

[14] 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)*, pp. 585–593, 2015.

[15] Y. Sun, C. E. Koksal, and N. B. Shroff, "On delay-optimal scheduling in queueing systems with replications," *arXiv preprint arXiv:1603.07322*, 2016.

[16] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *Proc. IEEE ISIT*, 2015.

[17] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *Proc. ITA*, Feb. 2015.

[18] 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*, vol. 63, no. 11, pp. 7492–7508, 2017.

[19] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions," *Journal of Communications and Networks*, vol. 21, no. 3, pp. 204–219, 2019.

[20] L. Ran, W. Junfeng, W. Haiying, and L. Gechen, "Design method of can bus network communication structure for electric vehicle," in *International Forum on Strategic Technology 2010*, pp. 326–329, IEEE, 2010.

[21] K. H. Johansson, M. Törngren, and L. Nielsen, "Vehicle applications of controller area network," in *Handbook of networked and embedded control systems*, pp. 741–765, Springer, 2005.

[22] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: an up-to-date survey," *Applied System Innovation*, vol. 3, no. 1, p. 14, 2020.

[23] N. F. Timmons and W. G. Scanlon, "Analysis of the performance of ieee 802.15. 4 for medical sensor body area networking," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks. IEEE SECON 2004.*, pp. 16–24, 2004.

[24] C. A. S. C. P. D. rev 1.0, 17 November 2003.

[25] D. for MC9S08RE8 motorola microcontroller.

[26] R. D. Yates and S. K. Kaul, "Status updates over unreliable multiaccess channels," in *Proc. IEEE ISIT*, pp. 331–335, 2017.

[27] R. Talak, S. Karaman, and E. Modiano, "Distributed scheduling algorithms for optimizing information freshness in wireless networks," in *Proc. IEEE SPAWC*, pp. 1–5, 2018.

[28] R. Li, A. Eryilmaz, and B. Li, "Throughput-optimal wireless scheduling with regulated inter-service times," in *Proc. IEEE INFOCOM*, pp. 2616–2624, 2013.

[29] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2637–2650, 2018.

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

[31] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, "Timely status update in massive iot systems: Decentralized scheduling for wireless uplinks," *arXiv preprint arXiv:1801.03975*, 2018.

[32] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *Proc. INFOCOM*, pp. 1844–1852, 2018.

[33] R. Talak, S. Karaman, and E. Modiano, "Optimizing information freshness in wireless networks under general interference constraints," in *Proc. MobiHoc*, pp. 61–70, 2018.

[34] Q. He, D. Yuan, and A. Ephremides, "Optimal link scheduling for age minimization in wireless systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 7, pp. 5381–5394, 2017.

[35] X. Guo, R. Singh, P. R. Kumar, and Z. Niu, "A risk-sensitive approach for packet inter-delivery time optimization in networked cyber-physical systems," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1976–1989, 2018.

[36] Y. Sun, E. Uysal-Biyikoglu, and S. Kompella, "Age-optimal updates of multiple information flows," in *IEEE INFOCOM - the 1st Workshop on the Age of Information (AoI Workshop)*, pp. 136–141, 2018.

[37] S. S. Kowshik, K. Andreev, A. Frolov, and Y. Polyanskiy, "Energy efficient coded random access for the wireless uplink," *arXiv preprint arXiv:1907.09448*, 2019.

[38] S. S. Kowshik and Y. Polyanskiy, "Fundamental limits of many-user mac with finite payloads and fading," *arXiv preprint arXiv:1901.06732*, 2019.

[39] S. Chen, T. Bansal, Y. Sun, P. Sinha, and N. B. Shroff, "Life-add: Lifetime adjustable design for wifi networks with heterogeneous energy supplies," in *Proc. WiOpt*, pp. 508–515, 2013.

[40]

[41] 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, 2016.

[42] Y. Sang, B. Li, and B. Ji, "The power of waiting for more than one response in minimizing the age-of-information," 2017. https://arxiv.org/abs/1704.04848.

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

[44] 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.

[45] 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.

[46] 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, p. 12, 2017.

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

[48] P. Papadimitratos, A. D. L. Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 84–95, 2009.

[49] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 350–358, 2011.

[50] S. Kaul, R. Yates, and M. Gruteser, "On piggybacking in vehicular networks," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pp. 1–5, 2011.

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

[52] J. Selen, Y. Nazarathy, L. L. Andrew, and H. L. Vu, "The age of information in gossip networks," in *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, pp. 364–379, Springer, 2013.

[53]

[54] T. Shreedhar, S. K. Kaul, and R. D. Yates, "Acp: An end-to-end transport protocol for delivering fresh updates in the internet-of-things," *arXiv preprint arXiv:1811.03353*, 2018.

[55] R. D. Yates, "Age of information in a network of preemptive servers," *arXiv preprint arXiv:1803.07993*, 2018.

[56] R. D. Yates, "The age of information in networks: Moments, distributions, and sampling," *arXiv preprint arXiv:1806.03487*, 2018.

[57] C. Joo and A. Eryilmaz, "Wireless scheduling for information freshness and synchrony: Drift-based design and heavy-traffic analysis," in *Proc. WiOpt*, pp. 1–8, 2017.

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

[59] S. Farazi, A. G. Klein, J. A. McNeill, and D. R. Brown, "On the age of information in multi-source multi-hop wireless status update networks," in *Proc. IEEE SPAWC.*, pp. 1–5, 2018.

[60] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, "The stationary distribution of the age of information in fcfs single-server queues," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 571–575, 2017.

[61] A. M. Bedewy, Y. Sun, R. Singh, and N. B. Shroff, "Optimizing information freshness using low-power status updates via sleep-wake scheduling," in *Proc. MobiHoc*, pp. 51–60, 2020.

[62] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Minimizing the age of information in broadcast wireless networks," in *Proc. Allerton*, pp. 844–851, 2016.

[63] Y. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *arXiv preprint arXiv:1712.07419*, 2017.

[64]

[65] R. Talak, S. Karaman, and E. Modiano, "Optimizing age of information in wireless networks with perfect channel state information," in *Proc. WiOpt*, pp. 1–8, 2018.

[66] R. Talak, I. Kadota, S. Karaman, and E. Modiano, "Scheduling policies for age minimization in wireless networks with unknown channel state," in *Proc. IEEE ISIT*, pp. 2564–2568, 2018.

[67] 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.

[68] D. P. Bertsekas, *Dynamic Programming and Optimal Control, 2nd ed*, vol. 2. Belmont, MA: Athena Scientific, 2001.

[69] M. L. Puterman, "Markov decision processes: Discrete stochastic dynamic programming (wiley series in probability and statistics)," 2005.

[70] L. P. Kaelbling and L. P. Kaelbling, *Recent advances in reinforcement learning*. Springer, 1996.

[71] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*, vol. 703. John Wiley & Sons, 2007.

[72] I. Bettesh and S. Shamai, "Optimal power and rate control for minimal average delay: The single-user case," vol. 52, no. 9, pp. 4115–4141, *IEEE Trans. Inf. Theory*, 2006.

[73] R. Wang and V. K. Lau, "Delay-optimal two-hop cooperative relay communications via approximate mdp and distributive stochastic learning," IEEE Trans. Inf. Theory, 2013.

[74] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Golden section search in one dimension," *Numerical Recipes in C: The Art of Scientific Computing*, p. 2, 1992.

[75] S. Yun, Y. Yi, J. Shin, *et al.*, "Optimal CSMA: a survey," in *Proc. ICCS*, pp. 199–204, 2012.

[76] A. Maatouk, M. Assaad, and A. Ephremides, "Minimizing the age of information in a CSMA environment," *arXiv preprint arXiv:1901.00481*, 2019.

[77] M. Wang and Y. Dong, "Broadcast age of information in CSMA/CA based wireless networks," *arXiv preprint arXiv:1904.03477*, 2019.

[78] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, 2010.

[79] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.

[80] A. N. Shiryaev, *Optimal stopping rules.* New York: Springer-Verlag, 1978.

[81] A. Wald, *Sequential analysis.* New York: Courier Corporation, 1973.

[82] A. transceiver TR1000 data sheet, RF Monolithic Inc.

[83] K. F. Ramadan, M. I. Dessouky, M. Abd-Elnaby, and F. E. A. El-Samie, "Energy-efficient dual-layer mac protocol with adaptive layer duration for wsns," in *11th International Conference on Computer Engineering Systems (ICCES)*, pp. 47–52, 2016.

[84] A. El-Hoiydi, "Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks," in *Proc. IEEE Int. Symp. Comput. Commun. (ISCC)*, pp. 685–692, 2002.

[85] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen, "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1013–1024, 2016.

[86] P. Hsieh and I. Hou, "A decentralized medium access protocol for real-time wireless ad hoc networks with unreliable transmissions," in *IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 972–982, 2018.

[87] NS-3. https://www.nsnam.org/.".

[88] A. Kumar, D. Manjunath, and J. Kuri, *Wireless networking.* Elsevier, 2008.

[89] Y. Sun, C. E. K., and N. B. Shroff, "Near delay-optimal scheduling of batch jobs in multi-server systems," *Ohio State Univ., Tech. Rep*, 2017.

[90] L. I. Sennott, "Average cost optimal stationary policies in infinite state markov decision processes with unbounded costs," *Operations Research*, vol. 37, no. 4, pp. 626–633, 1989.

[91] D. P. Bertsekas, *Nonlinear Programming.* Belmont, MA: Athena Scientific, 2 ed., 1999.

[92] D. Butnariu and A. N. Iusem, *Totally Convex Functions for Fixed Points Computation and Infinite Dimensional Optimization.* Norwell, MA, USA: Kluwer Academic Publisher, 2000.

[93] R. Durrett, *Probability: theory and examples.* Cambridge university press, 2010.

[94] R. G. Gallager, *Discrete stochastic processes.* Boston: Kluwer Academic Publishers, 1996.

[95] S. Boyd and L. Vandenberghe, *Convex optimization*. New York, NY, USA: Cambridge University Press, 2004.

APPENDIX A: PROOFS FOR CHAPTER 2

A.1 Proof of Theorem 2.3.1

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

Definition A.1.1. *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,*

$$U_\pi(t) \leq \alpha_{[m],\pi}(t) \leq \ldots \leq \alpha_{[1],\pi}(t). \tag{A.1.1}$$

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 2.3.1 is the following lemma, where we compare policy P with any work-conserving policy π.

Lemma A.1.1. *Suppose that* $\mathbf{V}_P(0^-) = \mathbf{V}_\pi(0^-)$ *for all work conserving policies* π*, then for all* $\mathcal{I}$

$$[\{\mathbf{V}_P(t), t \in [0,\infty)\}|\mathcal{I}] \geq_{st} [\{\mathbf{V}_\pi(t), t \in [0,\infty)\}|\mathcal{I}]. \tag{A.1.2}$$

We use coupling and forward induction to prove Lemma A.1.1. 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 2.2.5, if we can show

$$\mathbb{P}[\widetilde{\mathbf{V}}_P(t) \geq \widetilde{\mathbf{V}}_\pi(t), t \in [0,\infty)|\mathcal{I}] = 1, \tag{A.1.3}$$

then (A.1.2) 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 (A.1.3):

Lemma A.1.2. *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*

$$U_P \geq U_\pi, \tag{A.1.4}$$

then,

$$\alpha_{[i],P} \geq \alpha_{[i],\pi}, \quad \forall i = 1, \ldots, m. \tag{A.1.5}$$

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 A.1.2 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 (A.1.1), and the definition of policy P, we have

$$\begin{aligned} &\alpha_{[i],P} = \max\{s_{[j]}, U_P\}, \ \forall i = (j-1)r+1, \ldots, jr, \ \forall j = 1, \ldots, k, \\ &\alpha_{[i],P} = \max\{s_{[k+1]}, U_P\}, \ \forall i = kr+1, \ldots, m. \end{aligned} \tag{A.1.6}$$

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

$$\begin{aligned} &\alpha_{[i],\pi} \leq \max\{s_{[j]}, U_\pi\}, \ \forall i = (j-1)r+1, \ldots, jr, \ \forall j = 1, \ldots, k, \\ &\alpha_{[i],\pi} \leq \max\{s_{[k+1]}, U_\pi\}, \ \forall i = kr+1, \ldots, m. \end{aligned} \tag{A.1.7}$$

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 (A.1.6) and (A.1.7) imply

$$\alpha_{[i],P} \geq \alpha_{[i],\pi}, \quad \forall i = 1, \ldots, m, \tag{A.1.8}$$

which completes the proof. $\square$

Lemma A.1.3. *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*

$$\alpha_{[i],P} \geq \alpha_{[i],\pi}, \quad \forall i = 1, \ldots, m, \tag{A.1.9}$$

then,

$$U'_P \geq U'_\pi, \alpha'_{[i],P} \geq \alpha'_{[i],\pi}, \quad \forall i = 1, \ldots, m. \tag{A.1.10}$$

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

$$U'_P = \alpha_{[l],P} \geq \alpha_{[l],\pi} = U'_\pi. \tag{A.1.11}$$

This, together with Lemma A.1.2, implies

$$\alpha'_{[i],P} \geq \alpha'_{[i],\pi}, \quad i = 1, \ldots, m. \tag{A.1.12}$$

Hence, (A.1.10) holds for any queue size $B \geq 0$, which completes the proof. $\square$

Lemma A.1.4. *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*

$$U_P \geq U_\pi, \tag{A.1.13}$$

then

$$U'_P \geq U'_\pi, \alpha'_{[i],P} \geq \alpha'_{[i],\pi}, \quad \forall i = 1, \ldots, m. \tag{A.1.14}$$

Proof. Since there is no packet delivery, we have

$$U'_P = U_P \geq U_\pi = U'_\pi. \tag{A.1.15}$$

This, together with Lemma A.1.2, implies

$$\alpha'_{[i],P} \geq \alpha'_{[i],\pi}, \quad i = 1, \ldots, m. \tag{A.1.16}$$

Hence, (A.1.14) holds for any queue size $B \geq 0$, which completes the proof. □

Proof of Lemma A.1.1. 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 A.1.3 and A.1.4, 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, (A.1.3) follows which implies (A.1.2) by Proposition 2.2.5. This completes the proof. □

Proof of Theorem 2.3.1. As a result of Lemma A.1.1, 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

$$[\{\Delta_P(t), t \in [0, \infty)\}|\mathcal{I}] \leq_{\text{st}} [\{\Delta_\pi(t), t \in [0, \infty)\}|\mathcal{I}], \tag{A.1.17}$$

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, (2.3.2) follows directly from (2.3.1) using the properties of stochastic ordering [47]. This completes the proof. □

A.2 Proof of Lemma 2.3.1

The proof of Lemma 2.3.1 is motivated by the proof idea of [15, 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

$$\Gamma_i = \min\{v_j : s_j \geq s_i\}, \tag{A.2.1}$$

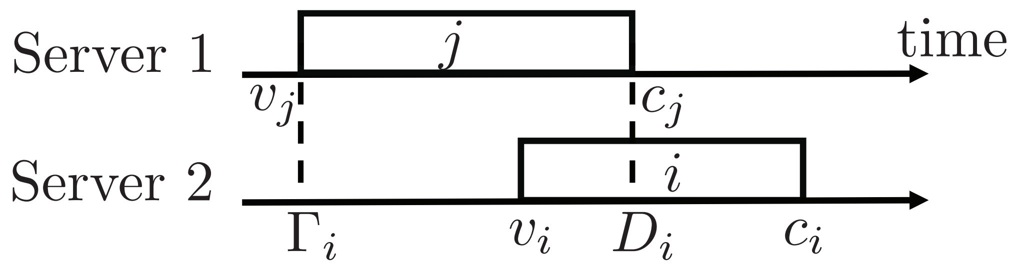


Figure A.1: An illustration of v_i, c_i, Γ_i, and D_i. There are 2 servers, and $s_j > s_i$. There is no packet with generation time greater than s_i that is assigned to any of the servers before time v_j. Packet j is assigned to Server 1 at time v_j and delivered to the destination at time c_j; while 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 $D_i = c_j$.

$$D_i = \min\{c_j : s_j \geq s_i\}. \tag{A.2.2}$$

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. A.1. Suppose that there are n update packets, where n is an arbitrary positive integer, no matter finite or infinite. Define the vectors $\mathbf{\Gamma} = (\Gamma_1, \ldots, \Gamma_n)$, and $\mathbf{D} = (D_1, \ldots, D_n)$. All these quantities are functions of the scheduling policy π.

Notice that we can deduce from (2.2.2) 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 (2.3.4) that the process $\{\Delta_P^{\text{LB}}(t), t \in [0, \infty)\}$ is an increasing function of $\mathbf{\Gamma}(P)$. According to Proposition 2.2.2, if we can show

$$[\mathbf{\Gamma}(P)|\mathcal{I}] \leq_{\text{st}} [\mathbf{D}(\pi)|\mathcal{I}], \tag{A.2.3}$$

holds for all $\pi \in \Pi_m$, then (2.3.5) is proven. Hence, (A.2.3) is what we need to show. We pick an arbitrary policy $\pi \in \Pi_m$ and prove (A.2.3) using Proposition 2.2.1 into two steps.

Step 1: Consider packet 1. Define $i^* = \arg\min_i a_i$, 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 (A.2.1), we obtain

$$[\Gamma_1(P)|\mathcal{I}] = [v_{i^*}(P)|\mathcal{I}] = a_{i^*}. \tag{A.2.4}$$

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

$$[c_i(\pi)|\mathcal{I}] \geq a_{i^*},\ \forall i \geq 1. \tag{A.2.5}$$

This with (A.2.2) imply

$$[D_1(\pi)|\mathcal{I}] \geq a_{i^*}. \tag{A.2.6}$$

Combining (A.2.4) and (A.2.6), we get

$$[\Gamma_1(P)|\mathcal{I}] \leq [D_1(\pi)|\mathcal{I}]. \tag{A.2.7}$$

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

$$\begin{aligned} &[\Gamma_j(P)|\mathcal{I}, \Gamma_1(P) = \gamma_1, \ldots, \Gamma_{j-1}(P) = \gamma_{j-1}] \\ \leq_{\text{st}} &[D_j(\pi)|\mathcal{I}, D_1(\pi) = d_1, \ldots, D_{j-1}(\pi) = d_{j-1}] \\ &\text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1. \end{aligned} \tag{A.2.8}$$

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 (A.2.8) holds in either case.

