# Age-of-Information in Wireless Networks: Theory and Implementation

by

Igor Kadota

BSc Electrical Engineering, ITA, Brazil, 2010
SM Telecommunications, ITA, Brazil, 2013
SM Communication Networks, MIT, USA, 2016

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2020

© 2020 Massachusetts Institute of Technology. All rights reserved.

Author . . . . . .
Department of Aeronautics and Astronautics
August 12, 2020

Certified by . . . . . .
Eytan Modiano
Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by . . . . . .
Zoltan Spakovszky
Professor, Department of Aeronautics and Astronautics
Chair, Graduate Program Committee

# Age-of-Information in Wireless Networks: Theory and Implementation

by Igor Kadota

Submitted to the Department of Aeronautics and Astronautics on August 12, 2020,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

## Abstract

Emerging data-driven applications will increasingly rely on sharing time-sensitive information for monitoring and control. Examples are abundant: mobile robots in automated warehouses sharing status information to cooperate with each other and with humans, self-driving cars exchanging safety-related information with other vehicles and infrastructure, and smart-cities analyzing data from Internet-of-Things (IoT) sensors to provide real-time feedback for vehicles and traffic management systems. In such application domains, it is essential to keep state information fresh, as outdated information loses its value and can lead to system failures and safety risks. The Age of Information (AoI) is a recently proposed performance metric that captures the freshness of information from the perspective of the destination. Optimizing AoI is a challenging objective that goes beyond low latency, it requires that packets with low delay are delivered regularly over time to every destination in the network. In this thesis, we use rigorous theory to gain insight into the AoI optimization problem and to develop practical network control mechanisms, and we leverage system implementation to evaluate the performance of these mechanisms in real operating scenarios.

We consider a broadcast single-hop wireless network with a base station and a number of nodes sharing time-sensitive information through unreliable communication links. We formulate a discrete-time decision problem and use tools from mathematical optimization and stochastic control to develop network control mechanisms that optimize AoI. Our first approach is to develop an algorithm that computes the optimal transmission scheduling decision at every time t. As expected, this optimal solution is impractical due to its high computational complexity - shown to grow exponentially with the size of the network. To overcome this challenge, we propose low-complexity transmission scheduling policies with provable performance guarantees in terms of AoI. For example, we use Lyapunov Optimization to develop an AoI-based Max-Weight policy, show that this policy is optimal for symmetric networks, and show that, for general networks, this policy is guaranteed to be within a factor of two away from the optimal AoI. Numerical results suggest that this Max-Weight policy achieves near-optimal performance in various network settings. Throughout the thesis, we analyze, optimize, and evaluate important classes of centralized and dis-

tributed low-complexity transmission scheduling algorithms, namely Max-Weight, Maximum Age First, Stationary Randomized, Whittle's Index, Slotted-ALOHA and Carrier-Sense Multiple Access, using tools from Dynamic Programming, Lyapunov Optimization, Renewal Theory and the Restless Multi-Armed Bandits framework.

Leveraging the theoretical results, we propose WiFresh: an unconventional network architecture that scales gracefully, achieving near optimal information freshness in wireless networks of any size, even when the network is overloaded. We propose and realize two strategies for implementing WiFresh: one at the MAC layer in a network of FPGA-enabled Software Defined Radios using hardware-level programming, and another at the Application layer, without modifications to lower layers of the communication system, in a network of Raspberry Pis using Python 3. Our experimental results show that the more congested the network, the more prominent is the superiority of WiFresh when compared to an equivalent WiFi network, with WiFresh achieving two orders of magnitude improvement over standard WiFi. Our measurements suggest that WiFresh is well-suited for large-scale applications that rely on sharing time-sensitive information.

---

Thesis Supervisor: Eytan Modiano
Title: Professor, Department of Aeronautics and Astronautics

In memory of Professor Alessandro Anzaloni.

THIS PAGE INTENTIONALLY LEFT BLANK

# Acknowledgments

I would like to sincerely thank my PhD adviser, Prof. Eytan Modiano, for making this amazing journey possible. Eytan is an incredible mentor. He taught me how to write properly, how to present, how to do research, how to teach, and how to mentor students. I would like to thank Eytan for the time and effort, they are very much appreciated.

I am thankful to my SM adviser in ITA, Prof. Alessandro Anzaloni, for inspiring me to do research on networks. I learned to love networks by watching Alessandro's lectures, and my first research experience was during an internship with Alessandro and Prof. Andrea Baiocchi in Italy. After this internship, my goal became to do a PhD in communication networks. I sincerely thank Alessandro for teaching me about networks and for always believing in me.

I would like to thank my PhD thesis committee: Professors Mor Harchol-Balter, Mohammad Alizadeh, Moe Win, and Yin Sun. I sincerely appreciate the thoughtful research advise and the time you invested into improving this thesis.

I am very grateful towards the people in LIDS, AeroAstro, Westgate, Ashdown, and MIT for fostering a fun and creative work environment. I am thankful for sharing this journey with so many good friends. I am thankful for collaborating with so many bright people. I am grateful towards staff members at LIDS and AeroAstro that always helped me with everything.

Last but certainly not least, I want to thank my wife Debora, my parents Conceição and Jorge, and my sister and brother-in-law, Paola and Marco, for their support throughout the years. They always catch me when I fall, and they give me the strength to follow my

dreams. They have always supported me, and I owe them everything.

Thank you!!!

# Previously Published Material

**Chapter 2** includes prior work:

- [52] **Igor Kadota**, Elif Uysal-Biyikoglu, Rahul Singh and Eytan Modiano, "Minimizing Age of Information in Broadcast Wireless Networks," in Proceedings of IEEE Allerton, Sept. 2016, pp. 844–851.
- [51] **Igor Kadota**, Abhishek Sinha, Elif Uysal-Biyikoglu, Rahul Singh and Eytan Modiano, "Scheduling Policies for Minimizing Age of Information in Broadcast Wireless Networks," IEEE/ACM Transactions on Networking, vol. 26, no. 6, pp. 2637–2650, Dec. 2018.

**Chapter 3** includes prior work:

- [49] **Igor Kadota**, Abhishek Sinha and Eytan Modiano, "Optimizing Age of Information in Wireless Networks with Throughput Constraints," in Proceedings of IEEE INFOCOM, April 2018, pp. 1844–1852. This publication received the **Best Paper Award** and was featured in the MIT News article entitled "Keeping data fresh for wireless networks".
- [50] **Igor Kadota**, Abhishek Sinha and Eytan Modiano, "Scheduling Algorithms for Optimizing Age of Information in Wireless Networks with Throughput Constraints," IEEE/ACM Transactions on Networking, vol. 27, no. 4, pp. 1359–1372, Aug. 2019.