As illustrated in Fig. A.2, 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 $v_j(\pi) = \min_{w=1,\ldots,u} \tau_w$.[1] 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 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 $X_{l_w}^{\pi} = \alpha_w - \tau_w$ denote the service time of one copy of packet j in server l_w under

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

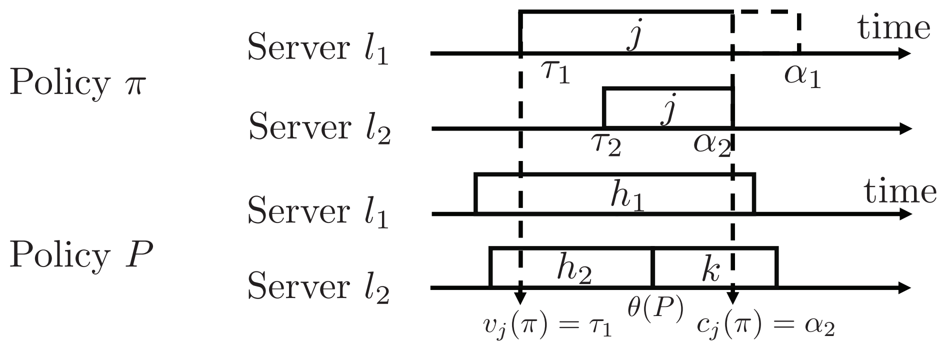


Figure A.2: Illustration of packet assignments under policy π and policy P. In policy π, two copies of packet j are replicated on the server l_1 and server l_2 at time τ_1 and τ_2, where $v_j(\pi) = \min\{\tau_1, \tau_2\} = \tau_1$. Server l_2 completes one copy of packet j at time $c_j(\pi) = \alpha_2$, 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 $\theta(P)$.

policy π and $X_{l_w}^P = \chi_{l_w} + R_{l_w}$ denote the service time of packet h_w in server l_w under policy P. The CCDF of R_{l_w} is given by

$$\mathbb{P}[R_{l_w} > s] = \mathbb{P}[X_{l_w}^P - \chi_{l_w} > s | X_{l_w}^P > \chi_{l_w}]. \tag{A.2.9}$$

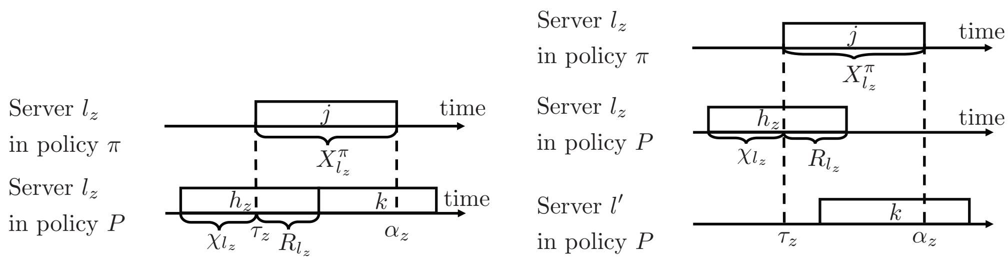
Because the packet service times are NBU, we can obtain that for all $s, \chi_{l_w} \geq 0$

$$\mathbb{P}[X_{l_w}^P - \chi_{l_w} > s | X_{l_w}^P > \chi_{l_w}] = \mathbb{P}[X_{l_w}^\pi - \chi_{l_w} > s | X_{l_w}^\pi > \chi_{l_w}] \leq \mathbb{P}[X_{l_w}^\pi > s]. \tag{A.2.10}$$

By combining (A.2.9) and (A.2.10), we obtain

$$R_{l_w} \leq_{\text{st}} X_{l_w}^\pi. \tag{A.2.11}$$

Because the packet service times are independent across the servers, by Lemma 13



(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 A.3: The possible cases to occur after the completion of packet h_z.

of [15], $R_{l_1}, \ldots, R_{l_u}$ are mutually independent. By Proposition 2.2.3 and (A.2.11), we can obtain

$$\min_{w=1,\ldots,u} \tau_w + R_{l_w} \leq_{\text{st}} \min_{w=1,\ldots,u} \tau_w + X^{\pi}_{l_w} = \min_{w=1,\ldots,u} \alpha_w. \tag{A.2.12}$$

From (A.2.12) 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 $c_j(\pi) = \min\limits_{w=1,\ldots,u} \alpha_w$. Let $\theta(P)$ denote the time that server l_z becomes available to serve a new packet in policy P. According to (A.2.12), we have

$$\begin{aligned} &[\theta(P)|\mathcal{I}, \Gamma^{j-1}] \leq_{\text{st}} [c_j(\pi)|\mathcal{I}, D^{j-1}] \\ &\text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1. \end{aligned} \tag{A.2.13}$$

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. A.3(a). Hence, we obtain

$$\begin{aligned} [v_k(P)|\mathcal{I}, \Gamma^{j-1}] = [\theta(P)|\mathcal{I}, \Gamma^{j-1}] \leq_{\text{st}} [c_j(\pi)|\mathcal{I}, D^{j-1}] \\ \text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1. \end{aligned} \tag{A.2.14}$$

Since $s_k \geq s_j$, (A.2.1) implies

$$[\Gamma_j(P)|\mathcal{I}, \Gamma^{j-1}] \leq [v_k(P)|\mathcal{I}, \Gamma^{j-1}] \tag{A.2.15}$$

Since there is no packet with generation time greater than s_j that has been delivered before packet j under policy π, (A.2.2) implies

$$[D_j(\pi)|\mathcal{I}, D^{j-1}] = [c_j(\pi)|\mathcal{I}, D^{j-1}] \tag{A.2.16}$$

By combining (A.2.14), (A.2.15), and (A.2.16), (A.2.8) 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. A.3(b). Hence, we obtain

$$\begin{aligned} [v_k(P)|\mathcal{I}, \Gamma^{j-1}] \leq [\theta(P)|\mathcal{I}, \Gamma^{j-1}] \leq_{\text{st}} [c_j(\pi)|\mathcal{I}, D^{j-1}] \\ \text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1. \end{aligned} \tag{A.2.17}$$

Similar to Case 1, we can use (A.2.1), (A.2.2), and (A.2.17) to show that (A.2.8) 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 (A.2.8) to (A.2.17) to obtain

$$\begin{aligned}&[\Gamma_y(P)|\mathcal{I}, \Gamma^{j-1}] \leq_{\text{st}} [D_y(\pi)|\mathcal{I}, D^{j-1}] \\ &\text{whenever} \quad \gamma_i \leq d_i, i = 1, 2, \ldots, j-1.\end{aligned} \tag{A.2.18}$$

Observing that $s_y > s_j$, (A.2.1) implies

$$[\Gamma_j(P)|\mathcal{I}, \Gamma^{j-1}] \leq [\Gamma_y(P)|\mathcal{I}, \Gamma^{j-1}]. \tag{A.2.19}$$

Since $c_y(\pi) < c_j(\pi)$ and $s_y > s_j$, (A.2.2) implies

$$[D_j(\pi)|\mathcal{I}, D^{j-1}] = [D_y(\pi)|\mathcal{I}, D^{j-1}]. \tag{A.2.20}$$

By combining (A.2.18), (A.2.19), and (A.2.20), we can prove (A.2.8) in this case too. Now, substituting (A.2.7) and (A.2.8) into Proposition 2.2.1, (A.2.3) is proven. This completes the proof. □

A.3 Proof of Theorem 2.3.2

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

Proof of Theorem 2.3.2.(a). We prove Theorem 2.3.2.(a) into two steps:

Step 1: We will show that the average gap between Δ_P^{LB} and Δ_P is upper bounded by $\mathbb{E}[X]$. Recall the definitions of Γ_i and D_i from (A.2.1) and (A.2.2), 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 (A.2.2), 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

$$\mathbb{E}[d_i] \leq \mathbb{E}[X], \ \forall i. \tag{A.3.1}$$

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

$$[\bar{G}|\mathcal{I}] = \limsup_{T\rightarrow\infty} \frac{\mathbb{E}[\int_0^T G(t)dt]}{T}. \tag{A.3.2}$$

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

$$\sum_{i=1}^{N(T)} \tau_i \leq T. \tag{A.3.3}$$

The area defined by the integral in (A.3.2) can be decomposed into a sum of disjoint geometric parts. Observing Fig. A.4, the area can be approximated to the concatenation of the parallelograms $G_1, G_2, \ldots$ (G_i's are highlighted in Fig. A.4). 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

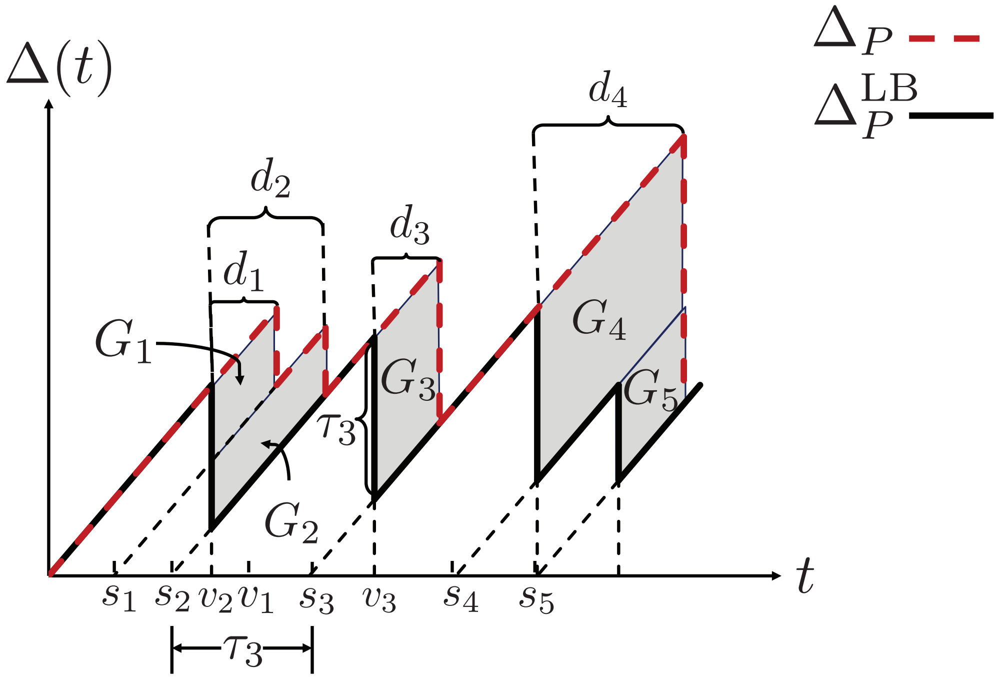


Figure A.4: The evolution of Δ_P^{LB} and Δ_P in a queue with 4 servers and $r = 2$.

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

$$\sum_{i=1}^{N(T)} G_i \geq \int_0^T G(t)dt. \tag{A.3.4}$$

Combining (A.3.3) and (A.3.4), we get

$$\frac{\int_0^T G(t)dt}{T} \leq \frac{\sum_{i=1}^{N(T)} G_i}{\sum_{i=1}^{N(T)} \tau_i}. \tag{A.3.5}$$

Then, take conditional expectation given τ and $N(T)$ on both sides of (A.3.5), we obtain

$$\frac{\mathbb{E}[\int_0^T G(t)dt|\tau, N(T)]}{T} \leq \frac{\mathbb{E}[\sum_{i=1}^{N(T)} G_i|\tau, N(T)]}{\sum_{i=1}^{N(T)} \tau_i} = \frac{\sum_{i=1}^{N(T)} \mathbb{E}[G_i|\tau, N(T)]}{\sum_{i=1}^{N(T)} \tau_i}, \quad \text{(A.3.6)}$$

where the second equality follows from the linearity of the expectation. From Fig. A.4, G_i can be calculated as

$$G_i = \tau_i d_i. \quad \text{(A.3.7)}$$

Substituting by (A.3.7) into (A.3.6), yields

$$\frac{\mathbb{E}[\int_0^T G(t)dt|\tau, N(T)]}{T} \leq \frac{\sum_{i=1}^{N(T)} \mathbb{E}[\tau_i d_i|\tau, N(T)]}{\sum_{i=1}^{N(T)} \tau_i} = \frac{\sum_{i=1}^{N(T)} \tau_i \mathbb{E}[d_i|\tau, N(T)]}{\sum_{i=1}^{N(T)} \tau_i}. \quad \text{(A.3.8)}$$

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 (A.3.8), yields

$$\frac{\mathbb{E}[\int_0^T G(t)dt|\tau, N(T)]}{T} \leq \frac{\sum_{i=1}^{N(T)} \tau_i \mathbb{E}[X]}{\sum_{i=1}^{N(T)} \tau_i} = \mathbb{E}[X], \quad \text{(A.3.9)}$$

by the law of iterated expectations, we have

$$\frac{\mathbb{E}[\int_0^T G(t)dt]}{T} \leq \mathbb{E}[X]. \quad \text{(A.3.10)}$$

Taking the $\limsup$ of both side of (A.3.10) when $T \to \infty$, yields

$$\limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T G(t)dt]}{T} \leq \mathbb{E}[X]. \quad \text{(A.3.11)}$$

Equation (A.3.11) tells us that the average gap between Δ_P^{LB} and Δ_P is no larger than $\mathbb{E}[X]$.

Step 2: We prove (2.3.6). 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

$$[\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}] \leq [\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] + \mathbb{E}[X], \tag{A.3.12}$$

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 2.3.1, we have for all $\mathcal{I}$ satisfying $B \geq 1$, and $\pi \in \Pi_m$

$$[\{\Delta_P^{\text{LB}}(t), t \in [0,\infty)\}|\mathcal{I}] \leq_{\text{st}} [\{\Delta_\pi(t), t \in [0,\infty)\}|\mathcal{I}], \tag{A.3.13}$$

which implies that

$$[\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] \leq [\bar{\Delta}_\pi|\mathcal{I}], \tag{A.3.14}$$

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

$$[\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] \leq \min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}]. \tag{A.3.15}$$

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

$$\min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}]. \tag{A.3.16}$$

Combining (A.3.12), (A.3.15), and (A.3.16), we get

$$\min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}] \leq \min_{\pi\in\Pi_m} [\bar{\Delta}_\pi|\mathcal{I}] + \mathbb{E}[X], \tag{A.3.17}$$

which completes the proof. □

Proof of Theorem 2.3.2.(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 (A.2.2), 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

$$\mathbb{E}[d_i] \leq \mathbb{E}[\min_{l=1,\ldots,r} X_l], \forall i. \tag{A.3.18}$$

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

$$[\bar{G}|\mathcal{I}] = \limsup_{T \to \infty} \frac{\mathbb{E}[\int_0^T G(t)dt]}{T}. \tag{A.3.19}$$

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

$$\limsup_{T \to \infty} \frac{\mathbb{E}[\int_0^T G(t)dt]}{T} \leq \mathbb{E}[\min_{l=1,\ldots,r} X_l]. \tag{A.3.20}$$

Equation (A.3.20) 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

$$[\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}] \leq [\bar{\Delta}_P^{\text{LB}}|\mathcal{I}] + \mathbb{E}[\min_{l=1,\ldots,r} X_l]. \tag{A.3.21}$$

Similar to part (a), we can use (A.3.21) with Lemma 2.3.1 to show that

$$\min_{\pi\in\Pi_m}[\bar{\Delta}_\pi|\mathcal{I}] \leq [\bar{\Delta}_P|\mathcal{I}] \leq \min_{\pi\in\Pi_m}[\bar{\Delta}_\pi|\mathcal{I}] + \mathbb{E}[\min_{l=1,\ldots,r} X_l], \tag{A.3.22}$$

which completes the proof. □

A.4 Proof of Theorem 2.4.1

We follow the same proof technique of Theorem 2.3.1. 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 A.4.1. *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 2.4.1, we will need the following lemma.

Lemma A.4.1. *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 A.4.1:

Lemma A.4.2. *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

$$N'_P = N'_\pi, \gamma'_P = \gamma'_\pi. \tag{A.4.1}$$

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, (A.4.1) holds, which complete the proof. □

Lemma A.4.3. *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

$$N'_P = N'_\pi, \gamma'_P = \gamma'_\pi. \tag{A.4.2}$$

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, (A.4.2) holds, which complete the proof. □

Proof of Lemma A.4.1. 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 A.4.2 and A.4.3, 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 2.4.1. As a result of Lemma A.4.1, $[\{\gamma_P(t), t \in [0,\infty)\}|\mathcal{I}]$ 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 A.4.1, 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). □

APPENDIX B: PROOFS FOR CHAPTER 3

B.1 Proof of Theorem 3.3.1

Let us define the system state of a policy π:

Definition B.1.1. *At any time t, the* system state *of policy π is specified by $\mathbf{U}_\pi(t) = (U_{0,\pi}(t), U_{2,\pi}(t), \ldots, U_{N-1,\pi}(t))$, where $U_{j,\pi}(t)$ is the generation time of the freshest packet that arrived at node j by time t. Let $\{\mathbf{U}_\pi(t), t \in [0,\infty)\}$ be the state process of policy π, which is assumed to be right-continuous. For notational simplicity, let policy P represent the preemptive LGFS policy.*

The key step in the proof of Theorem 3.3.1 is the following lemma, where we compare policy P with any work-conserving policy π.

Lemma B.1.1. *Suppose that $\mathbf{U}_P(0^-) = \mathbf{U}_\pi(0^-)$ for all work conserving policies π, then for all $\mathcal{I}$,*

$$[\{\mathbf{U}_P(t), t \in [0,\infty)\}|\mathcal{I}] \geq_{st} [\{\mathbf{U}_\pi(t), t \in [0,\infty)\}|\mathcal{I}]. \tag{B.1.1}$$

We use coupling and forward induction to prove Lemma B.1.1. For any work-conserving policy π, suppose that stochastic processes $\widetilde{\mathbf{U}}_P(t)$ and $\widetilde{\mathbf{U}}_\pi(t)$ have the same distributions with $\mathbf{U}_P(t)$ and $\mathbf{U}_\pi(t)$, respectively. The state processes $\widetilde{\mathbf{U}}_P(t)$ and $\widetilde{\mathbf{U}}_\pi(t)$ are coupled in the following manner: If a packet is delivered from node i to node j at time t as $\widetilde{\mathbf{U}}_P(t)$ evolves in policy P, then there exists a packet delivery from node i

to node j at time t as $\widetilde{\mathbf{U}}_\pi(t)$ evolves in policy π. Such a coupling is valid since the transmission time is exponentially distributed and thus memoryless. Moreover, policy P and policy π have identical packet generation times $(s_1, s_2, \ldots, s_n)$ at the external source and packet arrival times $(a_{10}, a_{20}, \ldots, a_{n0})$ to node 0. According to Theorem 6.B.30 in [47], if we can show

$$\mathbb{P}[\widetilde{\mathbf{U}}_P(t) \geq \widetilde{\mathbf{U}}_\pi(t), t \in [0, \infty)|\mathcal{I}] = 1, \tag{B.1.2}$$

then (B.1.1) *is proven.*

To ease the notational burden, we will omit the tildes in this proof on the coupled versions and just use $\mathbf{U}_P(t)$ and $\mathbf{U}_\pi(t)$. Next, we use the following lemmas to prove (B.1.2)*:*

Lemma B.1.2. *Suppose that under policy P, $\mathbf{U}'_P$ is obtained by a packet delivery over the link (i, j) in the system whose state is $\mathbf{U}_P$. Further, suppose that under policy π, $\mathbf{U}'_\pi$ is obtained by a packet delivery over the link (i, j) in the system whose state is $\mathbf{U}_\pi$. If*

$$\mathbf{U}_P \geq \mathbf{U}_\pi, \tag{B.1.3}$$

then,

$$\mathbf{U}'_P \geq \mathbf{U}'_\pi. \tag{B.1.4}$$

Proof. Let s_P and s_π denote the generation times of the packets that are delivered over the link (i, j) under policy P and policy π, respectively. From the definition of the system state, we can deduce that

$$\begin{aligned} U'_{j,P} &= \max\{U_{j,P}, s_P\}, \\ U'_{j,\pi} &= \max\{U_{j,\pi}, s_\pi\}. \end{aligned} \tag{B.1.5}$$

Hence, we have two cases:

Case 1: If $s_P \geq s_\pi$. From (B.1.3), we have

$$U_{j,P} \geq U_{j,\pi}. \tag{B.1.6}$$

Also, $s_P \geq s_\pi$, together with (B.1.5) and (B.1.6) imply

$$U'_{j,P} \geq U'_{j,\pi}. \tag{B.1.7}$$

Since there is no packet delivery under other links, we get

$$U'_{k,P} = U_{k,P} \geq U_{k,\pi} = U'_{k,\pi}, \quad \forall k \neq j. \tag{B.1.8}$$

Hence, we have

$$\mathbf{U'}_P \geq \mathbf{U'}_\pi. \tag{B.1.9}$$

Case 2: If $s_P < s_\pi$. By the definition of the system state, $s_P \leq U_{i,P}$ and $s_\pi \leq U_{i,\pi}$. Then, using $U_{i,P} \geq U_{i,\pi}$, we obtain

$$s_P < s_\pi \leq U_{i,\pi} \leq U_{i,P}. \tag{B.1.10}$$

Because $s_P < U_{i,P}$, policy P is sending a stale packet on link (i, j). By the definition of policy P, this happens only when all packets that are generated after s_P in the queue of the link (i, j) have been delivered to node j. Since $s_\pi \leq U_{i,P}$, node i has already received a packet (say packet w) generated no earlier than s_π in policy P. Because $s_P < s_\pi$, packet w is generated after s_P. Hence, packet w must have been delivered to node j in policy P such that

$$s_\pi \leq U_{j,P}. \tag{B.1.11}$$

Also, from (B.1.3), we have

$$U_{j,\pi} \leq U_{j,P}. \tag{B.1.12}$$

Combining (B.1.11) and (B.1.12) with (B.1.5), we obtain

$$U'_{j,P} \geq U'_{j,\pi}. \tag{B.1.13}$$

Since there is no packet delivery under other links, we get

$$U'_{k,P} = U_{k,P} \geq U_{k,\pi} = U'_{k,\pi}, \quad \forall k \neq j. \tag{B.1.14}$$

Hence, we have

$$\mathbf{U'}_P \geq \mathbf{U'}_\pi, \tag{B.1.15}$$

which complete the proof. □

Lemma B.1.3. *Suppose that under policy P, $\mathbf{U'}_P$ is obtained by the arrival of a new packet to node 0 in the system whose state is $\mathbf{U}_P$. Further, suppose that under policy π, $\mathbf{U'}_\pi$ is obtained by the arrival of a new packet to node 0 in the system whose state* *is* $\mathbf{U}_\pi$. *If*

$$\mathbf{U}_P \geq \mathbf{U}_\pi, \tag{B.1.16}$$

then,

$$\mathbf{U'}_P \geq \mathbf{U'}_\pi. \tag{B.1.17}$$

Proof. Let s denote the generation time of the new arrived packet. From the definition of the system state, we can deduce that

$$\begin{aligned} U'_{0,P} &= \max\{U_{0,P}, s\}, \\ U'_{0,\pi} &= \max\{U_{0,\pi}, s\}. \end{aligned} \tag{B.1.18}$$

Combining this with (B.1.16), we obtain

$$U'_{0,P} \geq U'_{0,\pi}. \tag{B.1.19}$$

Since there is no packet delivery under other links, we get

$$U'_{k,P} = U_{k,P} \geq U_{k,\pi} = U'_{k,\pi}, \quad \forall k \neq 0. \tag{B.1.20}$$

Hence, we have

$$\mathbf{U'}_P \geq \mathbf{U'}_\pi, \tag{B.1.21}$$

which complete the proof. □

Proof of Lemma B.1.1. For any sample path, we have that $\mathbf{U}_P(0^-) = \mathbf{U}_\pi(0^-)$. This, together with Lemma B.1.2 and Lemma B.1.3, implies that

$$[\mathbf{U}_P(t)|\mathcal{I}] \geq [\mathbf{U}_\pi(t)|\mathcal{I}],$$

holds for all $t \in [0, \infty)$. Hence, (B.1.2) holds which implies (B.1.1) by Theorem 6.B.30 in [47]. This completes the proof. □

Proof of Theorem 3.3.1. According to Lemma B.1.1, we have

$$[\{\mathbf{U}_P(t), t \in [0, \infty)\}|\mathcal{I}] \geq_{\text{st}} [\{\mathbf{U}_\pi(t), t \in [0, \infty)\}|\mathcal{I}],$$

holds for all work-conserving policies π, which implies

$$[\{\mathbf{\Delta}_P(t), t \in [0, \infty)\}|\mathcal{I}]\leq_{\text{st}}[\{\mathbf{\Delta}_\pi(t), t \in [0, \infty)\}|\mathcal{I}],$$

holds for all work-conserving policies π.

Finally, transmission idling only postpones the delivery of fresh packets. Therefore, the age under non-work-conserving policies will be greater. As a result,

$$[\{\mathbf{\Delta}_P(t), t \in [0, \infty)\}|\mathcal{I}]\leq_{\text{st}}[\{\mathbf{\Delta}_\pi(t), t \in [0, \infty)\}|\mathcal{I}],$$

holds for all $\pi \in \Pi$. This completes the proof. □

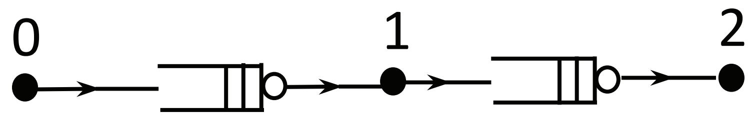
B.2 Lower bound construction

Let $v_{lj}(\pi)$ denote the transmission starting time of packet l over the incoming link to node j under policy π. We construct an infeasible policy which provides the age lower bound as follows:

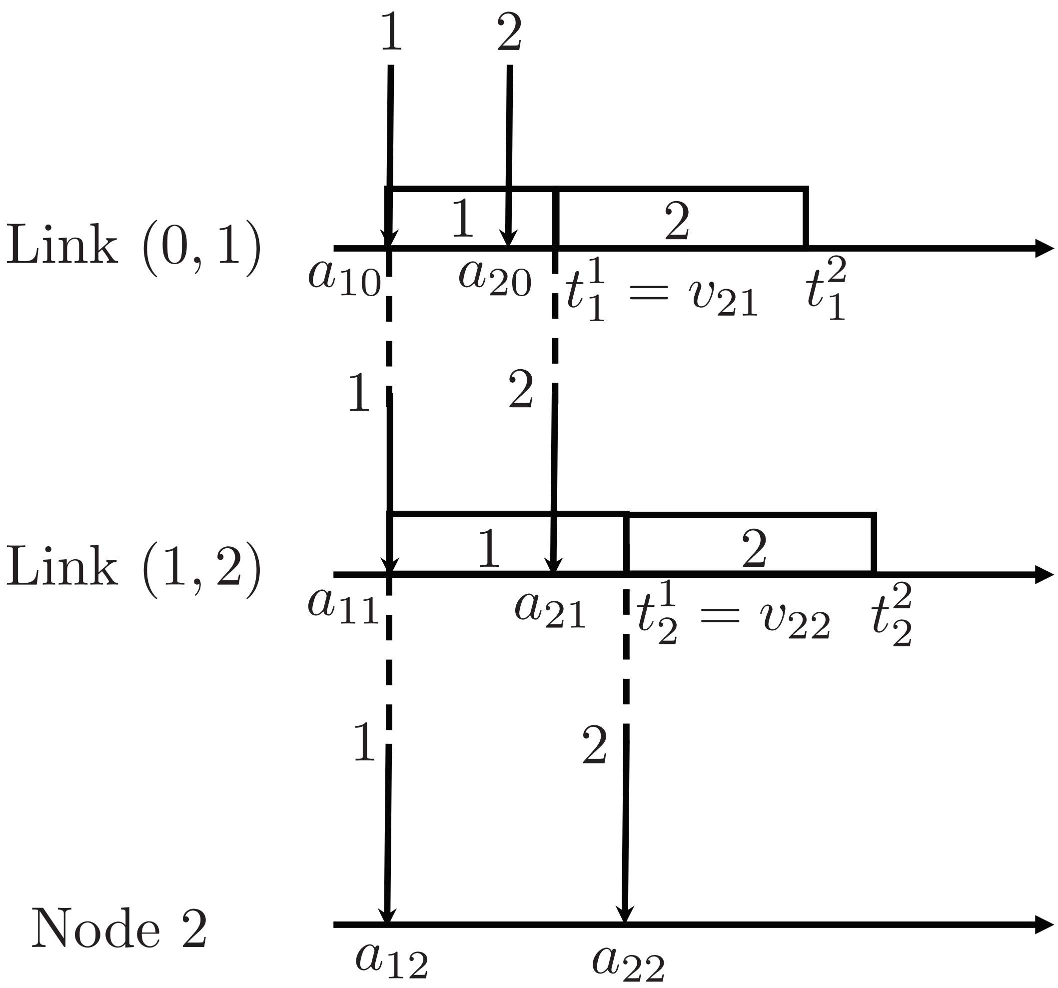
1- The infeasible policy (IP) is constructed as follows. At each link (i, j), the *packets are served by following a work-conserving LGFS principle. A packet l is deemed delivered from node i to node j once the transmission of packet l starts over the link (i, j) (this step is infeasible). After the transmission of packet l starts over the link (i, j), the link (i, j) will be busy for a time duration equal to the actual transmission time of packet l over the link (i, j). Hence, the next packet cannot start its transmission over the link (i, j) until the end of this time duration. We use $v_{lj}(IP)$ to denote the transmission starting time of packet l over the incoming link to node j under the infeasible policy (IP) constructed above.*

One example of the infeasible policy IP is illustrated in Fig. B.1, where we consider two hops of tandem queues. We use t_j^l to denote the time by which the incoming link to node j becomes idle again after the transmission of packet l starts. Since all links are idle at the beginning, packet 1 arrives to all nodes once it arrives to node 0 at time a_{10} (this is because each packet is deemed delivered to the next node once its transmission starts). However, each link is kept busy for a time duration equal to the actual transmission time of packet 1 over each link. Then, packet 2 arrives to node 0 at time a_{20} and finds the link $(0, 1)$ busy. Therefore, packet 2 cannot start its transmission until link $(0, 1)$ becomes idle again at time t_1^1 ($v_{21}(IP) = t_1^1$). Once packet 2 starts its transmission at time t_1^1 over the link $(0, 1)$, it is deemed delivered to node 1 ($a_{21}(IP) = v_{21}(IP) = t_1^1$) and link $(0, 1)$ is kept busy until time t_1^2. At time t_1^1, link $(1, 2)$ is busy. Thus, packet 2 cannot start its transmission over the link $(1, 2)$ until it becomes idle again at time t_2^1. Once packet 2 starts its transmission over the link $(1, 2)$ at time t_2^1, it is deemed delivered to node 2 ($a_{22}(IP) = v_{22}(IP) = t_2^1$).

2- *The age lower bound is constructed as follows. For each node $j \in \mathcal{V}$, define a*



(a) Two-hop tandem network.



(b) A sample path of the packet arrival processes.

Figure B.1: An illustration of the infeasible policy in a Two-hop network.

function $\Delta^{LB}_{j,IP}(t)$ *as*

$$\Delta^{LB}_{j,IP}(t) = t - \max\{s_l : v_{lj}(IP) \leq t\}. \quad \text{(B.2.1)}$$

The definition of the $\Delta^{LB}_{j,IP}(t)$ *is similar to that of the age in* (3.2.1) *except that the packets arrival times to node* j *are replaced by their transmission starting times over the incoming link to node* j *in the infeasible policy. In this case,* $\Delta^{LB}_{j,IP}(t)$ *increases linearly with* t *but is reset to a smaller value with the transmission start of a fresher packet over the incoming link to node* j*, as shown in Fig. B.2. The process of* $\Delta^{LB}_{j,IP}(t)$ *is given by* $\Delta^{LB}_{j,IP} = \{\Delta^{LB}_{j,IP}(t), t \in [0, \infty)\}$ *for each* $j \in \mathcal{V}$*. The age lower bound vector of all the network nodes is*

$$\boldsymbol{\Delta}^{LB}_{IP}(t) = (\Delta^{LB}_{0,IP}(t), \Delta^{LB}_{1,IP}(t), \ldots, \Delta^{LB}_{N-1,IP}(t)). \quad \text{(B.2.2)}$$

The age lower bound process of all the network nodes is given by

$$\boldsymbol{\Delta}^{LB}_{IP} = \{\boldsymbol{\Delta}^{LB}_{IP}(t), t \in [0, \infty)\}. \quad \text{(B.2.3)}$$

The next Lemma tells us that the process $\boldsymbol{\Delta}^{LB}_{IP}$ *is an age lower bound of all policies in* Π *in the following sense.*

Lemma B.2.1. *Suppose that the packet transmission times are NBU, independent across links, and i.i.d. across time, then for all* $\mathcal{I}'$ *satisfying* $B_{ij} \geq 1$ *for each* $(i, j) \in \mathcal{L}$*, and* $\pi \in \Pi$

$$[\boldsymbol{\Delta}^{LB}_{IP} | \mathcal{I}'] \leq_{st} [\boldsymbol{\Delta}_{\pi} | \mathcal{I}']. \quad \text{(B.2.4)}$$

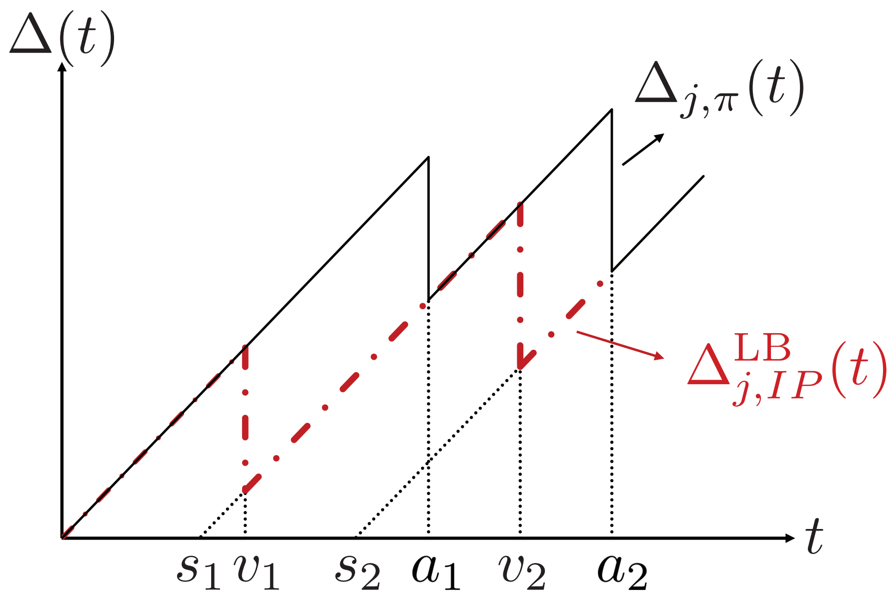


Figure B.2: The evolution of $\Delta^{\text{LB}}_{j,IP}(t)$ and $\Delta_{j,\pi}(t)$ at node $j \in \mathcal{H}_1$. For figure clarity, we use v_1 and v_2 to denote $v_{1j}(IP)$ and $v_{2j}(IP)$, respectively. Also, we use a_1 and a_2 to denote $a_{1j}(\pi)$ and $a_{2j}(\pi)$, respectively. We suppose that $a_{10} > s_1$ and $a_{20} > a_1 > s_2$, such that $a_{10} = v_1$ and $a_{20} = v_2$.

Proof. Condition (3.3.4) is very crucial in proving Lemma B.2.1. In particular, (3.3.4) implies that for NBU service time distributions, the remaining service time of a packet that has already spent τ seconds in service is probably shorter than the service time of a new packet (i.e., $\mathbb{P}[X > t+\tau | X > \tau] \leq \mathbb{P}[X > t]$, where X represents the service time). This is used to show that the transmission starting times of the fresh packets under policy IP are stochastically smaller than their corresponding delivery times under policy π, and hence (B.2.4) follows. For more details, see Appendix B.3. □

B.3 Proof of Lemma B.2.1

For notation simplicity, let policy IP represent the infeasible policy. We need to define the following parameters: Recall that v_{lj} denotes the transmission starting time of packet l over the incoming link to node j and a_{lj} denotes the arrival time of packet l to node j. We define Γ_{lj} and D_{lj} as

$$\Gamma_{lj} = \min_{q \geq l} \{v_{qj}\}, \tag{B.3.1}$$

$$D_{lj} = \min_{q \geq l} \{a_{qj}\}, \tag{B.3.2}$$

where Γ_{lj} and D_{lj} are the smallest transmission starting time over the incoming link to node j and arrival time to node j, respectively, of all packets that are fresher than the packet l. An illustration of these parameters is provided in Fig. B.3. Suppose that there are n update packets, where n is an arbitrary positive integer, no matter finite or infinite. Define the vectors $\mathbf{\Gamma}_j = (\Gamma_{1j}, \ldots, \Gamma_{nj})$, and $\mathbf{D}_j = (D_{1j}, \ldots, D_{nj})$. Also, a packet l is said to be an informative packet at node i, if all packets that arrive to node i before packet l are staler than packet l, i.e., $s_{l'} \leq s_l$ for all packets l' satisfying $a_{l'i} \leq a_{li}$. All these quantities are functions of the scheduling policy π (except the

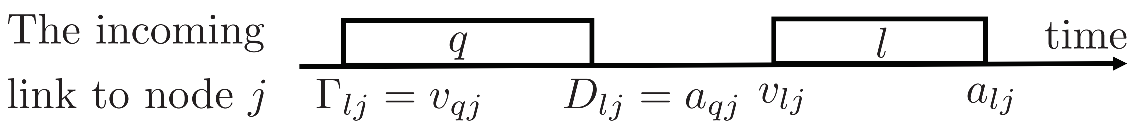


Figure B.3: An illustration of v_{lj}, a_{lj}, Γ_{lj} and D_{lj}. We consider the incoming link to node j, and $s_q > s_l$. The transmission starting time over this link and the arrival time to node j of packet q are earlier than those of packet l. Thus, we have $\Gamma_{lj} = v_{qj}$ and $D_{lj} = a_{qj}$.

packet arrival times $(a_{10}, a_{20}, \ldots, a_{n0})$ *to node 0 which are invariant of the scheduling policy).*

We can deduce from (3.2.1) *that the age process* $\mathbf{\Delta}_\pi$ *under any policy* π *is increasing in* $[\mathbf{D}_1(\pi), \ldots, \mathbf{D}_{N-1}(\pi)]$*. Moreover, we can deduce from* (B.2.1) *that the process* $\mathbf{\Delta}_{IP}^{LB}$ *is increasing in* $[\mathbf{\Gamma}_1(IP), \ldots, \mathbf{\Gamma}_{N-1}(IP)]$*. According to Theorem 6.B.16.(a) of [47], if we can show*

$$[\mathbf{\Gamma}_1(IP), \ldots, \mathbf{\Gamma}_{N-1}(IP)|\mathcal{I}'] \leq_{st} [\mathbf{D}_1(\pi), \ldots, \mathbf{D}_{N-1}(\pi)|\mathcal{I}'], \tag{B.3.3}$$

holds for all $\pi \in \Pi$*, then* (B.2.4) *is proven. Hence,* (B.3.3) *is what we need to show. We pick an arbitrary policy* $\pi \in \Pi$ *and prove* (B.3.3) *into two steps:*

Step 1*: We first show that, at any link* (i, j)*, if the arrival times of the informative packets at node* i *under policy* IP *are earlier than those of the informative packets at node* i *under policy* π*, then the arrival times of the informative packets at node* j *under policy* IP *are earlier than those of the informative packets at node* j *under policy* π*. Observe that the vector* $\mathbf{\Gamma}_i(IP)$ *represents the arrival times of the informative packets at node* i *under policy* IP *(recall the construction of the infeasible policy* IP *and* *its age evolution in* (B.2.1)*), while the vector* $\mathbf{D}_i(\pi)$ *represents the arrival times of the informative packets at node* i *under policy* π*. Then, the previous statement is manifested in the following lemma.*

Lemma B.3.1. *For any link* $(i, j) \in \mathcal{L}$*, if (i) the packet transmission times are NBU, and (ii)* $\mathbf{\Gamma}_i(IP) \leq \mathbf{D}_i(\pi)$*, then*

$$[\mathbf{\Gamma}_j(IP)|\mathcal{I}'] \leq_{st} [\mathbf{D}_j(\pi)|\mathcal{I}'], \tag{B.3.4}$$

holds for all $\mathcal{I}'$ *satisfying* $B_{ij} \geq 1$.

Proof. See Appendix B.4 □

Step 2*: We use Lemma B.3.1 to prove* (B.3.3)*. Consider a node* $j \in \mathcal{H}_k$*. We prove* (B.3.3) *using Theorem 6.B.3 and Theorem 6.B.16.(c) of [47] into two steps:*

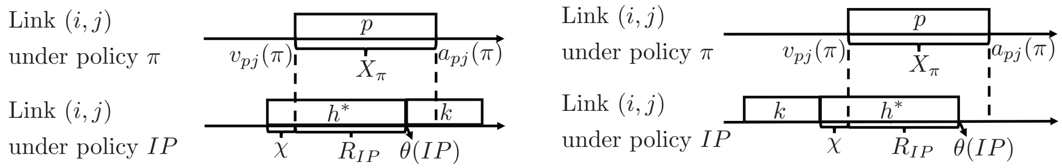
Step A*: Consider node* $i_{j,1}$*. Observe that node* $i_{j,1}$ *receives update packets from node 0. Since the packet arrival times* $(a_{10}, \ldots, a_{n0})$ *to node 0 are invariant of the scheduling policy, both conditions of Lemma B.3.1 are satisfied and we can apply it on the link* $(0, i_{j,1})$ *to obtain*

$$[\mathbf{\Gamma}_{i_{j,1}}(IP)|\mathcal{I}'] \leq_{st} [\mathbf{D}_{i_{j,1}}(\pi)|\mathcal{I}']. \tag{B.3.5}$$

Step B*: Consider node* $i_{j,m}$*, where* $2 \leq m \leq k$*. We need to prove that*

$$\begin{aligned}&[\mathbf{\Gamma}_{i_{j,m}}(IP)|\mathcal{I}', \mathbf{\Gamma}_{i_{j,1}}(IP)=\gamma_{i_{j,1}}, \ldots, \mathbf{\Gamma}_{i_{j,m-1}}(IP)=\gamma_{i_{j,m-1}}]\\ &\leq_{st}[\mathbf{D}_{i_{j,m}}(\pi)|\mathcal{I}', \mathbf{D}_{i_{j,1}}(\pi)=\mathbf{d}_{i_{j,1}}, \ldots, \mathbf{D}_{i_{j,m-1}}(\pi)=\mathbf{d}_{i_{j,m-1}}],\\ &\textit{whenever} \quad \gamma_{i_{j,t}} \leq \mathbf{d}_{i_{j,t}}, t = 1, \ldots, m-1.\end{aligned} \tag{B.3.6}$$

Since node $i_{j,m}$ *receives update packets from node* $i_{j,m-1}$ *and* $\mathbf{\Gamma}_{i_{j,m-1}}(IP) \leq \mathbf{D}_{i_{j,m-1}}(\pi)$ *in* (B.3.6)*, both conditions of Lemma B.3.1 are satisfied in this case as well (in*



(a) Case 1: Link (i, j) sends packet k after the time $\theta(IP)$.