**Chapter 4** includes prior work:

- [44] **Igor Kadota** and Eytan Modiano, "Minimizing the Age of Information in Wireless Networks with Stochastic Arrivals," in Proceedings of ACM MobiHoc, July 2019, pp. 221–230. This publication was a **Best Paper Award Finalist**.
- [45] **Igor Kadota** and Eytan Modiano, "Minimizing the Age of Information in Wireless Networks with Stochastic Arrivals," IEEE Transactions on Mobile Computing, 2019. [Accepted for publication in Dec. 2019]

**Chapter 5** includes prior work:

- [47] **Igor Kadota**, Muhammad Shahir Rahman and Eytan Modiano, "Poster: Age of Information in Wireless Networks: from Theory to Implementation", ACM MobiCom, 2020. [Accepted for publication in Aug. 2020]
- [48] **Igor Kadota**, Muhammad Shahir Rahman and Eytan Modiano, "WiFresh: Age-of-Information from Theory to Implementation", Dec. 2019. [Under Review]

**Chapter 6** includes prior work:

- [46] **Igor Kadota** and Eytan Modiano, "Age of Information in Random Access Networks with Stochastic Arrivals", Aug. 2020. [Under Review]

**Chapters 2, 3 and 4** appeared in part in the book:

- [98] Yin Sun, **Igor Kadota**, Rajat Talak and Eytan Modiano, "Age of Information: A New Metric for Measuring Information Freshness", San Rafael, CA: Morgan & Claypool Publishers, 2019.

# Contents

# List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

# Introduction

Future Internet-of-Things (IoT) applications will increasingly rely on sharing time-sensitive information for monitoring and control. Examples are abundant: autonomous vehicles, smart factories, smart homes, immersive gaming, command and control, et al. Self-driving cars need to exchange safety-critical information with other vehicles and infrastructure. Swarms of drones need to exchange position, velocity and control information to enable collision prevention mechanisms. Cyber-physical systems in smart factories need to share status information to cooperate with each other and with humans. In such application domains, it is essential to keep information fresh, as outdated information loses its value and can lead to *system failures* and *safety risks*. *Information freshness is a challenging objective that goes beyond low latency, it requires that packets with low delay are delivered regularly over time to every destination in the network. In this thesis, we address the problem of keeping information fresh in wireless networks.*

To illustrate this challenging problem, consider a *monitoring system* composed of a remote monitor, a wireless base station, and $N$ mobile nodes. Each node $i \in \{1, 2, \cdots, N\}$ moves with velocity $v_i$ meters per second, generates position information with an average rate of $\lambda_i$ packets per second, and sends these packets to the remote monitor via the wireless base station. Assume that at time $t$, the latest packet received by the remote monitor from node $i$ had information about its position at time $\tau_i(t)$. Then, the uncertainty about node $i$'s position is given by $v_i(t - \tau_i(t))$, as illustrated in Fig. 1-1.

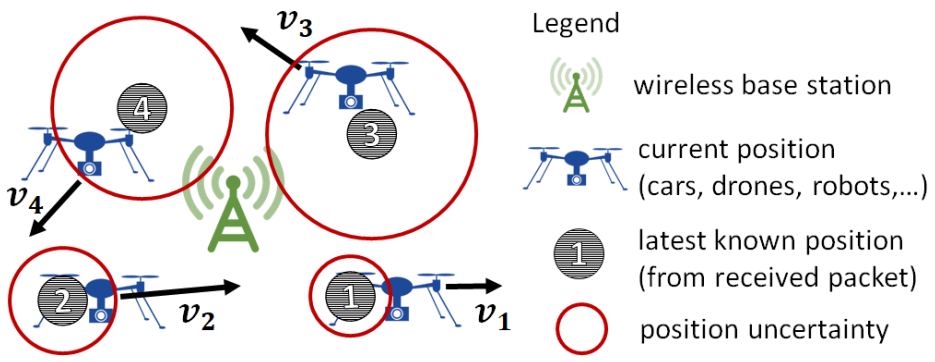



Figure 1-1: Illustration of a monitoring system. The wireless base station receives position information from the $N = 4$ mobile nodes and forwards this information to the remote monitor. The position uncertainty associated with node $i$ from the perspective of the remote monitor is represented by the red circle with radius $v_i(t - \tau_i(t))$ centered at the last known position of node $i$. The remote monitor does not know the current position of node $i$, which is illustrated by the blue drone.

The quantity $t - \tau_i(t)$ captures the freshness of information from the perspective of the remote monitor. In particular, $t - \tau_i(t) = 2$ seconds represents that at time $t$ the remote monitor knows the location of node $i$ two seconds ago. Ideally, if all mobile nodes could continuously generate position updates and transmit these updates to the remote monitor without delay, then the information at the remote monitor would be always fresh, with $\tau_i(t) = t$, and there would be no position uncertainty, i.e. $v_i(t - \tau_i(t)) = 0$ . However, real networks have inherent sources of latency (such as buffers) and limited communication resources (especially in wireless channels). Hence, to keep the information at the destination fresh, it is necessary to consider the networked system as a whole and optimize across the generation of packets at the sources, the queueing discipline at the buffers, and the transmission scheduling policy of the network.

In this thesis, we model the networked system and its performance requirements in terms of information freshness and use tools from stochastic control and mathematical optimization to develop network control algorithms with *provable* performance guarantees and low computational complexity. We then implement these algorithms using FPGA-based Software Defined Radios and/or Raspberry Pis to evaluate their performance in real operating conditions. Next, we formalize the concept of information freshness which will be used throughout this thesis.

## 1.1 Definition of Age of Information

The Age of Information (AoI) is a performance metric that was recently proposed in [57,59] and has been receiving increasing attention in the literature [7–11, 19, 21, 22, 31, 32, 37–39, 42, 49, 51–55, 57–59, 66, 74, 84, 87, 99–105, 113–115] for its application to communication systems that carry time-sensitive data. Consider a system in which packets are time-stamped upon arrival. Naturally, the higher the time-stamp of a packet, the fresher its information. Let $\tau(t)$ be the time-stamp of the *freshest packet received by the destination* by time $t$. Then, the AoI is defined as $h(t) := t - \tau(t)$. The AoI measures the time that elapsed since the generation of the freshest packet received by the destination. The value of $h(t)$ increases linearly over time while no fresher packet is received, representing the information getting older. At the moment a fresher packet is received, the time-stamp at the destination $\tau(t)$ is updated and the AoI is reduced to the packet delay, as illustrated in Fig. 1-2. *The AoI captures how fresh the information is from the perspective of the destination.*