(b) Case 2: Link (i, j) sends packet k before the time IP.

Figure B.4: Illustration of packet transmissions under policy π and policy IP. In policy π, link (i, j) starts to send packet p at time $v_{pj}(\pi)$ and will complete its transmission at time $a_{pj}(\pi)$. Hence, the transmission duration of packet p is $[v_{pj}(\pi), a_{pj}(\pi)]$ in policy π. Under policy IP, link (i, j) is kept busy before time $v_{pj}(\pi)$ for a time duration equal to the actual transmission time of packet h^* and becomes available to send a new packet at the time $\theta(IP) < a_{pj}(\pi)$.

particular, for the link $(i_{j,m-1}, i_{j,m})$*), and we can use it to prove* (B.3.6)*. By using* (B.3.5) *and* (B.3.6) *with Theorem 6.B.3 of [47], we can show*

$$[\mathbf{\Gamma}_{i_{j,1}}(IP), \ldots, \mathbf{\Gamma}_{i_{j,k}}(IP)|\mathcal{I}'] \leq_{st} [\mathbf{D}_{i_{j,1}}(\pi), \ldots, \mathbf{D}_{i_{j,k}}(\pi)|\mathcal{I}']. \tag{B.3.7}$$

Following the previous argument, we can show that (B.3.7) *holds for all* $j \in \mathcal{V}$*. Note that the transmission times are independent across links. Using this with Theorem 6.B.3 and Theorem 6.B.16.(c) of [47], we prove* (B.3.3)*. This completes the proof.* □

B.4 Proof of Lemma B.3.1

The proof of Lemma B.3.1 is motivated by the proof idea of [15, Lemma 1] and [89, Lemma 2]. We prove (B.3.4) *using Theorem 6.B.3 of [47] into two steps.*

Step 1*: Consider packet 1. Note that packet 1 may not be the first packet to arrive at node* i *under policy* IP*. Thus, we use* l^* *to denote the index of the first arrived packet at node* i *under policy* IP*, where* $s_{l^*} \geq s_1$*. From the construction of the policy*

IP and (B.3.1)*,* $\Gamma_{1i}(IP)$ *is the arrival time of the first arrived packet at node* i *under policy* IP*. Since the link* (i, j) *is idle before the arrival of the first arrived packet at node* i*, and policy* IP *is a work-conserving policy, packet* l^* *will start its transmission under policy* IP *over the link* (i, j) *once it arrives to node* i *(at time* $\Gamma_{1i}(IP)$*). Thus, from* (B.3.1)*, we obtain*

$$[\Gamma_{1j}(IP)|\mathcal{I}'] = [v_{l^*j}(IP)|\mathcal{I}'] = [\Gamma_{1i}(IP)|\mathcal{I}']. \tag{B.4.1}$$

Observe that we have

$$[\Gamma_{1i}(IP)|\mathcal{I}'] \leq [D_{1i}(\pi)|\mathcal{I}']. \tag{B.4.2}$$

Also, we must have

$$[D_{1i}(\pi)|\mathcal{I}'] \leq [D_{1j}(\pi)|\mathcal{I}'], \tag{B.4.3}$$

because a packet must spend a time over the link (i, j) *(its transmission time over the link* (i, j)*) before it is delivered from node* i *to node* j *under policy* π*. Combining* (B.4.1)*,* (B.4.2)*, and* (B.4.3)*, we get*

$$[\Gamma_{1j}(IP)|\mathcal{I}'] = [\Gamma_{1i}(IP)|\mathcal{I}'] \leq [D_{1i}(\pi)|\mathcal{I}'] \leq [D_{1j}(\pi)|\mathcal{I}']. \tag{B.4.4}$$

Step 2*: Consider a packet* p*, where* $2 \leq p \leq n$*. We suppose that no packet with generation time greater than* s_p *has arrived to node* j *before packet* p *under policy* π*.*

We need to prove that

$$\begin{aligned}
&[\Gamma_{pj}(IP)|\mathcal{I}', \Gamma_{1j}(IP) = \gamma_1, \ldots, \Gamma_{(p-1)j}(IP) = \gamma_{p-1}] \\
\leq_{st} &[D_{pj}(\pi)|\mathcal{I}', D_{1j}(\pi) = d_1, \ldots, D_{(p-1)j}(\pi) = d_{p-1}], \\
&\textit{whenever} \quad \gamma_l \leq d_l, l = 1, 2, \ldots, p-1.
\end{aligned} \tag{B.4.5}$$

For notation simplicity, define $\Gamma^{p-1} \triangleq \{\Gamma_{1j}(IP) = \gamma_1, \ldots, \Gamma_{(p-1)j}(IP) = \gamma_{p-1}\}$ *and* $D^{p-1} \triangleq \{D_{1j}(\pi) = d_1, \ldots, D_{(p-1)j}(\pi) = d_{p-1}\}$. *We will show that the link* (i, j) *under policy* IP *can send a new packet at a time that is stochastically smaller than the arrival time of packet* p *at node* j *under policy* π. *At this time, there are two possible cases under policy* IP. *One of them is that the link* (i, j) *sends a packet with generation time greater than* s_p. *The other one is that the link* (i, j) *sends a packet with generation time less than* s_p *or there is no packet to be sent. We will show that* (B.4.5) *holds in either case.*

As illustrated in Fig. B.4, suppose that under policy π, *link* (i, j) *starts to send packet* p *at time* $v_{pj}(\pi)$ *and will complete its transmission at time* $a_{pj}(\pi)$. *Under policy* IP, *define* $h^* = \arg\max_h \{v_{hj}(IP) : v_{hj}(IP) \leq v_{pj}(\pi)\}$ *as the index of the last packet whose transmission starts over the link* (i, j) *before time* $v_{pj}(\pi)$. *Note that the link* (i, j) *under policy* IP *is kept busy after time* $v_{h^*j}(IP)$ *for a time duration equal to the actual transmission time of packet* h^* *over the link* (i, j). *Suppose that under policy* IP, *link* (i, j) *is kept busy for* χ ($\chi \geq 0$) *seconds of the actual transmission time of packet* h^* *before time* $v_{pj}(\pi)$. *Let* R_{IP} *denote the remaining busy period of the link* (i, j) *under policy* IP *after time* $v_{pj}(\pi)$ *(this remaining busy period is due to the remaining transmission time of packet* h^* *after time* $v_{pj}(\pi)$*). Hence, link* (i, j) *becomes available to send a new packet at time* $v_{pj}(\pi) + R_{IP}$. *Let* $X_\pi = a_{pj}(\pi) - v_{pj}(\pi)$ *denote the transmission time of packet* p *under policy* π *and* $X_{IP} = \chi + R_{LB}$ *denote* *the actual transmission time of packet h^*. Then, the CCDF of R_{IP} is given by*

$$\mathbb{P}[R_{IP} > s] = \mathbb{P}[X_{IP} - \chi > s | X_{IP} > \chi]. \tag{B.4.6}$$

Because the packet transmission times are NBU, we can obtain that for all $s, \chi \geq 0$

$$\begin{aligned}\mathbb{P}[X_{IP} - \chi > s | X_{IP} > \chi] &= \mathbb{P}[X_\pi - \chi > s | X_\pi > \chi] \\ &\leq \mathbb{P}[X_\pi > s].\end{aligned} \tag{B.4.7}$$

By combining (B.4.6) *and* (B.4.7), *we obtain*

$$R_{IP} \leq_{st} X_\pi, \tag{B.4.8}$$

which implies

$$v_{pj}(\pi) + R_{IP} \leq_{st} v_{pj}(\pi) + X_\pi = a_{pj}(\pi). \tag{B.4.9}$$

From (B.4.9), *we can deduce that link (i, j) becomes available to send a new packet under policy IP at a time that is stochastically smaller than the time $a_{pj}(\pi)$. Let $\theta(IP)$ denote the time that link (i, j) becomes available to send a new packet under policy IP. According to* (B.4.9), *we have*

$$\begin{aligned}&[\theta(IP)|\mathcal{I}', \Gamma^{p-1}] \leq_{st} [a_{pj}(\pi)|\mathcal{I}', D^{p-1}], \\ &\textit{whenever} \quad \gamma_l \leq d_l, l = 1, 2, \ldots, p-1.\end{aligned} \tag{B.4.10}$$

It is important to note that, since we have $[\Gamma_{pi}(IP)|\mathcal{I}'] \leq [D_{pi}(\pi)|\mathcal{I}']$, there is a packet with generation time greater than s_p is available to the link (i, j) before time $v_{pj}(\pi)$ under policy IP. At the time $\theta(IP)$, we have two possible cases under policy IP:

Case 1: Link (i,j) starts to send a fresh packet k with $k \geq p$ at the time $\theta(IP)$ under policy IP, as shown in Fig. B.4(a). Hence we obtain

$$[v_{kj}(IP)|\mathcal{I}', \Gamma^{p-1}] = [\theta(IP)|\mathcal{I}', \Gamma^{p-1}] \leq_{st} [a_{pj}(\pi)|\mathcal{I}', D^{p-1}] \quad \text{whenever} \quad \gamma_l \leq d_l, l = 1, 2, \ldots, p-1. \tag{B.4.11}$$

Since $s_k \geq s_p$, (B.3.1) *implies*

$$[\Gamma_{pj}(IP)|\mathcal{I}', \Gamma^{p-1}] \leq [v_{kj}(IP)|\mathcal{I}', \Gamma^{p-1}]. \tag{B.4.12}$$

Since there is no packet with generation time greater than s_p that has been arrived to node j before packet p under policy π, (B.3.2) *implies*

$$[D_{pj}(\pi)|\mathcal{I}', D^{p-1}] = [a_{pj}(\pi)|\mathcal{I}', D^{p-1}]. \tag{B.4.13}$$

By combining (B.4.11)*,* (B.4.12)*, and* (B.4.13)*,* (B.4.5) *follows.*

Case 2: Link (i,j) starts to send a stale packet (with generation time smaller than s_p) or there is no packet transmission over the link (i,j) at the time $\theta(LB)$ under policy IP. Since the packets are served by following a work-conserving LGFS principle under policy IP, and a packet with generation time greater than s_p is available to the link (i,j) before time $v_{pj}(\pi)$ under policy IP, the link (i,j) must have sent a fresh packet k with $k \geq p$ before time $\theta(IP)$, as shown in Fig. B.4(b). Hence, we have

$$[v_{kj}(IP)|\mathcal{I}', \Gamma^{p-1}] \leq [\theta(IP)|\mathcal{I}', \Gamma^{p-1}] \leq_{st} [a_{pj}(\pi)|\mathcal{I}', D^{p-1}] \quad \text{whenever} \quad \gamma_l \leq d_l, l = 1, 2, \ldots, p-1. \tag{B.4.14}$$

Similar to Case 1, we can use (B.3.1)*,* (B.3.2)*, and* (B.4.14) *to show that* (B.4.5) *holds in this case.*

Notice that if there is a fresher packet y with $s_y > s_p$ and $a_{yj}(\pi) < a_{pj}(\pi)$ (this may occur if packet y preempts the transmission of packet p under policy π or packet y arrives to node i before packet p under policy π), then we replace packet p by packet y in the arguments and equations from (B.4.5) *to* (B.4.14) *to obtain*

$$\begin{aligned} &[\Gamma_{yj}(IP)|\mathcal{I}', \Gamma^{p-1}] \leq [D_{yj}(\pi)|\mathcal{I}', D^{p-1}] \\ &\text{whenever} \quad \gamma_l \leq d_l, l = 1, 2, \ldots, p-1. \end{aligned} \tag{B.4.15}$$

Observing that $s_y > s_p$, (B.3.1) *implies*

$$[\Gamma_{pj}(IP)|\mathcal{I}', \Gamma^{p-1}] \leq [\Gamma_{yj}(IP)|\mathcal{I}', \Gamma^{p-1}]. \tag{B.4.16}$$

Since $a_{yj}(\pi) < a_{pj}(\pi)$ and $s_y > s_p$, (B.3.2) *implies*

$$[D_{pj}(\pi)|\mathcal{I}', D^{p-1}] = [D_{yj}(\pi)|\mathcal{I}', D^{p-1}]. \tag{B.4.17}$$

By combining (B.4.15), (B.4.16), *and* (B.4.17), *we can prove* (B.4.5) *in this case too. Finally, substitute* (B.4.4) *and* (B.4.5) *into Theorem 6.B.3 of [47],* (B.3.4) *is proven.* □

B.5 Proof of Theorem 3.3.2

For notation simplicity, let policy P represent the non-prmp-LGFS policy and policy IP represent the infeasible policy (the construction of the infeasible policy and the age lower bound are provided in Appendix B.2). We will need the definitions that are provided at the beginning of Appendix B.3 throughout this proof. Consider a node $j \in \mathcal{H}_k$ with $k \geq 1$. We prove Theorem 3.3.2 into three steps:

Step 1: *We provide an upper bound on the time differences between the arrival*

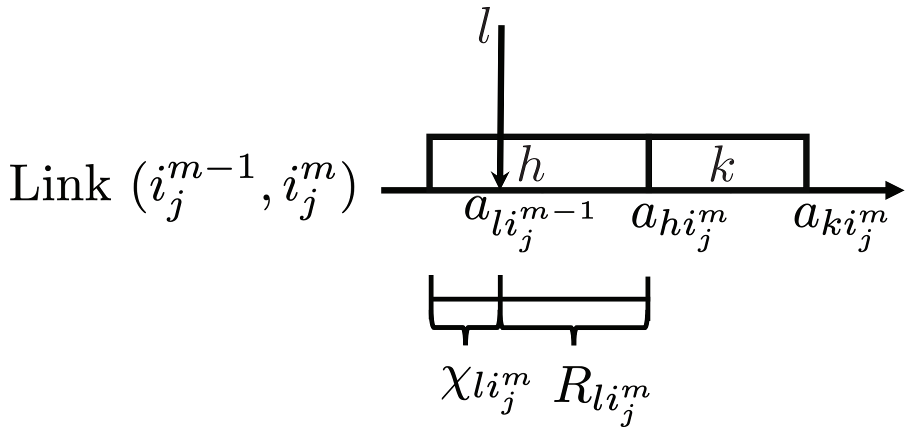


Figure B.5: An illustration of $R_{li_{j,m}}$ and $\chi_{li_{j,m}}$. Packet l arrives to node $i_{j,m-1}$ at time $a_{li_{j,m-1}}$, while packet h with $h < l$ is being transmitted over the link $(i_{j,m-1}, i_{j,m})$. After the delivery of packet h to node $i_{j,m}$ at time $a_{hi_{j,m}}$, packet k with $k \geq l$ is transmitted over the link $(i_{j,m-1}, i_{j,m})$. The duration $R_{li_{j,m}}$ is the waiting time of packet l in the queue of the link $(i_{j,m-1}, i_{j,m})$ until the packet k starts its transmission. The duration $\chi_{li_{j,m}}$ is the time spent by the link $(i_{j,m-1}, i_{j,m})$ on sending the packet h before the time $a_{li_{j,m-1}}$.

times of the informative packets at node j under policy IP and those under policy P. To achieve that, we need the following definitions. For each link in the path to node j (i.e., $(i_{j,m-1}, i_{j,m})$ for all $1 \leq m \leq k$), define $R_{li_{j,m}} = \Gamma_{li_{j,m}} - D_{li_{j,m-1}}$ as the time spent in the queue of the link $(i_{j,m-1}, i_{j,m})$ by the packet that arrives at node $i_{j,m-1}$ at time $D_{li_{j,m-1}}$, until the first transmission starting time over the link $(i_{j,m-1}, i_{j,m})$ of the packets with generation time greater than s_l. If there is a packet that is being transmitted over the link $(i_{j,m-1}, i_{j,m})$ at time $D_{li_{j,m-1}}$, let $\chi_{li_{j,m}}$ ($\chi_{li_{j,m}} \geq 0$) denote the amount of time that the link $(i_{j,m-1}, i_{j,m})$ has spent on sending this packet by the time $D_{li_{j,m-1}}$. These parameters ($R_{li_{j,m}}$ and $\chi_{li_{j,m}}$) are functions of the scheduling policy π. An illustration of these parameters is provided in Fig. B.5. Note that policy P is a LGFS work-conserving policy. Also, the packets under policy IP are served by following a work-conserving LGFS principle. Thus, we can express $R_{li_{j,m}}$ under these *policies as* $R_{li_{j,m}} = [X_{i_{j,m}} - \chi_{li_{j,m}} | X_{i_{j,m}} > \chi_{li_{j,m}}]$*. Because the packet transmission times are NBU and i.i.d. across time, for all realization of* $\chi_{li_{j,m}}$

$$[R_{li_{j,m}} | \chi_{li_{j,m}}] \leq_{st} X_{i_{j,m}}, \text{ for } m = 1, \ldots, k, \ \forall l, \tag{B.5.1}$$

which implies that

$$\mathbb{E}[R_{li_{j,m}} | \chi_{li_{j,m}}] \leq \mathbb{E}[X_{i_{j,m}}], \text{ for } m = 1, \ldots, k, \ \forall l, \tag{B.5.2}$$

holds for policy P *and policy* IP*. Define* $z_l = D_{lj}(P) - \Gamma_{lj}(IP)$*. Note that,* $\Gamma_{lj}(IP)$ *represents the arrival time at node* j *of a packet* p *with* $s_p \geq s_l$ *under policy* IP*, and* $D_{lj}(P)$ *represents the arrival time at node* j *of a packet* h *with* $s_h \geq s_l$ *under policy* P*. Therefor,* z_l*'s represent the time differences between the arrival times of the informative packets at node* j *under policy* IP *and those under policy* P*, as shown in Fig. B.6. By invoking the construction of policy* IP*, we have* $D_{lj}(IP) = \Gamma_{lj}(IP)$ *for all* l*. Using this with the definition of* $R_{li_{j,m}}$*, we can express* $\Gamma_{lj}(IP)$ *as*

$$\Gamma_{lj}(IP) = a_{l0} + \sum_{m=1}^{k} [R_{li_{j,m}}(IP) | \chi_{li_{j,m}}(IP)], \tag{B.5.3}$$

where $\Gamma_{lj}(IP)$ *is considered as the arrival time at node* j *of the first packet with generation time greater than* s_l *under policy* IP*. Also, we can express* $D_{lj}(P)$ *as*

$$D_{lj}(P) = a_{l0} + \sum_{m=1}^{k} [R_{li_{j,m}}(P) | \chi_{li_{j,m}}(P)] + \sum_{m=1}^{k} X_{i_{j,m}}. \tag{B.5.4}$$

Observing that packet arrival times $(a_{10}, a_{20}, \ldots)$ *at node* 0 *and the packet transmission times are invariant of the scheduling policy* π*. Then, from the construction of policy* IP*, we have* $[R_{li_{j,1}}(IP) | \chi_{li_{j,1}}(IP)] = [R_{li_{j,1}}(P) | \chi_{li_{j,1}}(P)]$ *for all* l *(because all nodes in*

$\mathcal{H}_1$ receive the update packets from node 0). Using this with (B.5.3) *and* (B.5.4)*, we can obtain*

$$\begin{aligned} z_l &= D_{lj}(P) - \Gamma_{lj}(IP) \\ &= \sum_{m=2}^{k}[R_{li_{j,m}}(P)|\chi_{li_{j,m}}(P)] + \sum_{m=1}^{k} X_{i_{j,m}} \\ &- \sum_{m=2}^{k}[R_{li_{j,m}}(IP)|\chi_{li_{j,m}}(IP)] \\ &\leq \sum_{m=2}^{k}[R_{li_{j,m}}(P)|\chi_{li_{j,m}}(P)] + \sum_{m=1}^{k} X_{i_{j,m}} = z_l^{'}. \end{aligned} \tag{B.5.5}$$

Since the packet transmission times are independent of the packet generation process, we also have $z_l^{'}$'s are independent of the packet generation process. In addition, from (B.5.2)*, we have*

$$\mathbb{E}[z_l^{'}] \leq \mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k}\mathbb{E}[X_{i_{j,m}}]. \tag{B.5.6}$$

Step 2: *We use Step 1 to provide an upper bound on the average gap between $\Delta_{j,IP}^{LB}$ and $\Delta_{j,P}$. This gap process is denoted by $\{G_j(t), t \in [0,\infty)\}$. The average gap is given by*

$$[\bar{G}_j|\mathcal{I}'] = \limsup_{T\to\infty}\frac{\int_0^T G_j(t)dt}{T}. \tag{B.5.7}$$

Let τ_l denote the inter-generation time between packet l and packet $l-1$ (i.e., $\tau_l = s_l - s_{l-1}$), where $\tau = \{\tau_l, l \geq 1\}$. Define $N(T) = \max\{l : s_l \leq T\}$ as the number of generated packets by time T. Note that $[0, s_{N(T)}] \subseteq [0,T]$, where the length of the

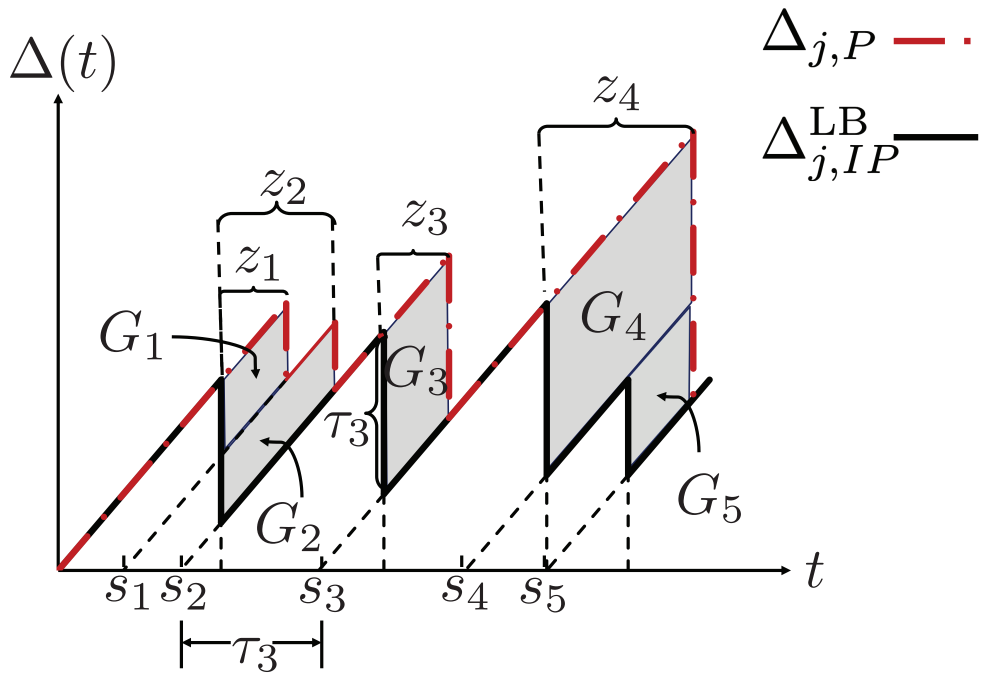


Figure B.6: The evolution $\Delta_{j,IP}^{\mathrm{LB}}$ and $\Delta_{j,P}$.

interval $[0, s_{N(T)}]$ *is* $\sum_{l=1}^{N(T)} \tau_l$. *Thus, we have*

$$\sum_{l=1}^{N(T)} \tau_l \leq T. \tag{B.5.8}$$

The area defined by the integral in (B.5.7) *can be decomposed into a sum of disjoint geometric parts. Observing Fig. B.6, the area can be approximated by the concatenation of the parallelograms* $G_1, G_2, \ldots$ *(*G_l*'s are highlighted in Fig. B.6). Note that the parallelogram* G_l *results after the generation of packet* l *(i.e., the gap that is corresponding to the packet* l*, occurs after its generation). Since the observing time* T *is chosen arbitrary, when* $T \geq s_l$*, the total area of the parallelogram* G_l *is accounted* *in the summation* $\sum_{l=1}^{N(T)} G_l$, *while it may not be accounted in the integral* $\int_0^T G_f(t)dt$. *This implies that*

$$\sum_{l=1}^{N(T)} G_l \geq \int_0^T G_j(t)dt. \tag{B.5.9}$$

Combining (B.5.8) *and* (B.5.9), *we get*

$$\frac{\int_0^T G_j(t)dt}{T} \leq \frac{\sum_{l=1}^{N(T)} G_l}{\sum_{l=1}^{N(T)} \tau_l}. \tag{B.5.10}$$

Then, take conditional expectation given τ *and* $N(T)$ *on both sides of* (B.5.10), *we obtain*

$$\begin{aligned} \frac{\mathbb{E}[\int_0^T G_j(t)dt|\tau, N(T)]}{T} &\leq \frac{\mathbb{E}[\sum_{l=1}^{N(T)} G_l|\tau, N(T)]}{\sum_{l=1}^{N(T)} \tau_l} \\ &= \frac{\sum_{l=1}^{N(T)} \mathbb{E}[G_l|\tau, N(T)]}{\sum_{l=1}^{N(T)} \tau_l}, \end{aligned} \tag{B.5.11}$$

where the second equality follows from the linearity of the expectation. From Fig. B.6, G_l *can be calculated as*

$$G_l = \tau_l z_l. \tag{B.5.12}$$

substituting by (B.5.12) *into* (B.5.11), *yields*

$$\begin{aligned} \frac{\mathbb{E}[\int_0^T G_j(t)dt|\tau, N(T)]}{T} &\leq \frac{\sum_{l=1}^{N(T)} \mathbb{E}[\tau_l z_l|\tau, N(T)]}{\sum_{l=1}^{N(T)} \tau_l} \\ &= \frac{\sum_{l=1}^{N(T)} \tau_l \mathbb{E}[z_l|\tau, N(T)]}{\sum_{l=1}^{N(T)} \tau_l}. \end{aligned} \tag{B.5.13}$$

Using (B.5.5)*, we obtain*

$$\begin{aligned}\frac{\mathbb{E}[\int_0^T G_j(t)dt|\tau, N(T)]}{T} &\leq \frac{\sum_{l=1}^{N(T)} \tau_l \mathbb{E}[z_l|\tau, N(T)]}{\sum_{l=1}^{N(T)} \tau_l} \\ &\leq \frac{\sum_{l=1}^{N(T)} \tau_l \mathbb{E}[z_l^{'}|\tau, N(T)]}{\sum_{l=1}^{N(T)} \tau_l}.\end{aligned} \tag{B.5.14}$$

Note that $z_l^{'}$*'s are independent of the packet generation process. Thus, we have* $\mathbb{E}[z_l^{'}|\tau, N(T)] = \mathbb{E}[z_l^{'}] \leq \mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}]$ *for all* l*. Using this in* (B.5.14)*, we get*

$$\begin{aligned}\frac{\mathbb{E}[\int_0^T G_j(t)dt|\tau, N(T)]}{T} &\leq \frac{\sum_{l=1}^{N(T)} \tau_l (\mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}])}{\sum_{l=1}^{N(T)} \tau_l} \\ &\leq \mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}],\end{aligned}$$

by the law of iterated expectations, we have

$$\frac{\mathbb{E}[\int_0^T G_j(t)dt]}{T} \leq \mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}]. \tag{B.5.15}$$

Taking lim sup *of both sides of* (B.5.15) *when* $T \to \infty$*, yields*

$$\limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T G_j(t)dt]}{T} \leq \mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}]. \tag{B.5.16}$$

Equation (B.5.16) *tells us that the average gap between* $\Delta_{f,IP}^{LB}$ *and* $\Delta_{f,P}$ *is no larger than* $\mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}]$.

Step 3: *We use the provided upper bound on the gap in Step 2 to prove* (3.3.5)*.*

Since $\Delta_{j,IP}^{LB}$ *is a lower bound of* $\Delta_{j,P}$*, we obtain*

$$\begin{aligned}&[\bar{\Delta}_{j,IP}^{LB}|\mathcal{I}'] \leq [\bar{\Delta}_{j,P}|\mathcal{I}'] \leq \\ &[\bar{\Delta}_{j,IP}^{LB}|\mathcal{I}'] + \mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}],\end{aligned} \tag{B.5.17}$$

where $\bar{\Delta}_{j,IP}^{LB} = \limsup_{T\to\infty} \frac{\mathbb{E}[\int_0^T \Delta_{j,IP}^{LB}(t)dt]}{T}$*. From Lemma B.2.1 in Appendix B.2, we have for all* $\mathcal{I}'$ *satisfying* $B_{ij} \geq 1$*, and* $\pi \in \Pi$

$$[\Delta_{j,IP}^{LB}|\mathcal{I}'] \leq_{st} [\Delta_{j,\pi}|\mathcal{I}'], \tag{B.5.18}$$

which implies that

$$[\bar{\Delta}_{j,IP}^{LB}|\mathcal{I}'] \leq [\bar{\Delta}_{j,\pi}|\mathcal{I}'], \tag{B.5.19}$$

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

$$[\bar{\Delta}_{j,IP}^{LB}|\mathcal{I}'] \leq \min_{\pi\in\Pi} [\bar{\Delta}_{j,\pi}|\mathcal{I}']. \tag{B.5.20}$$

Since policy P *is a feasible policy, we get*

$$\min_{\pi\in\Pi} [\bar{\Delta}_{j,\pi}|\mathcal{I}'] \leq [\bar{\Delta}_{j,P}|\mathcal{I}']. \tag{B.5.21}$$

Combining (B.5.17)*,* (B.5.20)*, and* (B.5.21)*, we get*

$$\begin{aligned}&\min_{\pi\in\Pi} [\bar{\Delta}_{j,\pi}|\mathcal{I}'] \leq [\bar{\Delta}_{j,P}|\mathcal{I}'] \leq \\ &\min_{\pi\in\Pi} [\bar{\Delta}_{j,\pi}|\mathcal{I}'] + \mathbb{E}[X_{i_{j,1}}] + 2\sum_{m=2}^{k} \mathbb{E}[X_{i_{j,m}}].\end{aligned} \tag{B.5.22}$$

Following the previous argument, we can show that (B.5.22) *holds for all* $j \in \mathcal{V}\backslash\{0\}$. *This proves* (3.3.5), *which completes the proof.* $\square$

B.6 Proof of Theorem 3.3.3

This proof is similar to that of Theorem 3.3.1. The difference between this proof and the proof of Theorem 3.3.1 is that policy π *cannot be a preemptive policy here. We will use the same definition of the system state of policy* π *used in Theorem 3.3.1. For notational simplicity, let policy* P *represent the non-preemptive LGFS policy.*

The key step in the proof of Theorem 3.3.3 is the following lemma, where we compare policy P *with an arbitrary policy* $\pi \in \Pi_{npwc}$.

Lemma B.6.1. *Suppose that* $\mathbf{U}_P(0^-) = \mathbf{U}_\pi(0^-)$ *for all* $\pi \in \Pi_{npwc}$, *then for all* $\mathcal{I}$,

$$[\{\mathbf{U}_P(t), t \in [0, \infty)\}|\mathcal{I}] \geq_{st} [\{\mathbf{U}_\pi(t), t \in [0, \infty)\}|\mathcal{I}]. \tag{B.6.1}$$

We use coupling and forward induction to prove Lemma B.6.1. For any work-conserving policy π, *suppose that stochastic processes* $\widetilde{\mathbf{U}}_P(t)$ *and* $\widetilde{\mathbf{U}}_\pi(t)$ *have the same distributions with* $\mathbf{U}_P(t)$ *and* $\mathbf{U}_\pi(t)$, *respectively. The state processes* $\widetilde{\mathbf{U}}_P(t)$ *and* $\widetilde{\mathbf{U}}_\pi(t)$ *are coupled in the following manner: If a packet is delivered from node* i *to node* j *at time* t *as* $\widetilde{\mathbf{U}}_P(t)$ *evolves in policy prmp-LGFS, then there exists a packet delivery from node* i *to node* j *at time* t *as* $\widetilde{\mathbf{U}}_\pi(t)$ *evolves in policy* π. *Such a coupling is valid since the transmission time distribution at each link is identical under all policies. Moreover, policy* π *can not be either preemptive or non-work-conserving policy, and both policies have the same packets generation times* $(s_1, s_2, \ldots, s_n)$ *at the exterenal source and packet arrival times* $(a_{10}, a_{20}, \ldots, a_{n0})$ *to node 0. According to Theorem*

6.B.30 in [47], if we can show

$$\mathbb{P}[\widetilde{\mathbf{U}}_P(t) \geq \widetilde{\mathbf{U}}_\pi(t), t \in [0, \infty)|\mathcal{I}] = 1, \tag{B.6.2}$$

then (B.6.1) *is proven.*

To ease the notational burden, we will omit the tildes henceforth on the coupled versions and just use $\mathbf{U}_P(t)$ *and* $\mathbf{U}_\pi(t)$*.*

Next, we use the following lemmas to prove (B.6.2)*:*

Lemma B.6.2. *Suppose that under policy P, $\mathbf{U}_P(\nu)$ is obtained by a packet delivery over the link (i, j) at time ν in the system whose state is $\mathbf{U}_P(\nu^-)$. Further, suppose that under policy π, $\mathbf{U}_\pi(\nu)$ is obtained by a packet delivery over the link (i, j) at time ν in the system whose state is $\mathbf{U}_\pi(\nu^-)$. If*

$$\mathbf{U}_P(t) \geq \mathbf{U}_\pi(t), \tag{B.6.3}$$

holds for all $t \in [0, \nu^-]$, then

$$\mathbf{U}_P(\nu) \geq \mathbf{U}_\pi(\nu). \tag{B.6.4}$$

Proof. Let s_P and s_π denote the packet indexes and the generation times of the delivered packets over the link (i, j) at time ν under policy P and policy π, respectively. From the definition of the system state, we can deduce that

$$\begin{aligned} U_{j,P}(\nu) &= \max\{U_{j,P}(\nu^-), s_P\}, \\ U_{j,\pi}(\nu) &= \max\{U_{j,\pi}(\nu^-), s_\pi\}. \end{aligned} \tag{B.6.5}$$

Hence, we have two cases:

Case 1: If $s_P \geq s_\pi$. From (B.6.3), we have

$$U_{j,P}(\nu^-) \geq U_{j,\pi}(\nu^-). \tag{B.6.6}$$

By $s_P \geq s_\pi$, (B.6.5), and (B.6.6), we have

$$U_{j,P}(\nu) \geq U_{j,\pi}(\nu). \tag{B.6.7}$$

Since there is no packet delivery under other links, we get

$$U_{k,P}(\nu) = U_{k,P}(\nu^-) \geq U_{k,\pi}(\nu^-) = U_{k,\pi}(\nu), \quad \forall k \neq j. \tag{B.6.8}$$

Hence, we have

$$\mathbf{U}_P(\nu) \geq \mathbf{U}_\pi(\nu). \tag{B.6.9}$$

Case 2: If $s_P < s_\pi$. Let a_π represent the arrival time of packet s_π to node i under policy π. The transmission starting time of the delivered packets over the link (i, j) is denoted by τ under both policies. Apparently, $a_\pi \leq \tau \leq \nu^-$. Since packet s_π arrived to node i at time a_π in policy π, we get

$$s_\pi \leq U_{i,\pi}(a_\pi). \tag{B.6.10}$$

From (B.6.3), we obtain

$$U_{i,\pi}(a_\pi) \leq U_{i,P}(a_\pi). \tag{B.6.11}$$

Combining (B.6.10) and (B.6.11), yields

$$s_\pi \leq U_{i,P}(a_\pi). \tag{B.6.12}$$

Hence, in policy P, node i has a packet with generation time no smaller than s_π by the time a_π. Because the $U_{i,P}(t)$ is a non-decreasing function of t and $a_\pi \leq \tau$, we have

$$U_{i,P}(a_\pi) \leq U_{i,P}(\tau). \tag{B.6.13}$$

Then, (B.6.12) and (B.6.13) imply

$$s_\pi \leq U_{i,P}(\tau). \tag{B.6.14}$$

Since $s_P < s_\pi$, (B.6.14) tells us

$$s_P < U_{i,P}(\tau), \tag{B.6.15}$$

and hence policy P is sending a stale packet on link (i, j). By the definition of policy P, this happens only when all packets that are generated after s_P in the queue of the link (i, j) have been delivered to node j by time τ. In addition, (B.6.14) tells us that by time τ, node i has already received a packet (say packet h) generated no earlier than s_π in policy P. By $s_P < s_\pi$, packet h is generated after s_P. Hence, packet h must have been delivered to node j by time τ in policy P such that

$$s_\pi \leq U_{j,P}(\tau). \tag{B.6.16}$$

Because the $U_{j,P}(t)$ is a non-decreasing function of t, and $\tau \leq \nu^-$, (B.6.16) implies

$$s_\pi \leq U_{j,P}(\nu^-). \tag{B.6.17}$$

Also, from (B.6.3), we have

$$U_{j,\pi}(\nu^-) \leq U_{j,P}(\nu^-). \tag{B.6.18}$$

Combining (B.6.17) and (B.6.18) with (B.6.5), we obtain

$$U_{j,P}(\nu) \geq U_{j,\pi}(\nu). \tag{B.6.19}$$

Since there is no packet delivery under other links, we get

$$U_{k,P}(\nu) = U_{k,P}(\nu^-) \geq U_{k,\pi}(\nu^-) = U_{k,\pi}(\nu), \quad \forall k \neq j. \tag{B.6.20}$$

Hence, we have

$$\mathbf{U}_P(\nu) \geq \mathbf{U}_\pi(\nu), \tag{B.6.21}$$

which complete the proof. □

Lemma B.6.3. *Suppose that under policy P, $\mathbf{U}'_P$ is obtained by the arrival of a new packet to node 0 in the system whose state is $\mathbf{U}_P$. Further, suppose that under policy π, $\mathbf{U}'_\pi$ is obtained by the arrival of a new packet to node 0 in the system whose state is $\mathbf{U}_\pi$. If*

$$\mathbf{U}_P \geq \mathbf{U}_\pi, \tag{B.6.22}$$

then,

$$\mathbf{U'}_P \geq \mathbf{U'}_\pi. \tag{B.6.23}$$

Proof. The proof of Lemma B.6.3 is similar to that of Lemma B.1.3, and hence is not provided. □

Proof of Lemma B.6.1. For any sample path, we have that $\mathbf{U}_P(0^-) = \mathbf{U}_\pi(0^-)$. This, together with Lemma B.6.2 and Lemma B.6.3, implies that

$$[\mathbf{U}_P(t)|\mathcal{I}] \geq [\mathbf{U}_\pi(t)|\mathcal{I}],$$

holds for all $t \in [0, \infty)$. Hence, (B.6.2) holds which implies (B.6.1) by Theorem 6.B.30 in [47]. This completes the proof. □

Proof of Theorem 3.3.3. According to Lemma B.6.1, we have

$$[\{\mathbf{U}_P(t), t \in [0, \infty)\}|\mathcal{I}] \geq_{\text{st}} [\{\mathbf{U}_\pi(t), t \in [0, \infty)\}|\mathcal{I}],$$

holds for all $\pi \in \Pi_{npwc}$, which implies

$$[\{\boldsymbol{\Delta}_P(t), t \in [0, \infty)\}|\mathcal{I}] \leq_{\text{st}} [\{\boldsymbol{\Delta}_\pi(t), t \in [0, \infty)\}|\mathcal{I}],$$

holds for all $\pi \in \Pi_{npwc}$. This completes the proof. □

APPENDIX C: PROOFS FOR CHAPTER 4

C.1 Proof of Proposition 4.3.1

We will need the following definitions: A set $U \subseteq \mathbb{R}^n$ is called upper if $\mathbf{y} \in U$ whenever $\mathbf{y} \geq \mathbf{x}$ and $\mathbf{x} \in U$.