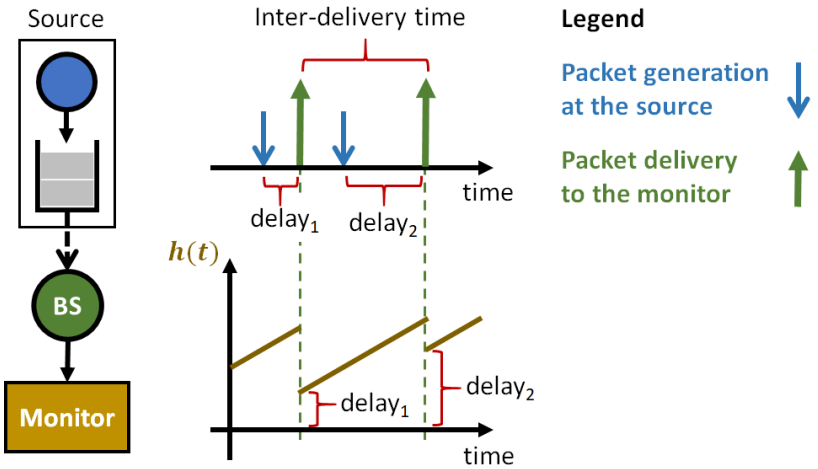



Figure 1-2: Illustration of the AoI evolution in a network with a single source sending packets to a single destination through a wireless base station (BS). Packets generated at the source wait in the queue before being served. The BS transmits packets in order of arrival, i.e. using a First-Come First-Served (FCFS) discipline.

The two parameters that influence AoI are packet delay and packet inter-delivery time. In general, controlling only one is insufficient for achieving good AoI performance. For example, consider a single server queue with Poisson arrivals and exponential service times,

i.e. an M/M/1 queue, with low arrival rate $\lambda \ll 1$ and fixed service rate $\mu = 1$. In this setting, the queue is often empty, resulting in low packet delay. Nonetheless, the AoI can still be high, since infrequent packet arrivals result in outdated information at the destination. In Fig. 1-3, we plot the expected AoI, the expected packet delay and the expected inter-delivery time as a function of $\lambda$ for the M/M/1 queue. The analytical expression for the AoI was obtained in [59] and the expressions for packet delay and inter-delivery time can be found in [30]. Notice that *the information is fresh, i.e. the AoI is low, when packets with low delay are delivered regularly to the destination.*

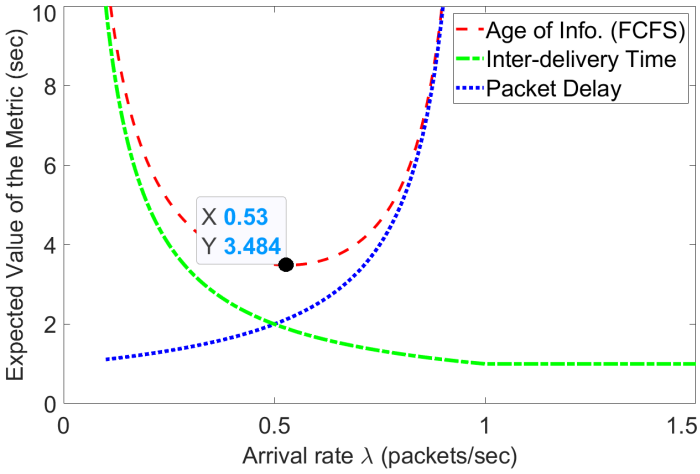



Figure 1-3: Expected delay, inter-delivery time and AoI for the M/M/1 queue with fixed service rate $\mu = 1$ and variable packet generation rate $\lambda$. The point of minimum AoI is $\lambda^* = 0.53$.

## 1.2 Literature Review

The problem of minimizing AoI was introduced in [57, 59] and has been explored using different approaches. Queueing Theory is used in [19, 21, 39, 54, 59, 66, 84, 115] to characterize the AoI performance of various important queueing systems. Game Theory is used in [26, 86, 112] to analyze the impact of communication interference on AoI. Information Theory is used in [81, 116] for designing source and channel coding schemes to improve AoI. The authors of [7, 8, 100, 113] consider the problem of optimizing the times in which packets are generated at the source in networks with energy-harvesting or maximum update

frequency constraints. Link scheduling optimization with respect to AoI has been recently considered in [10, 11, 31, 32, 37, 38, 42, 49, 51, 52, 58, 74, 87, 99, 101–105, 114]. Different applications of AoI in sensor networks, cellular networks and vehicular networks are analyzed and/or emulated in [5, 9, 22, 53]. A few papers [48, 57, 92, 95] have implemented AoI-based systems. This list of works is not exhaustive. For a comprehensive list we refer the reader to [65, 97, 98].

The problem of optimizing network scheduling decisions with respect to throughput and delivery times has been studied extensively in the literature. Throughput maximization of traffic with strict packet delay constraints has been addressed in [13, 34–36, 62, 62, 90]. Inter-delivery time is considered in [29, 70–72, 93, 94, 118] as a measure of service regularity. Most relevant to this thesis is the work on scheduling optimization with respect to Age of Information which is considered in [10, 11, 31, 32, 37, 38, 42, 49, 51, 52, 58, 74, 87, 99, 101–105, 114] and is further described next.

The authors of [10, 102] studied multi-hop networks, while other works addressed single-hop networks. Deterministic packet arrivals were considered in [49, 51, 52, 58, 101–105, 114], arbitrary arrivals in [10, 11, 31, 32, 99] and stochastic arrivals in [37, 38, 42, 74, 87, 104]. Networks with no queueing, i.e. when packets are discarded if not scheduled immediately upon arrival, were considered in [37, 38], First-Come First-Served (FCFS) queues were considered in [31, 32, 42, 104] and other works considered Last-Generated First-Served queues, which are often equivalent to the simpler Last-Come First-Served (LCFS) queues. Reliable links over which transmissions are always successful are considered in [10, 11, 31, 32, 37, 38, 42, 87, 99, 102, 114] and other works considered unreliable links. Additional details on related works are provided within the chapters in this thesis.

## 1.3 Outline and Main Contributions

In this thesis, we address the problem of minimizing the Age of Information in wireless networks. In particular, we consider a single-hop broadcast wireless network with a base station and a number of nodes sharing time-sensitive information through wireless links. We formulate a discrete-time decision problem to find a transmission scheduling policy that

minimizes AoI and evaluate its performance using analytical, numerical and experimental results.