Definition C.1.1. ***Univariate Stochastic Ordering:*** *[47] 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 C.1.2. ***Multivariate Stochastic Ordering:*** *[47] 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*

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

Definition C.1.3. ***Stochastic Ordering of Stochastic Processes:*** *[47] 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

$$(X(t_1), X(t_2), \ldots, X(t_n)) \leq_{st} (Y(t_1), Y(t_2), \ldots, Y(t_n)), \tag{C.1.1}$$

where the multivariate stochastic ordering in (C.1.1) *was defined in Definition C.1.2.*

Now, we prove Proposition 4.3.1. Let the vector $\mathbf{\Delta}_\pi(t) = (\Delta_{[1],\pi}(t), \ldots, \Delta_{[m],\pi}(t))$ denote the system state at time t of the scheduler π, where $\Delta_{[l],\pi}(t)$ is the l-th largest age of the sources at time t under the scheduler π. Let $\{\mathbf{\Delta}_\pi(t), t \geq 0\}$ denote the state process of the scheduler π. For notational simplicity, let P represent the MAF scheduler. Throughout the proof, we assume that $\mathbf{\Delta}_\pi(0^-) = \mathbf{\Delta}_P(0^-)$ for all π and the sampler is fixed to an arbitrarily chosen one. The key step in the proof of Proposition 4.3.1 is the following lemma, where we compare the scheduler P with any arbitrary scheduler π.

Lemma C.1.1. *Suppose that $\mathbf{\Delta}_\pi(0^-) = \mathbf{\Delta}_P(0^-)$ for all scheduler π and the sampler is fixed, then we have*

$$\{\mathbf{\Delta}_P(t), t \geq 0\} \leq_{st} \{\mathbf{\Delta}_\pi(t), t \geq 0\} \tag{C.1.2}$$

We use a coupling and forward induction to prove Lemma C.1.1. For any scheduler π, suppose that the stochastic processes $\widetilde{\mathbf{\Delta}}_P(t)$ and $\widetilde{\mathbf{\Delta}}_\pi(t)$ have the same stochastic laws as $\mathbf{\Delta}_P(t)$ and $\mathbf{\Delta}_\pi(t)$. The state processes $\widetilde{\mathbf{\Delta}}_P(t)$ and $\widetilde{\mathbf{\Delta}}_\pi(t)$ are coupled such that the packet service times are equal under both scheduling policies, i.e., Y_i's are the same under both scheduling policies. Such a coupling is valid since the service time distribution is fixed under all policies. Since the sampler is fixed, such a coupling implies that the packet generation and delivery times are the same under both *schedulers. According to Theorem 6.B.30 of [47], if we can show*

$$\mathbb{P}\left[\widetilde{\boldsymbol{\Delta}}_P(t) \leq \widetilde{\boldsymbol{\Delta}}_\pi(t), t \geq 0\right] = 1, \tag{C.1.3}$$

then (C.1.2) *is proven. To ease the notational burden, we will omit the tildes on the coupled versions in this proof and just use* $\boldsymbol{\Delta}_P(t)$ *and* $\boldsymbol{\Delta}_\pi(t)$*. Next, we compare scheduler* P *and scheduler* π *on a sample path and prove* (C.1.2) *using the following lemma:*

Lemma C.1.2 (Inductive Comparison)**.** *Suppose that a packet with generation time* S *is delivered under the scheduler* P *and the scheduler* π *at the same time* t*. The system state of the scheduler* P *is* $\boldsymbol{\Delta}_P$ *before the packet delivery, which becomes* $\boldsymbol{\Delta}'_P$ *after the packet delivery. The system state of the scheduler* π *is* $\boldsymbol{\Delta}_\pi$ *before the packet delivery, which becomes* $\boldsymbol{\Delta}'_\pi$ *after the packet delivery. If*

$$\Delta_{[i],P} \leq \Delta_{[i],\pi}, i = 1, \ldots, m, \tag{C.1.4}$$

then

$$\Delta'_{[i],P} \leq \Delta'_{[i],\pi}, i = 1, \ldots, m. \tag{C.1.5}$$

Lemma C.1.2 is proven by following the proof idea of [36, Lemma 2]. For the sake of completeness, we provide proof of Lemma C.1.2 as follows:

Proof. Since only one source can be scheduled at a time and the scheduler P is the MAF one, the packet with generation time S must be generated from the source with maximum age $\Delta_{[1],P}$, call it source l^*. In other words, the age of source l^* is reduced from the maximum age $\Delta_{[1],P}$ to the minimum age $\Delta'_{[m],P} = t - S$, and the age of the other $(m-1)$ sources remain unchanged. Hence,

$$\begin{aligned}
&\Delta'_{[i],P} = \Delta_{[i+1],P}, i = 1, \ldots, m-1, \\
&\Delta'_{[m],P} = t - S.
\end{aligned} \tag{C.1.6}$$

In the scheduler π, this packet can be generated from any source. Thus, for all cases of scheduler π, it must hold that

$$\Delta'_{[i],\pi} \geq \Delta_{[i+1],\pi}, i = 1, \ldots, m-1. \tag{C.1.7}$$

By combining (C.1.4), (C.1.6), and (C.1.7), we have

$$\Delta'_{[i],\pi} \geq \Delta_{[i+1],\pi} \geq \Delta_{[i+1],P} = \Delta'_{[i],P}, i = 1, \ldots, m-1. \tag{C.1.8}$$

In addition, since the same packet is also delivered under the scheduler π, the source from which this packet is generated under policy π will have the minimum age after the delivery, i.e., we have

$$\Delta'_{[m],\pi} = t - S = \Delta'_{[m],P}. \tag{C.1.9}$$

By this, (C.1.5) is proven. □

Proof of Lemma C.1.1. Using the coupling between the system state processes, and for any given sample path of the packet service times, we consider two cases:

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

Case 2: When a packet is delivered, the ages of the sources evolve according to Lemma C.1.2.

By induction over time, we obtain

$$\Delta_{[i],P}(t) \leq \Delta_{[i],\pi}(t), i = 1, \ldots, m, t \geq 0. \tag{C.1.10}$$

Hence, (C.1.3) follows which implies (C.1.2) by Theorem 6.B.30 of [47]. This completes the proof. □

Proof of Proposition 4.3.1. Since the Ta-APD and Ta-AP for any scheduling policy π are the expectation of non-decreasing functional of the process $\{\boldsymbol{\Delta}_\pi(t), t \geq 0\}$, (C.1.2) implies (4.3.1) and (4.3.2) using the properties of stochastic ordering [47]. This completes the proof. □

C.2 Proof of Theorem 4.3.1

The optimality of the MAF scheduler follows from Proposition 4.3.1. Now, we need to show the optimality of the zero-wait sampler. We need to show that the Ta-APD is an increasing function of the packets waiting times Z_i's. Define K_{li} as the number of packets that have been transmitted since the last received service by source l before time D_i. Also, let γ_l be the index of the first delivered packet from source l.

For $i > \gamma_l$, the last service that source l has received before time D_i^- was at time $D_{i-K_{li}}$. Since the age process increases linearly with time when there is no packet delivery, we have

$$\Delta_l(D_i^-) = D_i - D_{i-K_{li}} + Y_{i-K_{li}},\ i > \gamma_l, \tag{C.2.1}$$

where $Y_{i-K_{li}}$ is the service time of packet $i-K_{li}$. Note that $Y_{i-K_{li}}$ is also the age value of source l at time $D_{i-K_{li}}$, i.e., $\Delta_l(D_{i-K_{li}}) = Y_{i-K_{li}}$. Note that $D_i = Y_i + Z_{i-1} + D_{i-1}$.

Repeating this, we can express $(D_i - D_{i-K_{li}})$ *in terms of* Z_i*'s and* Y_i*'s, and hence we get*

$$\Delta_l(D_i^-) = \sum_{k=0}^{K_{li}} Y_{i-k} + \sum_{k=1}^{K_{li}} Z_{i-k},\ i > \gamma_l. \tag{C.2.2}$$

For example, in Fig. 4.2, we have $\Delta_2(D_2^-) = Y_1 + Z_1 + Y_2$.

For $i \leq \gamma_l$, $\Delta_l(D_i^-)$ *is simply the initial age value of source* l $(\Delta_l(0))$ *plus the length of the time interval* $[0, D_i)$. *Hence, we have*

$$\Delta_l(D_i^-) = \Delta_l(0) + D_i,\ i \leq \gamma_l. \tag{C.2.3}$$

Again using $D_i = Y_i + Z_{i-1} + D_{i-1}$ *and the fact that* $D_0 = 0$, *we get*

$$\Delta_l(D_i^-) = \Delta_l(0) + \sum_{k=1}^{i} Y_k + \sum_{k=0}^{i} Z_k,\ i \leq \gamma_l. \tag{C.2.4}$$

In Fig. 4.2, For example, we have $\Delta_1(D_1^-) = \Delta_1(0) + Z_0 + Y_1$.

Substituting (C.2.2) *and* (C.2.4) *into* (4.2.3), *we get*

$$\begin{aligned} \Delta_{avg\text{-}D}(\pi, f) = \limsup_{n\to\infty} \frac{1}{n}\mathbb{E}\Bigg[\sum_{l=1}^{m}\sum_{i=1}^{\gamma_l} g\left(\Delta_l(0) + \sum_{k=1}^{i} Y_k + \sum_{k=0}^{i} Z_k\right) + \\ \sum_{i=\gamma_l}^{n} g\left(\sum_{k=0}^{K_{li}} Y_{i-k} + \sum_{k=1}^{K_{li}} Z_{i-k}\right)\Bigg]. \end{aligned} \tag{C.2.5}$$

Since the function $g(\cdot)$ *is non-decreasing,* (C.2.5) *implies that the Ta-APD is a non-decreasing function of the waiting times. This completes the proof.* □

C.3 Proof of Lemma 4.3.1

Part (i) is proven in two steps:

Step 1: *We will prove that* $\bar{\Delta}_{avg\text{-}opt} \leq \beta$ *if and only if* $\Theta(\beta) \leq 0$*. If* $\bar{\Delta}_{avg\text{-}opt} \leq \beta$*, there exists a sampling policy* $f = (Z_0, Z_1, \ldots) \in \mathcal{F}$ *that is feasible for* (4.3.10) *and* (4.3.11)*, which satisfies*

$$\limsup_{n\to\infty} \frac{\sum_{i=0}^{n-1} \mathbb{E}\left[\sum_{l=1}^{m} \int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau)d\tau\right]}{\sum_{i=0}^{n-1} \mathbb{E}[Z_i + Y_{i+1}]} \leq \beta. \tag{C.3.1}$$

Hence,

$$\limsup_{n\to\infty} \frac{\frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}\left[\sum_{l=1}^{m} \int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau)d\tau - \beta(Z_i+Y_{i+1})\right]}{\frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}[Z_i + Y_{i+1}]} \leq 0. \tag{C.3.2}$$

Since Z_i*'s and* Y_i*'s are bounded and positive and* $\mathbb{E}[Y_i] > 0$ *for all* i*, we have* $0 < \liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}[Z_i + Y_{i+1}] \leq \limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}[Z_i + Y_{i+1}] \leq q$ *for some* $q \in \mathbb{R}^+$*. By this, we get*

$$\limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}\left[\sum_{l=1}^{m} \int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau)d\tau - \beta(Z_i+Y_{i+1})\right] \leq 0. \tag{C.3.3}$$

Therefore, $\Theta(\beta) \leq 0$*.*

In the reverse direction, if $\Theta(\beta) \leq 0$*, then there exists a sampling policy* $f = (Z_0, Z_1, \ldots) \in \mathcal{F}$ *that is feasible for* (4.3.10) *and* (4.3.11)*, which satisfies* (C.3.3)*. Since we have* $0 < \liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}[Z_i + Y_{i+1}] \leq \limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}[Z_i + Y_{i+1}] \leq q$*, we can divide* (C.3.3) *by* $\liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbb{E}[Z_i + Y_{i+1}]$ *to get* (C.3.2)*, which implies* (C.3.1)*. Hence,* $\bar{\Delta}_{avg\text{-}opt} \leq \beta$*. By this, we have proven that* $\bar{\Delta}_{avg\text{-}opt} \leq \beta$ *if and only if* $\Theta(\beta) \leq 0$*.*

Step 2: *We need to prove that* $\bar{\Delta}_{avg\text{-}opt} < \beta$ *if and only if* $\Theta(\beta) < 0$*. This statement can be proven by using the arguments in Step 1, in which "*$\leq$*" should be replaced by "*$<$*". Finally, from the statement of Step 1, it immediately follows that* $\bar{\Delta}_{avg\text{-}opt} > \beta$ *if and only if* $\Theta(\beta) > 0$*. This completes part (i).*

Part(ii): We first show that each optimal solution to (4.3.10) *is an optimal solution to* (4.3.11)*. By the claim of part (i),* $\Theta(\beta) = 0$ *is equivalent to* $\bar{\Delta}_{avg\text{-}opt} = \beta$*. Suppose that policy* $f = (Z_0, Z_1, \ldots) \in \mathcal{F}$ *is an optimal solution to* (4.3.10)*. Then,* $\Delta_{avg(\pi_{MAF}, f)} = \bar{\Delta}_{avg\text{-}opt} = \beta$*. Applying this in the arguments of* (C.3.1)-(C.3.3)*, we can show that policy* f *satisfies*

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{E}\left[\sum_{l=1}^{m} \int_{a_{li}}^{a_{li}+Z_i+Y_{i+1}} g(\tau) d\tau - \beta(Z_i + Y_{i+1}) \right] = 0. \tag{C.3.4}$$

This and $\Theta(\beta) = 0$ *imply that policy* f *is an optimal solution to* (4.3.11)*.*

Similarly, we can prove that each optimal solution to (4.3.11) *is an optimal solution to* (4.3.10)*. By this, part (ii) is proven.* □

C.4 Proof of Proposition 4.3.2

According to [68, Proposition 4.2.1 and Proposition 4.2.6], it is enough to show that for every two states $\mathbf{s}$ *and* $\mathbf{s}'$*, there exists a stationary deterministic policy* f *such that for some* k*, we have*

$$\mathbb{P}\left[\mathbf{s}(k) = \mathbf{s}' | \mathbf{s}(0) = \mathbf{s}, f\right] > 0. \tag{C.4.1}$$

From the state evolution equation (4.3.13)*, we can observe that any state in* $\mathcal{S}$ *can be represented in terms of the waiting and service times. This implies* (C.4.1)*. To clarify this, let us consider a system with 3 sources. Assume that the elements of* *state* $\mathbf{s}'$ *are as follows:*

$$\begin{aligned} a'_{[1]} &= y_3 + z_2 + y_2 + z_1 + y_1, \\ a'_{[2]} &= y_3 + z_2 + y_2, \\ a'_{[3]} &= y_3, \end{aligned} \tag{C.4.2}$$

where y_i*'s and* z_i*'s are any arbitrary elements in* $\mathcal{Y}$ *and* $\mathcal{Z}$*, respectively. Then, we will show that from any arbitrary state* $\mathbf{s} = (a_{[1]}, a_{[2]}, a_{[3]})$*, a sequence of service and waiting times can be followed to reach state* $\mathbf{s}'$*. If we have* $Z_0 = z_1$*,* $Y_1 = y_1$*,* $Z_1 = z_1$*,* $Y_2 = y_2$*,* $Z_2 = z_2$*, and* $Y_3 = y_3$*, then according to* (4.3.13)*, we have in the first stage*

$$\begin{aligned} a_{[1]1} &= a_{[2]} + z_1 + y_1, \\ a_{[2]1} &= a_{[3]} + z_1 + y_1, \\ a_{[3]1} &= y_1, \end{aligned} \tag{C.4.3}$$

and in the second stage, we have

$$\begin{aligned} a_{[1]2} &= a_{[3]} + z_1 + y_2 + z_1 + y_1, \\ a_{[2]2} &= y_2 + z_1 + y_1, \\ a_{[3]2} &= y_2, \end{aligned} \tag{C.4.4}$$

and in the third stage, we have

$$\begin{aligned} a_{[1]3} &= y_3 + z_2 + y_2 + z_1 + y_1 = a'_{[1]}, \\ a_{[2]3} &= y_3 + z_2 + y_2 = a'_{[2]}, \\ a_{[3]3} &= y_3 = a'_{[3]}. \end{aligned} \tag{C.4.5}$$

Hence, a stationary deterministic policy f can be designed to reach state $\mathbf{s}'$ from state $\mathbf{s}$ in 3 stages, if the aforementioned sequence of service times occurs. This implies that

$$\mathbb{P}\left[\mathbf{s}(3) = \mathbf{s}' | \mathbf{s}(0) = \mathbf{s}, f\right] = \prod_{i=1}^{3} \mathbb{P}(Y_i = y_i) > 0, \tag{C.4.6}$$

where we have used that Y_i's are i.i.d.[1] *The previous argument can be generalized to any number of sources. In particular, a forward induction over m can be used to show the result, where* (C.4.1) *trivially holds for $m = 1$, and the previous argument can be used to show that* (C.4.1) *holds for any general m. This completes the proof.* □

C.5 Proof of Proposition 4.3.3

We prove Proposition 4.3.3 into two steps:

***Step 1**: We first show that $h(\mathbf{s})$ is non-decreasing in $\mathbf{s}$. To do so, we show that $J_\alpha(\mathbf{s})$, defined in* (4.3.22)*, is non-decreasing in $\mathbf{s}$, which together with* (4.3.21) *imply that $h(\mathbf{s})$ is non-decreasing in $\mathbf{s}$.*

Given an initial state $\mathbf{s}(0)$, the total expected discounted cost under a sampling policy $f \in \mathcal{F}$ is given by

$$J_\alpha(\mathbf{s}(0); f) = \limsup_{n\to\infty} \mathbb{E}\left[\sum_{i=0}^{n-1} \alpha^i C(\mathbf{s}(i), Z_i)\right], \tag{C.5.1}$$

where $0 < \alpha < 1$ is the discount factor. The optimal total expected α-discounted cost

[1] We assume that all elements in $\mathcal{Y}$ have a strictly positive probability, where the elements with zero probability can be removed without affecting the proof.

function is defined by

$$J_\alpha(\mathbf{s}) = \min_{f \in \mathcal{F}} J_\alpha(\mathbf{s}; f), \ \mathbf{s} \in \mathcal{S}. \tag{C.5.2}$$

A policy is said to be α-optimal if it minimizes the total expected α-discounted cost. The discounted cost optimality equation of $J_\alpha(\mathbf{s})$ is discussed below.

Proposition C.5.1. *The optimal total expected α-discounted cost $J_\alpha(\mathbf{s})$ satisfies*

$$J_\alpha(\mathbf{s}) = \min_{z \in \mathcal{Z}} C(\mathbf{s}, z) + \alpha \sum_{\mathbf{s}' \in \mathcal{S}} \mathbb{P}_{\mathbf{s}\mathbf{s}'}(z) J_\alpha(\mathbf{s}'). \tag{C.5.3}$$

Moreover, a stationary deterministic policy that attains the minimum in equation (C.5.3) *for each $\mathbf{s} \in \mathcal{S}$ will be an α-optimal policy. Also, let $J_{\alpha,0}(\mathbf{s}) = 0$ for all $\mathbf{s}$ and any $n \geq 0$,*

$$J_{\alpha,n+1}(\mathbf{s}) = \min_{z \in \mathcal{Z}} C(\mathbf{s}, z) + \alpha \sum_{\mathbf{s}' \in \mathcal{S}} \mathbb{P}_{\mathbf{s}\mathbf{s}'}(z) J_{\alpha,n}(\mathbf{s}'). \tag{C.5.4}$$

Then, we have $J_{\alpha,n}(\mathbf{s}) \to J_\alpha(\mathbf{s})$ as $n \to \infty$ for every $\mathbf{s}$, and α.

Proof. Since we have bounded cost per stage, the proposition follows directly from [68, Proposition 1.2.2 and Proposition 1.2.3], and [90]. □

Next, we use the optimality equation (C.5.3) *and the value iteration in* (C.5.4) *to prove that $J_\alpha(\mathbf{s})$ is non-decreasing in $\mathbf{s}$.*

Lemma C.5.1. *The optimal total expected α-discounted cost function $J_\alpha(\mathbf{s})$ is non-decreasing in $\mathbf{s}$.*

Proof. We use induction on n in equation (C.5.4) to prove Lemma C.5.1. Obviously, the result holds for $J_{\alpha,0}(\mathbf{s})$.