In chapters 2 and 3, we consider sources that can generate packets with fresh information *on demand*. This assumption isolates the link scheduling problem from the effects of packet generation and queueing, allowing us to extract valuable insight from the scheduling problem. The more realistic case of stochastic arrivals is considered in chapters 4 and 6. In chapter 5, we study practical networks in which the packet generation is determined by the individual sensor. For example, position information from the GPS receiver is generated at 1 Hz and inertial measurements from the IMU at 100 Hz. The remainder of this thesis is organized as follows.

**Chapter 2. Age of Information in Wireless Networks**

In this chapter, we formulate a discrete-time decision problem for minimizing AoI in wireless networks, and show that the computational complexity of the optimal solution grows exponentially with the size of the network. Then, we consider symmetric networks and show that the optimal solution becomes a simple Maximum Age First policy which activates the link associated with the highest current AoI. For general networks, we discuss four low-complexity scheduling policies: Maximum Age First policy, Stationary Randomized policy, Max-Weight policy, and Whittle's Index policy; and derive performance guarantees for each of them as a function of the network configuration. Notably, we show that both Stationary Randomized policy and Max-Weight policy are guaranteed to be within a factor of two away from the minimum AoI possible, regardless of the network configuration. Numerical results show that both Max-Weight and Whittle's Index outperform the other scheduling policies in every configuration simulated, achieving near optimal AoI in various network settings.

To the best of our knowledge, this is the first work to derive performance guarantees for transmission scheduling policies that attempt to minimize AoI in wireless networks with unreliable channels.

**Chapter 3. Throughput Constrained AoI Optimization**

We consider the problem of minimizing AoI in the wireless network while simultaneously satisfying throughput requirements from the individual nodes. Throughput requirements can either capture an attribute of the nodes or be used to enforce fair allocation of resources in the network. Notice that a link scheduling policy that minimizes AoI in the network is not necessarily fair.

In this chapter, we develop two low-complexity transmission scheduling policies, namely Stationary Randomized policy and Drift-Plus-Penalty policy, and show that both are guaranteed to be within a factor of two away from the minimum AoI possible, while simultaneously satisfying any feasible[1] throughput requirements.

To the best of our knowledge, this is the first work to consider AoI-based policies that provably satisfy throughput constraints of multiple destinations simultaneously.

**Chapter 4. AoI in Wireless Networks with Stochastic Arrivals**

In this chapter, we consider sources that generate packets according to a stochastic process and enqueue them in separate (per source) queues. We address link scheduling optimization to minimize AoI in wireless networks operating under three common queueing disciplines. We develop both a Stationary Randomized policy and a Max-Weight policy under each queueing discipline. Our approach allows us to *evaluate the combined impact* of the stochastic arrivals, queueing discipline and scheduling policy on AoI. We evaluate the AoI performance both analytically and using simulations. Numerical results show that the Max-Weight policy with LCFS queues achieves near optimal performance.

**Chapter 5. WiFresh: AoI from Theory to Implementation**

In this chapter, we study AoI in practical wireless networks. Leveraging the theoretical results, we propose WiFresh: an unconventional architecture that scales gracefully, achieving near optimal information freshness in wireless networks of any size, even when the network is overloaded. We propose and realize two strategies for implementing WiFresh:

[1] We say that a set of throughput requirements is feasible if there exists an admissible scheduling policy that can satisfy the requirements.

one at the MAC layer in the network of FPGA-based Software Defined Radios in Fig.1-4 using hardware-level programming, and another at the Application layer, without modifications to lower layers of the communication system, in the network of Raspberry Pis in Fig. 1-5 using Python 3. Our experimental results show that WiFresh can *improve information freshness by two orders of magnitude* when compared to an equivalent standard WiFi network.

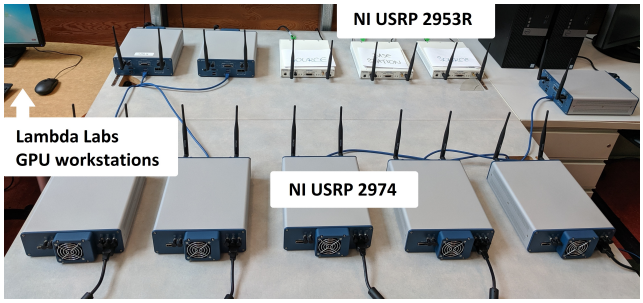



Figure 1-4: Software Defined Radio testbed.

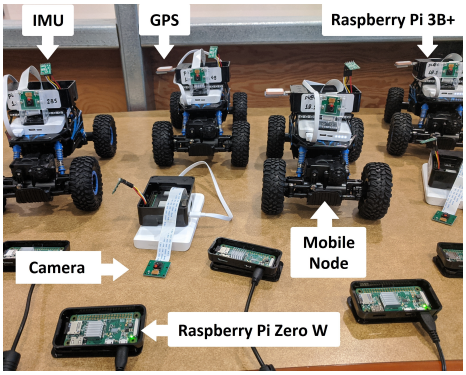



Figure 1-5: Raspberry Pi testbed.

To the best of our knowledge, this is the first experimental evaluation of a networked system that scales gracefully in terms of information freshness.

**Chapter 6. AoI in Random Access Networks**

In this chapter, we study AoI in wireless networks employing Random Access mechanisms, in particular Slotted-ALOHA and Carrier-Sense Multiple Access (CSMA). We propose a discrete-time framework to analyze and optimize the average AoI in the Random Access network. Furthermore, we implement the optimized Random Access mechanism in the Software Defined Radio testbed in Fig.1-4 and compare the AoI measurements with analytical and numerical results in order to validate our framework. Our approach allows us to evaluate the combined impact of the packet generation rate, transmission probability and size of the network on the AoI performance.

To the best of our knowledge, this is the first work to provide theoretical results on the optimization of a CSMA network with stochastic packet generation and packet collisions.

**Remark 1.1.** *For simplicity of notation, throughout the thesis we assume that limits exist and use* lim *instead of* lim sup *or* lim inf.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

# Age of Information in Wireless Networks

Traditionally, networks have been designed to maximize data throughput and minimize packet latency. With the emergence of new types of networks such as vehicular networks, UAV networks and sensor networks, other performance requirements are increasingly relevant. In particular, the Age of Information (AoI) has been receiving attention in the literature for its application in communication systems that carry time-sensitive data. In this chapter, we address the problem of minimizing AoI in broadcast wireless networks.

Consider a cyber-physical system where a number of nodes are transmitting time-sensitive information to a monitor over unreliable wireless channels. Each node samples information from a physical phenomena (e.g. position, proximity to obstacles and energy consumption) and transmits this information to the monitor. Ideally, the monitor receives fresh information about every physical phenomena continuously over time. However, due to limitations of the wireless channel, this is often impractical. In such cases, the system has to manage the use of the available channel resources in order to keep the information at the monitor as fresh as possible, i.e. to minimize the Age of Information in the network.

*In this chapter, we consider a broadcast singe-hop wireless network with a base station and a number of nodes sharing time-sensitive information through unreliable communication links*, as illustrated in Fig. 2-1. We formulate a discrete-time decision problem to find a transmission scheduling policy that minimizes the expected weighted sum Age of Information of the network. First, we obtain the optimal solution using Dynamic Programming

and show that the computational complexity of such solution grows exponentially with the size of the network. To overcome this problem, known as the *curse of dimensionality*, and gain insight into the minimization of AoI, we introduce four low-complexity scheduling policies: Maximum Age First, Randomized, Max-Weight, and Whittle's Index, and derive performance guarantees for each of them as a function of the network configuration. In particular, we show that the Maximum Age First policy which activates the link associated with highest current age is optimal for symmetric networks. For general networks, we show that both Randomized and Max-Weight are guaranteed to be within a factor of two away from the minimum AoI possible, regardless of the network configuration. Numerical results show that both Max-Weight and Whittle's Index outperform the other scheduling policies in every configuration simulated, and achieve near optimal performance. To the best of our knowledge, this is the first work[1] to:

- provide a transmission scheduling policy that minimizes AoI in wireless networks with unreliable channels; and
- derive performance guarantees for scheduling policies that attempt to minimize AoI in wireless networks with unreliable channels.

The remainder of this chapter is organized as follows. In Sec. 2.1, the network model is presented and the Dynamic Programming solution is proposed. In Sec. 2.2, we derive performance guarantees for the Maximum Age First, Randomized, Max-Weight and Whittle's Index policies. Numerical results are presented in Sec. 2.3. The chapter is concluded in Sec. 2.4 with a discussion about generalizations of the AoI minimization problem.

## 2.1 System Model

Consider a single-hop wireless network with a base station (BS) and $N$ nodes sharing time-sensitive information through unreliable communication links, as illustrated in Fig. 2-1. Let the time be slotted, with slot duration normalized to unity and slot index $t \in \{1,2,\cdots,T\}$, where $T$ is the time-horizon of this discrete-time system. The broadcast wireless channel allows at most one packet transmission per slot. In each time-slot $t$, the BS either idles

[1] This work was first published in [52] and [51].

or schedules a transmission in a selected link $i \in \{1,2,\cdots,N\}$. Let $u_i(t) \in \{0,1\}$ be the indicator function that is equal to 1 when the BS selects link $i$ during slot $t$, and $u_i(t) = 0$ otherwise. *When $u_i(t) = 1$ the corresponding source samples fresh information, generates a new packet and transmits this packet over link i.* Notice that packets are not enqueued. Packets are generated on-demand[2] and transmitted in the same slot. Since the BS can select at most one link at any given slot $t$, we have

$$\sum_{i=1}^{N} u_i(t) \leq 1, \ \forall t \in \{1,\cdots,T\} \ . \tag{2.1}$$

The transmission scheduling policy governs the sequence of decisions $\{u_i(t)\}_{i=1}^{N}$ of the BS over time.

Let $c_i(t) \in \{0,1\}$ represent the channel state associated with link $i$ during slot $t$. When the channel is *ON*, we have $c_i(t) = 1$, and when the channel is *OFF*, we have $c_i(t) = 0$. The channel state process is assumed i.i.d. over time and independent across different links, with $\mathbb{P}(c_i(t) = 1) = p_i, \forall i,t$.[3]

Let $d_i(t) \in \{0,1\}$ be the indicator function that is equal to 1 when the transmission in link $i$ during slot $t$ is successful, and $d_i(t) = 0$ otherwise. A successful transmission occurs when a link is selected and the associated channel is ON, implying that $d_i(t) = c_i(t)u_i(t), \forall i,t$. Moreover, since the BS does not know the channel states prior to making scheduling decisions, $u_i(t)$ and $c_i(t)$ are independent, and

$$\mathbb{E}\left[d_i(t)\right] = p_i \mathbb{E}\left[u_i(t)\right], \forall i,t \ . \tag{2.2}$$

The scheduling policies considered in this chapter are non-anticipative, i.e. policies that do not use future information in making scheduling decisions. Let $\Pi$ be the class of non-anticipative policies and let $\pi \in \Pi$ be an arbitrary admissible policy. Our goal is to develop scheduling policies $\pi$ that minimize the average AoI in the network. Next, we formulate the AoI minimization problem.

[2] Stochastic packet generation and queueing are discussed in chapter 4.

[3] The assumption of fixed and known channel reliabilities $\{p_i\}_{i=1}^{N}$ is used in chapters 2, 3 and 4. This assumption is not used in chapter 5 when the deployed system learns $p_i$ over time.

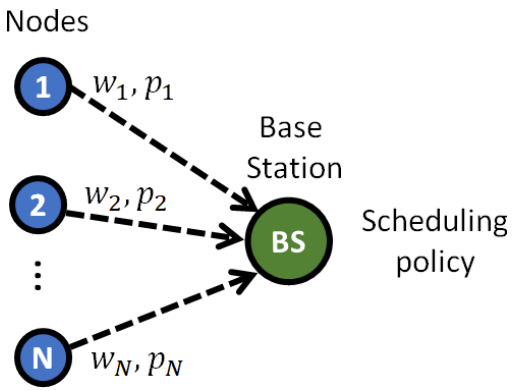



Figure 2-1: Illustration of the single-hop wireless network. On the left, we have $N$ nodes. In the center, we have $N$ links with their associated priority (or weight) $w_i$ and probability of a successfull packet transmission $p_i$. On the right, we have the base station running a transmission scheduling policy.