Now, assume that $J_{\alpha,n}(\mathbf{s})$ is non-decreasing in $\mathbf{s}$. We need to show that for any two states $\mathbf{s}_1$ and $\mathbf{s}_2$ with $\mathbf{s}_1 \leq \mathbf{s}_2$, we have $J_{\alpha,n+1}(\mathbf{s}_1) \leq J_{\alpha,n+1}(\mathbf{s}_2)$. First, we note that, since the age-penalty function $g(\cdot)$ is non-decreasing, the expected cost per stage $C(\mathbf{s}, z)$ is non-decreasing in $\mathbf{s}$, i.e., we have

$$C(\mathbf{s}_1, z) \leq C(\mathbf{s}_2, z). \tag{C.5.5}$$

From the state evolution equation (4.3.13) and the transition probability equation (4.3.15), the second term of the right-hand side (RHS) of (C.5.4) can be rewritten as

$$\sum_{\mathbf{s}' \in \mathcal{S}} \mathbb{P}_{\mathbf{s}\mathbf{s}'}(z) J_{\alpha,n}(\mathbf{s}') = \sum_{y \in \mathcal{Y}} \mathbb{P}(Y = y) J_{\alpha,n}(\mathbf{s}'(z, y)), \tag{C.5.6}$$

where $\mathbf{s}'(z, y)$ is the next state from state $\mathbf{s}$ given the values of z and y. Also, according to the state evolution equation (4.3.13), if the next states of $\mathbf{s}_1$ and $\mathbf{s}_2$ for given values of z and y are $\mathbf{s}'_1(z, y)$ and $\mathbf{s}'_2(z, y)$, respectively, then we have $\mathbf{s}'_1(z, y) \leq \mathbf{s}'_2(z, y)$. This implies that

$$\sum_{y \in \mathcal{Y}} \mathbb{P}(Y = y) J_{\alpha,n}(\mathbf{s}'_1(z, y)) \leq \sum_{y \in \mathcal{Y}} \mathbb{P}(Y = y) J_{\alpha,n}(\mathbf{s}'_2(z, y)), \tag{C.5.7}$$

where we have used the induction assumption that $J_{\alpha,n}(\mathbf{s})$ is non-decreasing in $\mathbf{s}$. Using (C.5.5), (C.5.7), and the fact that the minimum operator in (C.5.4) retains the non-decreasing property, we conclude that

$$J_{\alpha,n+1}(\mathbf{s}_1) \leq J_{\alpha,n+1}(\mathbf{s}_2). \tag{C.5.8}$$

This completes the proof. □

Step 2: *We use Step 1 to prove Proposition 4.3.3. From Step 1, we have that*

$h(\mathbf{s})$ is non-decreasing in $\mathbf{s}$. Similar to Step 1, this implies that the second term of the right-hand side (RHS) of (4.3.19) *($\sum_{\mathbf{s}'\in\mathcal{S}} \mathbb{P}_{\mathbf{ss}'}(z)h(\mathbf{s}')$) is non-decreasing in $\mathbf{s}'$. Moreover, from the state evolution* (4.3.13)*, we can notice that, for any state $\mathbf{s}$, the next state $\mathbf{s}'$ is increasing in z. This argument implies that the second term of the right-hand side (RHS) of* (4.3.19) *($\sum_{\mathbf{s}'\in\mathcal{S}} \mathbb{P}_{\mathbf{ss}'}(z)h(\mathbf{s}')$) is increasing in z. Thus, the value of $z \in \mathcal{Z}$ that achieves the minimum value of this term is zero. If, for a given state $\mathbf{s}$, the value of $z \in \mathcal{Z}$ that achieves the minimum value of the cost function $C(\mathbf{s}, z)$ is zero, then $z = 0$ solves the RHS of* (4.3.19)*. In the sequel, we obtain the condition on $\mathbf{s}$ under which $z = 0$ minimizes the cost function $C(\mathbf{s}, z)$.*

Now, we focus on the cost function $C(\mathbf{s}, z)$. In order to obtain the optimal z that minimizes this cost function, we need to obtain the one-sided derivative of it. The one-sided derivative of a function q in the direction of ω at z is given by

$$\delta q(z;\omega) \triangleq \lim_{\epsilon\to 0^+} \frac{q(z+\epsilon\omega) - q(z)}{\epsilon}. \tag{C.5.9}$$

Let $r(\mathbf{s}, z, Y) = \sum_{l=1}^{m} \int_{a_{[l]}}^{a_{[l]}+z+Y} g(\tau)d\tau$. Since $r(\mathbf{s}, z, Y)$ is the sum of integration of a non-decreasing function $g(\cdot)$, it is easy to show that $r(\mathbf{s}, z, Y)$ is convex. According to [18, Lemma 4], the function $q(z) = \mathbb{E}_Y [r(\mathbf{s}, z, Y)]$ is convex as well. Hence, the one-sided derivative $\delta q(z;\omega)$ of $q(z)$ exists [91, p.709]. Moreover, since $z \to r(\mathbf{s}, z, Y)$ is convex, the function $\epsilon \to [r(\mathbf{s}, z+\epsilon\omega, Y) - r(\mathbf{s}, z, Y)]/\epsilon$ is non-decreasing and bounded from above on $(0, \theta]$ for some $\theta > 0$ [92, Proposition 1.1.2(i)]. Using the monotone convergence theorem [93, Theorem 1.5.6], we can interchange the limit *and integral operators in $\delta q(z;\omega)$ such that*

$$\begin{aligned}
\delta q(z;\omega) &= \lim_{\epsilon\to 0^+} \frac{1}{\epsilon}\mathbb{E}_Y[r(\mathbf{s}, z+\epsilon\omega, Y) - r(\mathbf{s}, z, Y)] \\
&= \mathbb{E}_Y\left[\lim_{\epsilon\to 0^+} \frac{1}{\epsilon}\{r(\mathbf{s}, z+\epsilon\omega, Y) - r(\mathbf{s}, z, Y)\}\right] \\
&= \mathbb{E}_Y\left[\lim_{t\to z^+} \sum_{l=1}^{m} g(a_{[l]} + t + Y) w \mathbb{1}_{\{\omega>0\}} + \right. \\
&\qquad \left. \lim_{t\to z^-} \sum_{l=1}^{m} g(a_{[l]} + t + Y) w \mathbb{1}_{\{\omega<0\}}\right] \\
&= \lim_{t\to z^+} \mathbb{E}_Y\left[\sum_{l=1}^{m} g(a_{[l]} + t + Y) w \mathbb{1}_{\{\omega>0\}}\right] + \\
&\qquad \lim_{t\to z^-} \mathbb{E}_Y\left[\sum_{l=1}^{m} g(a_{[l]} + t + Y) w \mathbb{1}_{\{\omega<0\}}\right],
\end{aligned} \tag{C.5.10}$$

where $\mathbb{1}_E$ is the indicator function of event E. According to [91, p.710] and the convexity of $q(z)$, z is optimal to the cost function $C(\mathbf{s}, z)$ if and only if

$$\delta q(z;\omega) - \bar{\Delta}_{avg\text{-}opt}\omega \geq 0, \ \forall \omega \in \mathbb{R}. \tag{C.5.11}$$

As ω in (C.5.11) *is an arbitrary real number, considering $\omega = 1$,* (C.5.11) *becomes*

$$\lim_{t\to z^+} \mathbb{E}_Y\left[\sum_{l=1}^{m} g(a_{[l]} + t + Y)\right] - \bar{\Delta}_{avg\text{-}opt} \geq 0. \tag{C.5.12}$$

Likewise, considering $\omega = -1$, (C.5.11) *implies*

$$\lim_{t\to z^-} \mathbb{E}_Y\left[\sum_{l=1}^{m} g(a_{[l]} + t + Y)\right] - \bar{\Delta}_{avg\text{-}opt} \leq 0. \tag{C.5.13}$$

Since $g(\cdot)$ *is non-decreasing, we get from* (C.5.11)-(C.5.13) *that* z *must satisfy*

$$\mathbb{E}_Y\left[\sum_{l=1}^{m} g(a_{[l]}+t+Y)\right]-\bar{\Delta}_{avg\text{-}opt}\geq 0,\ \textit{if}\, t>z, \tag{C.5.14}$$

$$\mathbb{E}_Y\left[\sum_{l=1}^{m} g(a_{[l]}+t+Y)\right]-\bar{\Delta}_{avg\text{-}opt}\leq 0,\ \textit{if}\, t<z. \tag{C.5.15}$$

Subsequently, the smallest z *that satisfies* (C.5.14)-(C.5.15) *is*

$$z=\inf\left\{t\geq 0:\mathbb{E}_Y\left[\sum_{l=1}^{m} g(a_{[l]}+t+Y)\right]\geq\bar{\Delta}_{avg\text{-}opt}\right\}. \tag{C.5.16}$$

According to (C.5.16)*, Since* $g(\cdot)$ *is non-decreasing, if* $\mathbb{E}_Y\left[\sum_{l=1}^{m} g(a_{[l]}+Y)\right]\geq\bar{\Delta}_{avg\text{-}opt}$*, then* $z=0$ *minimizes* $C(\mathbf{s},z)$*. This completes the proof.* □

C.6 Proof of Theorem 4.3.3

We use the threshold test $A_s\geq(\bar{\Delta}_{avg\text{-}opt}-m\mathbb{E}[Y])$*, in Proposition 4.3.4, to prove Theorem 4.3.3. We will show that the condition in* (4.3.24) *implies that* $A_s\geq(\bar{\Delta}_{avg\text{-}opt}-m\mathbb{E}[Y])$ *holds for all states* $\mathbf{s}\in\mathcal{S}$*, and hence the zero-wait sampler is optimal under this condition. From the state evolution* (4.3.13)*, we can deduce that for any state* $\mathbf{s}\in\mathcal{S}$*, we have*

$$a_{[l]}\geq(m-l+1)y_{inf},\ \forall l=1,\ldots,m. \tag{C.6.1}$$

This implies

$$A_s\geq\sum_{l=1}^{m} l y_{inf}=\frac{m(m+1)}{2}y_{inf},\ \forall\mathbf{s}\in\mathcal{S}. \tag{C.6.2}$$

Moreover, it is easy to show that the total-average age of the zero-wait sampler, when the scheduling policy is fixed to the MAF scheduler, is given by

$$\bar{\Delta}_0 = \frac{\frac{m(m+1)}{2}\mathbb{E}[Y]^2 + \frac{m}{2}\mathbb{E}[Y^2]}{\mathbb{E}[Y]}. \tag{C.6.3}$$

Since $\bar{\Delta}_0 \geq \bar{\Delta}_{avg\text{-}opt}$, *we have*

$$\bar{\Delta}_0 - m\mathbb{E}[Y] \geq \bar{\Delta}_{avg\text{-}opt} - m\mathbb{E}[Y]. \tag{C.6.4}$$

Hence, if the following condition holds

$$\frac{m(m+1)}{2} y_{inf} \geq \frac{\frac{m(m+1)}{2}\mathbb{E}[Y]^2 + \frac{m}{2}\mathbb{E}[Y^2]}{\mathbb{E}[Y]} - m\mathbb{E}[Y], \tag{C.6.5}$$

which is equivalent to

$$y_{inf} \geq \frac{(m-1)\mathbb{E}[Y]^2 + \mathbb{E}[Y^2]}{(m+1)\mathbb{E}[Y]}, \tag{C.6.6}$$

then we have $A_s \geq (\bar{\Delta}_{avg\text{-}opt} - m\mathbb{E}[Y])$ *for all states* $\mathbf{s} \in \mathcal{S}$. *This implies that the zero-wait sampler is optimal under this condition. This completes the proof.* □

APPENDIX D: PROOFS FOR CHAPTER 5

D.1 Derivation of (5.2.5)

Define S_l as the residual sleeping period of source l after a sleep-wake cycle is over. Due to the memoryless property of exponential distribution, since the sleeping period of source l is exponentially distributed with mean value $\mathbb{E}[T]/r_l$, S_l is also exponentially distributed with mean value $\mathbb{E}[T]/r_l$. According to the proposed sleep-wake scheduler, source l gains access to the channel and transmits successfully in a given cycle if $S_i \geq S_l + t_s$ for all $i \neq l$. Hence, we have

$$\alpha_l = \mathbb{P}(S_i \geq S_l + t_s,\ \forall i \neq l) \tag{D.1.1}$$

$$\overset{(a)}{=} \mathbb{E}[\mathbb{P}(S_i \geq S_l + t_s,\ \forall i \neq l | S_l)] \tag{D.1.2}$$

$$\overset{(b)}{=} \mathbb{E}\left[\prod_{i \neq l} \mathbb{P}(S_i \geq S_l + t_s | S_l)\right] \tag{D.1.3}$$

$$= \int_0^\infty \left[\prod_{i \neq l} e^{-r_i \frac{s_l + t_s}{\mathbb{E}[T]}}\right] \frac{r_l}{\mathbb{E}[T]} e^{-r_l \frac{s_l}{\mathbb{E}[T]}} ds_l \tag{D.1.4}$$

$$= \frac{r_l e^{r_l \frac{t_s}{\mathbb{E}[T]}}}{e^{\sum_{i=1}^M r_i \frac{t_s}{\mathbb{E}[T]}} \sum_{i=1}^M r_i}, \tag{D.1.5}$$

where (a) is due to $\mathbb{P}[A] = \mathbb{E}[\mathbb{P}(A|B)]$, and (b) is due to the fact that S_l is independent for different sources. □

D.2 Derivation of (5.2.13)

Recall the definition of S_l at the beginning of Appendix D.1. Moreover, define P_l as the probability that source l transmits a packet in a given cycle, regardless whether packet collision occurs or not. For the sleep-wake scheduling mechanism the we are utilizing here, source l transmits in a given cycle as long as no other source wakes up before $S_l - t_s$, i.e., $S_i \geq S_l - t_s$ for all $i \neq l$. Hence, we have

$$P_l = \mathbb{P}(S_i \geq S_l - t_s, \ \forall i \neq l) \tag{D.2.1}$$

$$= \mathbb{P}(S_i \geq S_l - t_s, \ \forall i \neq l, \ S_l \geq t_s) + \mathbb{P}(S_l < t_s), \tag{D.2.2}$$

where the first term in the RHS is given by

$$\mathbb{P}(S_i \geq S_l - t_s \geq 0, \ \forall i \neq l) \tag{D.2.3}$$

$$=\mathbb{E}[\mathbb{P}(S_i \geq S_l - t_s \geq 0, \ \forall i \neq l | S_l)] \tag{D.2.4}$$

$$=\mathbb{E}\left[\prod_{i \neq l} \mathbb{P}(S_i \geq S_l - t_s \geq 0 | S_l))\right] \tag{D.2.5}$$

$$= \int_{t_s}^{\infty} \left[\prod_{i \neq l} e^{-r_i \frac{s_l - t_s}{\mathbb{E}[T]}}\right] \frac{r_l}{\mathbb{E}[T]} e^{-r_l \frac{s_l}{\mathbb{E}[T]}} ds_l \tag{D.2.6}$$

$$=e^{-r_l \frac{t_s}{\mathbb{E}[T]}} \frac{r_l}{\sum_{i=1}^{M} r_i}. \tag{D.2.7}$$

Since S_l is exponentially distributed with mean value $\mathbb{E}[T]/r_l$, we can determine the second term in the RHS of (D.2.2) *as follows:*

$$\mathbb{P}(S_l < t_s) = 1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}. \tag{D.2.8}$$

Substituting (D.2.7) *and* (D.2.8) *back into* (D.2.2), *we get*

$$P_l = 1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}} + e^{-r_l \frac{t_s}{\mathbb{E}[T]}} \frac{r_l}{\sum_{i=1}^{M} r_i}. \tag{D.2.9}$$

Let α_{col} *denote the collision probability in a given cycle. We have* $\alpha_{col} = 1 - \sum_{i=1}^{M} \alpha_i$, *because each cycle includes either a successful transmission or a collision. Moreover, let* $\mathbb{E}[\mathbf{Idle}]$ *denote the mean of the idle duration in a cycle. By the renewal theory in stochastic processes [94],* σ_l *is given by*

$$\sigma_l = \frac{P_l \mathbb{E}[T]}{(\sum_{i=1}^{M} \alpha_i + \alpha_{col})\mathbb{E}[T] + \mathbb{E}[\mathbf{Idle}]} \tag{D.2.10}$$

$$= \frac{P_l \mathbb{E}[T]}{\mathbb{E}[T] + \frac{\mathbb{E}[T]}{\sum_{i=1}^{M} r_i}} \tag{D.2.11}$$

$$= \frac{[1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}] \sum_{i=1}^{M} r_i + r_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}}}{\sum_{i=1}^{M} r_i + 1}. \tag{D.2.12}$$

□

D.3 Proof of Lemma 5.4.1

First of all, we need to show that (5.3.3) *has a solution for* $\beta^\star$.

Lemma D.3.1. *Suppose that* $w_l > 0$*, and* $b_l > 0$ *for all* l*. If* $\sum_{i=1}^{M} b_i \geq 1$*, then* (5.3.3) *has a unique solution on* $[0, \max_l (b_l/\sqrt{w_l})]$*; otherwise,* (5.3.3) *has no solution.*

Proof. It is clear that if $\sum_{i=1}^{M} b_i = 1$, then $\beta^\star$ satisfies (5.3.3) if and only if $\beta^\star \geq \max_l (b_l/\sqrt{w_l})$. Hence, (5.3.3) has a unique solution on $[0, \max_l (b_l/\sqrt{w_l})]$ in this case. We now focus on the case of $\sum_{i=1}^{M} b_i > 1$. In this case, we have the following:

- If $\beta^\star = 0$, then $\sum_{i=1}^{M} \min\{b_i, \beta^\star \sqrt{w_i}\} = 0$.
- If $\beta^\star = \max_l (b_l / \sqrt{w_l})$, then $\sum_{i=1}^{M} \min\{b_i, \beta^\star \sqrt{w_i}\} > 1$.
- The left hand side (LHS) of (5.3.3) is strictly increasing and continuous in $\beta^\star$ on $[0, \max_l (b_l / \sqrt{w_l})]$.