As defined in Sec. 1.1, the Age of Information depicts how old the information is from the perspective of the destination. Let $h_i(t)$ be the positive real number that represents the AoI associated link $i$ at the beginning of slot $t$. If the destination associated with link $i$ does not receive a packet during slot $t$, then $h_i(t+1) = h_i(t)+1$, since the information at the destination is one slot older. In contrast, if the destination receives a packet during slot $t$, then $h_i(t+1) = 1$, because the received packet was generated at the beginning of slot $t$. The evolution of $h_i(t)$ follows

$$h_i(t+1) = \begin{cases} 1 & \text{, if } d_i(t) = 1\,; \\ h_i(t)+1 & \text{, otherwise.} \end{cases} \tag{2.3}$$

The time-average AoI of link $i$ during the first $T$ slots is captured by $\mathbb{E}\left[\sum_{t=1}^{T} h_i(t)\right]/T$, where the expectation is with respect to the randomness in the channel state $c_i(t)$ and scheduling decisions $u_i(t)$. For capturing the freshness of the information of a network employing scheduling policy $\pi \in \Pi$, we define the Expected Weighted Sum AoI (EWSAoI) as

$$\mathbb{E}[J_T^{\pi}] = \frac{1}{TN}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} w_i h_i^{\pi}(t) \,\middle|\, \vec{h}(1)\right], \tag{2.4}$$

where $\vec{h}(1) = [h_1(1), \cdots, h_N(1)]^T$ is the vector of initial AoI, and $w_i$ is the positive real number that represents the priority (or weight) of link $i$. For notation simplicity, we omit

$\vec{h}(1)$ henceforth. We denote by *AoI-optimal*, the scheduling policy $\pi^* \in \Pi$ that achieves minimum EWSAoI, namely

Finite-horizon AoI optimization

$$\mathbb{E}[J_T^*] = \min_{\pi\in\Pi} \left\{ \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} w_i \mathbb{E}\left[h_i^{\pi}(t)\right] \right\} , \tag{2.5a}$$

$$\text{s.t.} \sum_{i=1}^{N} u_i(t) \leq 1 \ , \forall t \ . \tag{2.5b}$$

Throughout this chapter, we discuss both the finite-horizon problem in (2.5a)-(2.5b) and the related infinite-horizon problem

Infinite-horizon AoI optimization

$$\mathbb{E}[J^*] = \min_{\pi\in\Pi} \left\{ \lim_{T\to\infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} w_i \mathbb{E}\left[h_i^{\pi}(t)\right] \right\} , \tag{2.6a}$$

$$\text{s.t.} \sum_{i=1}^{N} u_i(t) \leq 1 \ , \forall t \ . \tag{2.6b}$$

Next, we address the AoI optimization using Dynamic Programming (DP) and evaluate the computational complexity of the solution.

### 2.1.1 Dynamic Programming Formulation

In this section, the finite-horizon AoI optimization in (2.5a)-(2.5b) is formulated and solved using DP. The objective function in (2.5a) evolves in discrete steps and has an additive cost, making it suitable for a DP formulation. The components of the DP formulation, namely network state, control variable, state transition, and cost function, are described next.

- **Network State**. The vector $\vec{h}(t)$ is the network state at the beginning of slot $t$.
- **Control Variable**. The set $\{u_i(t)\}_{i=1}^{N}$ are the control variables during slot $t$.
- **State Transitions**. The evolution of $h_i(t)$ is divided in two cases: i) when the scheduling policy selects link $i$ during slot $t$, namely $u_i(t) = 1$, then the state transi-

tion to slot $t+1$ depends on the channel condition as follows

$$\mathbb{P}\left(h_i(t+1)=h_i(t)+1|u_i(t)=1,h_i(t)\right)=1-p_i\,; \quad \text{[channel OFF]} \tag{2.7a}$$

$$\mathbb{P}\left(h_i(t+1)=1|u_i(t)=1,h_i(t)\right)=p_i\,, \quad \text{[channel ON]} \tag{2.7b}$$

and ii) when the scheduling policy does not select link $i$, namely $u_i(t)=0$, then the state transition is deterministic

$$\mathbb{P}\left(h_i(t+1)=h_i(t)+1|u_i(t)=0,h_i(t)\right)=1\,. \tag{2.8}$$

- **Cost Function**. The cost at the transition from slot $t$ to slot $t+1$ is given by

$$g_t(\vec{h}(t))=\sum_{i=1}^{N} w_i h_i(t)\,. \tag{2.9}$$

With the components of the DP formulation described, next we present the cost-to-go function. Substituting the cost $g_t(\vec{h}(t))$ into the objective function in (2.5a) yields

$$\mathbb{E}[J_T^*]=\frac{1}{TN}\min_{\pi\in\Pi}\left\{\sum_{t=1}^{T}\mathbb{E}\left[g_t(\vec{h}(t))\right]\right\}\,. \tag{2.10}$$

For a given $\vec{w}$, the optimization problem in (2.10) is solved by applying the cost-to-go function $\mathscr{J}_t(\vec{h}(t))$ iteratively, backwards in time. The initial value of the cost-to-go function is $\mathscr{J}_{T+1}(\vec{h}(T+1))=0$, for all vectors $\vec{h}(T+1)$, and the recursion for slot $t\in\{1,2,\cdots,T\}$ is given by

$$\begin{aligned}\mathscr{J}_t(\vec{h}(t))=&\min_{u_i(t)}\mathbb{E}[g_t(\vec{h}(t))+\mathscr{J}_{t+1}(\vec{h}(t+1))]\,;\\ =&g_t(\vec{h}(t))+\min_{u_i(t)}\mathbb{E}[\mathscr{J}_{t+1}(\vec{h}(t+1))]\,.\end{aligned} \tag{2.11}$$

At each step $t$ and for every possible state $\vec{h}(t)$, the value of $\mathscr{J}_t(\vec{h}(t))$ is attained by choosing the set of control variables $\{u_i(t)\}_{i=1}^N$ subject to $\sum_{i=1}^N u_i(t)\leq 1$, which minimizes the RHS of (2.11). This recursion is known as Value Iteration [12]. By keeping track of

the choices of $\{u_i(t)\}_{i=1}^N$ for every possible tuple $(t,\vec{h}(t))$, the AoI-optimal policy $\pi^*$ is obtained. The output of the recursion (2.11) at $t=1$ and for the initial vector $\vec{h}(1)$ is the AoI-optimal objective function $\mathbb{E}[J_T^*]$ in (2.5a).

A negative aspect of this approach is that evaluating the optimal scheduling decision $\{u_i(t)\}_{i=1}^N$ for every possible tuple $(t,\vec{h}(t))$ can be computationally demanding, especially for networks with a large number of nodes. The parameter $h_i(t)$ can take at least $t$ different values, $h_i(t) \in \{1,2,\cdots,h_i(1)+t-1\}$. Hence, the set of possible vectors $\vec{h}(t)$ has cardinality at least $t^N$. For each possible tuple $(t,\vec{h}(t))$, the Dynamic Program compares the outcome of $N+1$ possible sets $\{u_i(t)\}_{i=1}^N$. For a time-horizon of $T$ slots, this amounts to $\mathcal{O}(NT^N)$ operations. *Computational complexity grows exponentially with the number of nodes N in the network.* To overcome this problem, known as the *curse of dimensionality*, and gain insight into the minimization of the Age of Information, in the next section we discuss low-complexity scheduling policies and evaluate their performance.

## 2.2 Scheduling Policies

In this section, we consider four low-complexity scheduling policies, namely Maximum Age First, Stationary Randomized, Max-Weight, and Whittle's Index, and derive performance guarantees for each of them as a function of the network configuration. Unless stated otherwise, henceforth in this section we consider the infinite-horizon problem in (2.6a)-(2.6b) with $T \to \infty$. The focus on the long-term behavior of the system allows us to derive simpler and more insightful policies and performance guarantees.

The performance of an arbitrary admissible policy $\eta \in \Pi$ in the limit as $T$ goes to infinity is given by $\mathbb{E}[J^\eta] = \lim_{T\to\infty} \mathbb{E}[J_T^\eta]$ from (2.4) and the optimal performance is $\mathbb{E}[J^*] = \min_{\eta\in\Pi} \mathbb{E}[J^\eta]$ from (2.6a). Ideally, when expressions for $\mathbb{E}[J^*]$ and $\mathbb{E}[J^\eta]$ are available, we define the optimality ratio[4] $\mathbb{E}[J^\eta]/\mathbb{E}[J^*]$ and say that policy $\eta$ is $(\mathbb{E}[J^\eta]/\mathbb{E}[J^*])$-optimal. Naturally, the closer the optimality ratio is to unity, the better is the performance of policy $\eta$ in terms of Age of Information.

Alternatively, when expressions for $\mathbb{E}[J^*]$ and $\mathbb{E}[J^\eta]$ are not available, we define the

[4]Optimality Ratio is also known as Approximation Ratio.

ratio

$$\rho^{\eta} := \frac{U_B^{\eta}}{L_B}\,, \tag{2.12}$$

where $L_B$ is a lower bound to the AoI-optimal performance and $U_B^{\eta}$ is an upper bound to the performance of policy $\eta$, namely

$$L_B \leq \mathbb{E}\left[J^*\right] \leq \mathbb{E}\left[J^{\eta}\right] \leq U_B^{\eta}\,. \tag{2.13}$$

It follows from (2.13) that $\mathbb{E}\left[J^{\eta}\right]/\mathbb{E}\left[J^*\right] \leq \rho^{\eta}$ and we can say that policy $\eta$ is $\rho^{\eta}$-optimal. Hence, if policy $\eta$ is 2-optimal, it means that its performance is guaranteed to be within a factor of 2 away from the minimum AoI possible, i.e. $\mathbb{E}\left[J^*\right] \leq \mathbb{E}\left[J^{\eta}\right] \leq 2\mathbb{E}\left[J^*\right]$.

Next, we obtain a lower bound $L_B$ as a function of the network configuration $(N, p_i, w_i)$. Then, we analyze each of the four scheduling policies of interest and derive closed-form expressions for their upper bounds $U_B^{\eta}$ and performance guarantees $\rho^{\eta}$. Table 2.1 summarizes the key notation in this thesis.

Table 2.1: Description of key notation.

| | |
|---|---|
| $N$ | number of nodes. Link index is $i \in \{1, 2, \cdots, N\}$ |
| $T$ | number of slots. Slot index is $t \in \{1, 2, \cdots, T\}$ |
| $p_i$ | probability of successful transmission in link $i$ |
| $\pi$ | admissible non-anticipative scheduling policy |
| $h_i(t)$ | Age of Information associated with link $i$ at the beginning of slot $t$ |
| $w_i$ | weight of link $i$. Represents the relative importance of link $i$ |
| $\mathbb{E}[J_T^{\pi}]$ | Expected Weighted Sum Age of Information performance of policy $\pi$ |
| $L_B$ | Lower Bound on $\lim_{T\to\infty} \mathbb{E}[J_T^{\pi}]$ for *any admissible policy* $\pi$ |
| $U_B^{\pi}$ | Upper Bound on $\lim_{T\to\infty} \mathbb{E}[J_T^{\pi}]$ for a *particular policy* $\pi$ |
| $\rho^{\pi}$ | performance guarantee associated with policy $\pi$ |
| $D_i(T)$ | number of packet deliveries through link $i$ up to and including slot $T$ |
| $\Upsilon_i(T)$ | number of packet transmissions through link $i$ up to and including slot $T$ |
| $I_i[m]$ | number of slots between consecutive deliveries by link $i$ |
| $R_i$ | number of slots remaining after the last delivery by link $i$ |
| $\bar{\mathbb{M}}[.]$ | operator that calculates the sample mean of a set of values |
| $\bar{\mathbb{V}}[.]$ | operator that calculates the sample variance of a set of values |

### 2.2.1 Universal Lower Bound

In this section, we find a lower bound $L_B$ to the minimum AoI achievable by *any admissible scheduling policy* $\pi \in \Pi$. The expression for $L_B$ depends on the statistics of the sets $\{w_i\}_{i=1}^N$ and $\{\sqrt{w_i/p_i}\}_{i=1}^N$. We define the operators that calculate the sample mean and sample variance of a set of values $\mathbf{x}$ as $\bar{\mathbb{M}}[\mathbf{x}]$ and $\bar{\mathbb{V}}[\mathbf{x}]$, respectively. The sample mean and sample variance of $\{w_i\}_{i=1}^N$ are

$$\bar{\mathbb{M}}[w_i] = \frac{1}{N}\sum_{j=1}^{N} w_j \quad \text{and} \quad \bar{\mathbb{V}}[w_i] = \frac{1}{N}\sum_{j=1}^{N} \left(w_j - \bar{\mathbb{M}}[w_i]\right)^2 . \tag{2.14}$$

The sample mean and sample variance of $\{\sqrt{w_i/p_i}\}_{i=1}^N$ are calculated analogously.

**Theorem 2.1** (Lower Bound)**.** *For any wireless network with parameters* $(N, p_i, w_i)$ *and an infinite time-horizon, we have* $L_B \leq \lim_{T\to\infty} \mathbb{E}\left[J_T^{\pi}\right]$, $\forall \pi \in \Pi$, *where*

$$L_B = \frac{N}{2}\left(\bar{\mathbb{M}}\left[\sqrt{\frac{w_i}{p_i}}\right]\right)^2 + \frac{1}{2}\bar{\mathbb{M}}[w_i] . \tag{2.15}$$

*Proof.* First, we use a sample path argument to characterize the evolution of $\vec{h}(t)$ over time. Then, we derive an expression for the objective function of the infinite-horizon problem, namely $\lim_{T\to\infty} J_T^{\pi}$, and manipulate this expression to obtain the $L_B$ in (2.15). Fatou's lemma is employed to establish Theorem 2.1.