As a result, (5.3.3) has a unique solution on $[0, \max_l (b_l / \sqrt{w_l})]$ in this case as well. Finally, if $\sum_{i=1}^{M} b_i < 1$, then $\sum_{i=1}^{M} \min\{b_i, \beta^\star \sqrt{w_i}\} \leq \sum_{i=1}^{M} b_i < 1$. Hence, (5.3.3) has no solution if $\sum_{i=1}^{M} b_i < 1$. This completes the proof. □

Since we have $\sum_{i=1}^{M} b_i \geq 1$, *Lemma D.3.1 implies that* (5.3.3) *has a solution for* $\beta^\star$. *Now, we are ready to prove Lemma 5.4.1. Consider the following constraints:*

$$\frac{r_l \frac{t_s}{\mathbb{E}[T]} \sum_{i=1}^{M} r_i + r_l}{\sum_{i=1}^{M} r_i + 1} \leq b_l, \forall l. \tag{D.3.1}$$

Since we have

$$1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}} \leq r_l \frac{t_s}{\mathbb{E}[T]}, \tag{D.3.2}$$

$$e^{-r_l \frac{t_s}{\mathbb{E}[T]}} \leq 1, \tag{D.3.3}$$

then,

$$[1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}] \sum_{i=1}^{M} r_i + r_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}} \leq r_l \frac{t_s}{\mathbb{E}[T]} \sum_{i=1}^{M} r_i + r_l. \tag{D.3.4}$$

Thus, if the constraints in (D.3.1) *are satisfied for a given solution* $\mathbf{r}$*, then the constraints of Problem **1** are satisfied as well. We can observe that the constraints in* (D.3.1) *are equivalent to the following set of constraints:*

$$\begin{aligned} & r_l \leq b_l \frac{x+1}{1+\frac{t_s}{\mathbb{E}[T]}x}, \forall l \\ & \sum_{i=1}^{M} r_i = x. \end{aligned} \tag{D.3.5}$$

Now, it is easy to show that if $x \leq \sqrt{\mathbb{E}[T]/t_s}$*, then* $x \leq (x+1)/[1+(t_s/\mathbb{E}[T])x]$*. Meanwhile, our proposed solution* $\mathbf{r}^\star$ (5.3.1) - (5.3.3) *satisfies* $\sum_{i=1}^{M} r_i^\star = x^\star$*. Thus, if we can show that* $x^\star \leq \sqrt{\mathbb{E}[T]/t_s}$*, then*

$$r_l^\star = \min\{b_l, \beta^\star \sqrt{w_l}\} x^\star \leq b_l x^\star \leq b_l \frac{x^\star+1}{1+\frac{t_s}{\mathbb{E}[T]}x^\star}, \tag{D.3.6}$$

and the constraints in (D.3.5) *hold for our proposed solution* $\mathbf{r}^\star$*. What remains is to prove that* $x^\star \leq \sqrt{\mathbb{E}[T]/t_s}$*. We have*

$$x^\star = \frac{-1}{2} + \sqrt{\frac{1}{4} + \frac{\mathbb{E}[T]}{t_s}} \tag{D.3.7}$$

$$= \frac{\frac{\mathbb{E}[T]}{t_s}}{\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{\mathbb{E}[T]}{t_s}}} \tag{D.3.8}$$

$$\leq \frac{\frac{\mathbb{E}[T]}{t_s}}{\sqrt{\frac{\mathbb{E}[T]}{t_s}}} = \sqrt{\frac{\mathbb{E}[T]}{t_s}}. \tag{D.3.9}$$

Hence, our proposed solution $\mathbf{r}^\star$ (5.3.1) - (5.3.3) *satisfies* (D.3.5)*, which implies* (D.3.1)*. This completes the proof.* □

D.4 Proof of Lemma 5.4.3

By replacing $e^{-r_l(t_s/\mathbb{E}[T])}e^{\sum_{i=1}^{M} r_i(t_s/\mathbb{E}[T])}$ *in* (5.4.4) *of Problem **2** by 1, we obtain the following optimization problem:*

$$\min_{r_l>0} \sum_{l=1}^{M} \frac{w_l}{r_l}\left(1+\sum_{i=1}^{M} r_i\right)+\sum_{l=1}^{M} w_l \tag{D.4.1}$$

$$\textbf{\textit{s.t.}}\ r_l \le b_l\left(\sum_{i=1}^{M} r_i + 1\right), \forall l. \tag{D.4.2}$$

Since $e^{-r_l(t_s/\mathbb{E}[T])}e^{\sum_{i=1}^{M} r_i(t_s/\mathbb{E}[T])} \ge 1$, *Problem* (D.4.1) *serves as a lower bound of Problem **2**, and hence a lower bound of Problem **1** as well. Define an auxiliary variable* $y = \sum_{i=1}^{M} r_i + 1$. *By this, we solve a two-layer nested optimization problem. In the inner layer, we optimize* $\mathbf{r}$ *for a given* y. *After solving* $\mathbf{r}$, *we will optimize* y *in the outer layer. Now, fix the value of* y, *we obtain the following optimization problem (the inner layer):*

$$\min_{r_i>0} \sum_{i=1}^{M} \left[\frac{w_i y}{r_i} + w_i\right] \tag{D.4.3}$$

$$\textbf{\textit{s.t.}}\ r_l \le b_l y, \forall l, \tag{D.4.4}$$

$$\sum_{i=1}^{M} r_i + 1 = y. \tag{D.4.5}$$

The objective function in (D.4.3) *is a convex function. Moreover, the constraints in* (D.4.4) *and* (D.4.5) *are affine. Hence, Problem* (D.4.3) *is convex. We use the Lagrangian duality approach to solve Problem* (D.4.3). *Problem* (D.4.3) *satisfies Slater's conditions. Thus, the Karush-Kuhn-Tucker (KKT) conditions are both necessary and sufficient for optimality [95]. Let* $\gamma = (\gamma_1, \ldots, \gamma_M)$ *and* μ *be the Lagrange multipliers associated with constraints* (D.4.4) *and* (D.4.5), *respectively. Then, the Lagrangian* *of Problem* (D.4.3) *is given by*

$$L(\mathbf{r}, \gamma, \mu) = \sum_{i=1}^{M} \left[\frac{w_i y}{r_i} + w_i \right] + \sum_{i=1}^{M} \gamma_i (r_i - b_i y) + \mu \left(\sum_{i=1}^{M} r_i + 1 - y \right). \tag{D.4.6}$$

Take the derivative of (D.4.6) *with respect to* r_l *and set it equal to 0, we get*

$$\frac{-w_l y}{r_l^2} + \gamma_l + \mu = 0. \tag{D.4.7}$$

This and KKT conditions imply

$$r_l = \sqrt{\frac{w_l y}{\gamma_l + \mu}}, \tag{D.4.8}$$

$$\gamma_l \geq 0, r_l - b_l y \leq 0, \tag{D.4.9}$$

$$\gamma_l (r_l - b_l y) = 0, \tag{D.4.10}$$

$$\sum_{i=1}^{M} r_i + 1 = y. \tag{D.4.11}$$

If $\gamma_l = 0$*, then* $r_l = \sqrt{(w_l y)/\mu}$ *and* $r_l \leq b_l y$*; otherwise, if* $\gamma_l > 0$*, then* $r_l = b_l y$ *and* $r_l < \sqrt{(w_l y)/\mu}$*. Hence, we have*

$$r_l = \min \left\{ b_l y, \sqrt{\frac{w_l y}{\mu^\star}} \right\}, \tag{D.4.12}$$

where by (D.4.5)*,* $\mu^\star$ *satisfies*

$$\sum_{i=1}^{M} \min \left\{ b_i y, \sqrt{\frac{w_i y}{\mu^\star}} \right\} + 1 = y. \tag{D.4.13}$$

We can observe that $\mu^\star$ *is a function of* y*. Because of that, we can define* $\beta^\star(y) = \sqrt{1/(y\mu^\star)}$*, which is a function of* y *as well. Then, the optimum solution to* (D.4.3) *can be rewritten as*

$$r_l = \min\{b_l, \beta^\star(y)\sqrt{w_l}\}y, \forall l, \tag{D.4.14}$$

where $\beta^\star(y)$ *satisfies*

$$\sum_{i=1}^{M} \min\{b_i, \beta^\star(y)\sqrt{w_i}\} + \frac{1}{y} = 1. \tag{D.4.15}$$

Substituting (D.4.14) *and* (D.4.15) *back in Problem* (D.4.3)*, we get the following optimization problem (the outer layer):*

$$\min_{y>1} \sum_{i=1}^{M} \left[\frac{w_i}{\min\{b_i, \beta^\star(y)\sqrt{w_i}\}} + w_i\right] \tag{D.4.16}$$

$$\textbf{\textit{s.t.}} \sum_{i=1}^{M} \min\{b_i, \beta^\star(y)\sqrt{w_i}\} + \frac{1}{y} = 1. \tag{D.4.17}$$

Problem (D.4.16) *serves as a lower bound of Problem* ***2****, and hence a lower bound of Problem* ***1****. We can observe that the objective function in* (D.4.16) *is decreasing in* $\beta^\star(y)$*. Moreover,* (D.4.17) *implies that* $\beta^\star(y)$ *is strictly increasing in* y *if* $\sum_{i=1}^{M} b_i \geq 1$*. As a result,* $y = \infty$ *is the optimal solution of Problem* (D.4.16)*. At the limit, the constraint* (D.4.17) *converges to* (5.3.3)*. Since* $\beta^\star$ *serves as a solution for* (5.3.3)*, we can deduce that* $\lim_{y\to\infty} \beta^\star(y) = \beta^\star$*. Thus, we have the following lower bound:*

$$\bar{\Delta}_{opt}^{\textit{w-peak}} \geq \bar{\Delta}_{opt,2}^{\textit{w-peak}} \geq \sum_{i=1}^{M} \left[\frac{w_i}{\min\{b_i, \beta^\star\sqrt{w_i}\}} + w_i\right]. \tag{D.4.18}$$

This completes the proof. □

D.5 Proof of Lemma 5.4.4

Because $1 - e^{-x} \leq x$*, we can obtain*

$$\begin{aligned} & r_l e^{-r_l \frac{t_s}{\mathbb{E}[T]}} + [1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}] \sum_{i=1}^{M} r_i \\ & = r_l + [1 - e^{-r_l \frac{t_s}{\mathbb{E}[T]}}] \left(\sum_{i=1}^{M} r_i - r_l \right) \\ & \leq r_l + r_l \frac{t_s}{\mathbb{E}[T]} \left(\sum_{i=1}^{m} r_i - r_l \right), \end{aligned} \tag{D.5.1}$$

Hence, if $\mathbf{r}$ *satisfies the constraint*

$$\frac{r_l + r_l \frac{t_s}{\mathbb{E}[T]} \left(\sum_{i=1}^{M} r_i - r_l \right)}{\sum_{i=1}^{M} r_i + 1} \leq b_l, \tag{D.5.2}$$

then $\mathbf{r}$ *also satisfies the constraint of Problem* ***1*** *in* (5.2.17)*. Consider the following set of solution indexed by a parameter* $c > 0$*:*

$$r_l = c u_l, \ \forall l, \tag{D.5.3}$$

$$u_l = \frac{b_l}{1 - \sum_{i=1}^{M} b_i}, \ \forall l \tag{D.5.4}$$

We want to find a c *such that the solution in* (D.5.3) *and* (D.5.4) *is feasible for Problem* ***1****. To achieve this, we first substitute the solution* (D.5.3) *and* (D.5.4) *into the constraint* (D.5.2)*, and get*

$$\frac{c u_l + c^2 u_l \frac{t_s}{\mathbb{E}[T]} \left(\sum_{i=1}^{M} u_i - u_l \right)}{c \sum_{i=1}^{M} u_i + 1} \leq b_l. \tag{D.5.5}$$

If equality is satisfied in (D.5.5)*, we can obtain the following quadratic equation for c:*

$$c^2\left[u_l\frac{t_s}{\mathbb{E}[T]}\left(\sum_{i=1}^{M}u_i - u_l\right)\right] + c\left(u_l - b_l\sum_{i=1}^{M}u_i\right) - b_l = 0. \tag{D.5.6}$$

The solution to (D.5.6) *is given by* c_l *in* (5.3.9)*. Hence,* $r_l = c_l u_l$ *is feasible for the constraint* (D.5.2) *for source* l*.*

As feasibility for one source only is insufficient, we further prove that the solution in (D.5.3) *and* (D.5.4) *with* $c = \min_l c_l$ *is feasible for satisfying the energy constraints of all sources* $l = 1, \ldots, M$*. To that end, let us consider the monotonicity of the LHS of* (D.5.5)*. By taking the derivative with respect to c, we get*

$$\frac{u_l\frac{t_s}{\mathbb{E}[T]}\left(\sum_{i=1}^{M}u_i - u_l\right)\left(c^2\sum_{i=1}^{M}u_i + 2c\right) + u_l}{(c\sum_{i=1}^{M}u_i + 1)^2} > 0. \tag{D.5.7}$$

Hence,

$$r_l = \left(\min_l c_l\right)u_l,\ \forall l, \tag{D.5.8}$$

is feasible for the energy constraints of all sources $l = 1, \ldots, M$*. After some manipulations, the solution in* (D.5.4) *and* (D.5.8) *are equivalently expressed as* (5.3.1) *and* (5.3.8) - (5.3.10)*. This completes the proof.* □

D.6 Proof of Lemma 5.4.6

By replacing $e^{-r_l(t_s/\mathbb{E}[T])}/r_l$ *by* $e^{-\sum_{i=1}^{M} r_i(t_s/\mathbb{E}[T])}/[\ b_l(\sum_{i=1}^{M} r_i+1)]$ *and* $e^{\sum_{i=1}^{M} r_i(t_s/\mathbb{E}[T])}$ *by 1 in* (5.4.4) *of Problem* ***2****, we obtain the following optimization problem:*

$$\begin{aligned}&\min_{r_l>0} \sum_{l=1}^{M} \frac{w_l e^{-\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}}}{b_i} + \sum_{l=1}^{M} w_l \\ &\boldsymbol{s.t.}\ r_l \leq b_l \left(\sum_{i=1}^{M} r_i + 1 \right), \forall l.\end{aligned} \tag{D.6.1}$$

Since $r_l \leq b_l(\sum_{i=1}^{M} r_i\ +1)$*, we have*

$$\frac{e^{-r_l \frac{t_s}{\mathbb{E}[T]}}}{r_l} \geq \frac{e^{-\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}}}{b_l \left(\sum_{i=1}^{M} r_i + 1 \right)}. \tag{D.6.2}$$

Moreover, we have $e^{\sum_{i=1}^{M} r_i(t_s/\mathbb{E}[T])} \geq 1$*. Thus, Problem* (D.6.1) *serves as a lower bound of Problem* ***2****, and hence a lower bound of Problem* ***1*** *as well. By removing the constant term* $\sum_{l=1}^{M} w_l$ *in the objective function of Problem* (D.6.1) *and then taking the logarithm, Problem* (D.6.1) *is reformulated as*

$$\begin{aligned}&\min_{r_i>0} \log \left(\sum_{i=1}^{M} \frac{w_i}{b_i} \right) - \sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]} \\ &\boldsymbol{s.t.}\ r_l \leq b_l \left(\sum_{i=1}^{M} r_i + 1 \right), \forall l.\end{aligned} \tag{D.6.3}$$

Obviously, Problem (D.6.3) *is a convex optimization problem and satisfies Slater's conditions. Thus, the KKT conditions are are necessary and sufficient for optimality. Let* $\tau = (\tau_1, \ldots, \tau_M)$ *be the Lagrange multipliers associated with the constraints of*

Problem (D.6.3). *Then, the Lagrangian of Problem* (D.6.3) *is given by*

$$\begin{aligned} L(\mathbf{r}, \tau) = & \log\left(\sum_{i=1}^{M} \frac{w_i}{b_i}\right) - \left(\sum_{i=1}^{M} r_i \frac{t_s}{\mathbb{E}[T]}\right) \\ & + \sum_{i=1}^{M} \tau_i \left[r_i - b_i \left(\sum_{i=1}^{M} r_i + 1\right)\right]. \end{aligned} \tag{D.6.4}$$

Take the derivative of (D.6.4) *with respect to* r_l *and set it equal to 0, we get*

$$\frac{-t_s}{\mathbb{E}[T]} + \tau_l(1 - b_l) - \sum_{i \neq l} \tau_i b_i = 0. \tag{D.6.5}$$

This and KKT conditions imply

$$\tau_l = \frac{t_s}{\mathbb{E}[T](1 - b_l)} + \frac{\sum_{i \neq l} \tau_i b_i}{1 - b_l}, \tag{D.6.6}$$

$$\tau_l \geq 0, r_l - b_l \left(\sum_{i=1}^{M} r_i + 1\right) \leq 0, \tag{D.6.7}$$

$$\tau_l \left[r_l - b_l \left(\sum_{i=1}^{M} r_i + 1\right)\right] = 0. \tag{D.6.8}$$

Since $\sum_{i=1}^{M} b_i < 1$, (D.6.6) *implies that* $\tau_l > 0$ *for all* l. *This and* (D.6.8) *result in*

$$r_l = b_l \left(\sum_{i=1}^{M} r_i + 1\right), \forall l. \tag{D.6.9}$$

Because $\sum_{i=1}^{M} b_i < 1$, (D.6.9) *has a unique solution, which is given by*

$$r_l = \frac{b_l}{1 - \sum_{i=1}^{M} b_i}, \forall l. \tag{D.6.10}$$

Hence, the solution to (D.6.1) *and* (D.6.3) *is given by* (D.6.10)*. Substitute* (D.6.10) *into* (D.6.1)*, we get the following lower bound:*

$$\bar{\Delta}_{opt}^{w\text{-}peak} \geq \bar{\Delta}_{opt,2}^{w\text{-}peak} \geq \sum_{l=1}^{M} \frac{w_l e^{\frac{-\sum_{i=1}^{M} b_i}{1-\sum_{i=1}^{M} b_i} \frac{t_s}{\mathbb{E}[T]}}}{b_l} + \sum_{l=1}^{M} w_l. \tag{D.6.11}$$

This completes the proof. □

D.7 Proof of Corollary 5.3.2.2

We start by solving Problem (5.3.21) *for optimal* **a***. Problem* (5.3.21) *is a convex optimization problem and satisfies Slater's conditions. Thus, the KKT conditions are necessary and sufficient for optimality. Let* $\lambda = (\lambda_1, \ldots, \lambda_M)$ *and* ν *be the Lagrange multipliers associated with the constraints* (5.3.22) *and* (5.3.23)*, respectively. Then, the Lagrangian of Problem* (5.3.21) *is given by*

$$\begin{aligned} L(\mathbf{a}, \lambda, \nu) = &\sum_{i=1}^{M} \left[\frac{w_i}{a_i} + w_i \right] \\ &+ \sum_{i=1}^{M} \lambda_i (a_i - b_i) + \nu \left(\sum_{i=1}^{M} a_i - 1 \right). \end{aligned} \tag{D.7.1}$$

Take the derivative of (D.7.1) *with respect to* a_l *and set it equal to 0, we get*

$$\frac{-w_l}{a_l^2} + \lambda_l + \nu = 0. \tag{D.7.2}$$

This and KKT conditions imply

$$a_l = \sqrt{\frac{w_l}{\lambda_l + \nu}}, \tag{D.7.3}$$

$$\lambda_l \geq 0,\ a_l - b_l \leq 0, \tag{D.7.4}$$

$$\lambda_l(a_l - b_l) = 0, \tag{D.7.5}$$

$$\nu \geq 0,\ \sum_{i=1}^{M} a_i - 1 \leq 0, \tag{D.7.6}$$

$$\nu \left(\sum_{i=1}^{M} a_i - 1 \right) = 0. \tag{D.7.7}$$

If $\lambda_l = 0$*, then we have* $a_l = \sqrt{w_l/\nu}$ *and* $a_l \leq b_l$*. This implies that* $\nu > 0$ *and hence* $\sum_{i=1}^{M} a_i = 1$*, which holds when* $\sum_{i=1}^{M} b_i \geq 1$*.*

If $\lambda_l > 0$*, then we have* $a_l = b_l$ *and* $a_l \leq \sqrt{w_l/\nu}$*. In this case, we either have* $\nu > 0$*, which implies* $\sum_{i=1}^{M} a_i = 1$ *and this holds when* $\sum_{i=1}^{M} b_i \geq 1$*; or* $\nu = 0$*, which implies* $\sum_{i=1}^{M} a_i \leq 1$ *and this holds when* $\sum_{i=1}^{M} b_i \leq 1$*.*

From the above argument, the solution can be driven according to the following two cases:

*Case 1 (**Energy-adequate regime** (*$\sum_{i=1}^{M} b_i \geq 1$*)): In this case, the optimal solution is given by*

$$a_l^\star = \min\left\{ b_l, \sqrt{\frac{w_l}{\nu^\star}} \right\},\ \forall l, \tag{D.7.8}$$

where we must have $\nu^\star > 0$*, which implies* $\sum_{i=1}^{M} a_i^\star = 1$*. Hence,* $\nu^\star$ *satisfies*

$$\sum_{i=1}^{M} \min\left\{ b_i, \sqrt{\frac{w_i}{\nu^\star}} \right\} = 1. \tag{D.7.9}$$

By comparing (D.7.9) *with* (5.3.3), *we can deduce that* $\sqrt{1/\nu^\star} = \beta^\star$, *where* $\beta^\star$ *satisfies*

$$\sum_{i=1}^{M} \min\{b_i, \beta^\star \sqrt{w_i}\} = 1. \tag{D.7.10}$$

Since $\sum_{i=1}^{M} b_i \geq 1$, (D.7.10) *has a solution for* $\beta^\star$ *as shown in Lemma D.3.1. Hence, the solution to Problem* (5.3.21) *can be rewritten as*

$$a_l^\star = \min\{b_l, \beta^\star \sqrt{w_l}\}, \ \forall l. \tag{D.7.11}$$

Substituting (D.7.11) *into* (5.3.21), *we obtain*

$$\bar{\Delta}_{\textit{opt-s}}^{\textit{w-peak}} = \sum_{i=1}^{M} \left[\frac{w_i}{\min\{b_i, \beta^\star \sqrt{w_i}\}} + w_i \right], \tag{D.7.12}$$

which is equal to the asymptotic optimal objective value of Problem ***1*** *in energy-adequate regime in* (5.3.7).

*Case 2 (**Energy-scarce regime (**$\sum_{i=1}^{M} b_i < 1$**)**): In this case, the optimal solution is*

$$a_l^\star = b_l, \ \forall l. \tag{D.7.13}$$

Substituting by this into (5.3.21), *we obtain*

$$\bar{\Delta}_{\textit{opt-s}}^{\textit{w-peak}} = \sum_{i=1}^{M} \left[\frac{w_i}{b_i} + w_i \right], \tag{D.7.14}$$

which is equal to the asymptotic optimal objective value of Problem ***1*** *in energy-scarce regime in* (5.3.13). *This completes the proof.* □