Consider an admissible scheduling policy $\pi \in \Pi$ running on a network for the time-horizon of $T$ slots. Let $\Omega$ be the sample space associated with this network and let $\omega \in \Omega$ be a sample path. For this sample path, let $D_i(T)$ be the total number of packets delivered by link $i$ up to and including slot $T$, let $I_i[m]$ be the number of slots between the $(m-1)$th and $m$th deliveries by link $i$, i.e. the inter-delivery times of link $i$, and let $R_i$ be the number of slots remaining after the last packet delivery by the same link. Then, the time-horizon

can be written as follows

$$T = \sum_{m=1}^{D_i(T)} I_i[m] + R_i, \forall i \in \{1,2,\cdots,N\} \,. \tag{2.16}$$

The evolution of $h_i(t)$ is well-defined in each of the time intervals $I_i[m]$ and $R_i$, as illustrated in Fig. 2-2. During the slots associated with the interval $I_i[m]$, the parameter $h_i(t)$ evolves as $1,2,\cdots,I_i[m]$. During the slots associated with the interval $R_i$, the value of $h_i(t)$ evolves as $1,2,\cdots,R_i$. Hence, the objective function in (2.5a) can be rewritten as

$$\begin{aligned} J_T^{\pi} =& \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} w_i \, h_i(t) = \frac{1}{N} \sum_{i=1}^{N} \frac{w_i}{T} \left[ \sum_{t=1}^{T} h_i(t) \right] \\ =& \frac{1}{N} \sum_{i=1}^{N} \frac{w_i}{T} \left[ \sum_{m=1}^{D_i(T)} \frac{(I_i[m]+1) I_i[m]}{2} + \frac{(R_i+1)R_i}{2} \right] , \end{aligned} \tag{2.17}$$

and, using (2.16) to substitute the sum of the linear terms $I_i[m]$ and $R_i$ by $T$, we get

$$\begin{aligned} J_T^{\pi} =& \frac{1}{2N} \sum_{i=1}^{N} \frac{w_i}{T} \left[ \sum_{m=1}^{D_i(T)} I_i^2[m] + R_i^2 + T \right] \\ =& \frac{1}{2N} \sum_{i=1}^{N} w_i \left[ \frac{D_i(T)}{T} \left( \frac{1}{D_i(T)} \sum_{m=1}^{D_i(T)} I_i^2[m] \right) + \frac{R_i^2}{T} + 1 \right] . \end{aligned} \tag{2.18}$$

Using the operator $\bar{\mathbb{M}}[.]$, let the sample mean of $I_i[m]$ and $I_i^2[m]$ for a fixed link $i$ be

$$\bar{\mathbb{M}}[I_i] = \frac{1}{D_i(T)} \sum_{m=1}^{D_i(T)} I_i[m] \quad \text{and} \quad \bar{\mathbb{M}}[I_i^2] = \frac{1}{D_i(T)} \sum_{m=1}^{D_i(T)} I_i^2[m] \,. \tag{2.19}$$

and, using $\bar{\mathbb{V}}[.]$, let the sample variance for a fixed link $i$ be

$$\bar{\mathbb{V}}[I_i] = \frac{1}{D_i(T)} \sum_{m=1}^{D_i(T)} \left( I_i[m] - \bar{\mathbb{M}}[I_i] \right)^2 \,. \tag{2.20}$$

Notice that the sample variance is positive valued and $\bar{\mathbb{V}}[I_i] = \bar{\mathbb{M}}[I_i^2] - \left(\bar{\mathbb{M}}[I_i]\right)^2$. For simplicity of notation, the time-horizon $T$ is omitted in both operators.

Combining (2.16) and (2.19) yields

$$\frac{T}{D_i(T)} = \frac{\sum_{m=1}^{D_i(T)} I_i[m] + R_i}{D_i(T)} = \bar{\mathbb{M}}[I_i] + \frac{R_i}{D_i(T)} \,. \tag{2.21}$$

Substituting (2.19) and (2.21) into the objective function in (2.18) gives

$$J_T^{\pi} = \frac{1}{2N} \sum_{i=1}^{N} w_i \left[ \left[ \bar{\mathbb{M}}[I_i] + \frac{R_i}{D_i(T)} \right]^{-1} \bar{\mathbb{M}}[I_i^2] + \frac{R_i^2}{T} + 1 \right] \quad \text{w.p.1} \,. \tag{2.22}$$

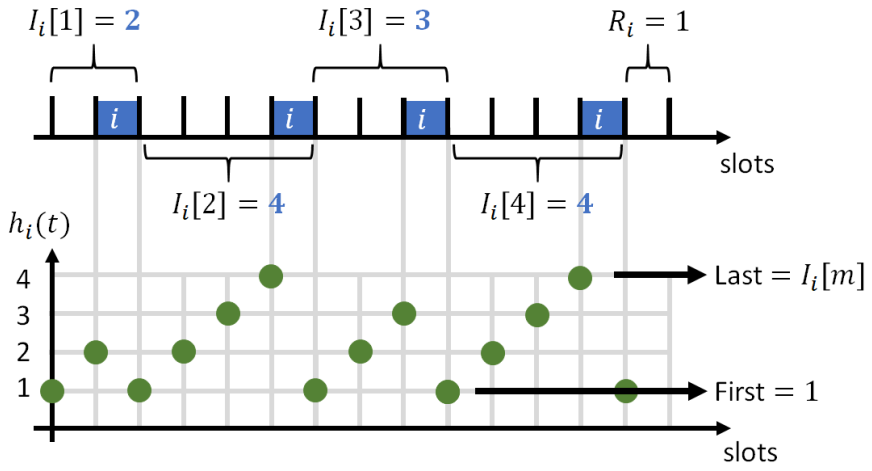



Figure 2-2: Evolution of $h_i(t)$ over the time-horizon of $T = 14$ slots for a target link $i$. On the top, successful packet transmissions over link $i$ are represented by blue boxes. Notice that $D_i(T) = 4$ packet deliveries and that $T = I_i[1] + I_i[2] + I_i[3] + I_i[4] + R_i$. On the bottom, evolution of $h_i(t)$ according to (2.3). Notice the sawtooth pattern. During any interval $I_i[m]$ the AoI always grows from 1 to $I_i[m]$.

To simplify (2.22), consider the infinite-horizon problem with $T \to \infty$ and assume that the admissible class $\Pi$ *does not contain policies that starve links*. A policy $\pi$ is said to starve link $i$ if, with a positive probability, it stops transmitting packet via link $i$ after slot $T' < \infty$. Notice that if $\pi$ starves link $i$, then the expected number of slots after the last packet delivery grows indefinitely $\mathbb{E}[R_i] \to \infty$ and, as a result, the objective function $\mathbb{E}[J_T^{\pi}] \to \infty$. Therefore, policies that starve links are excluded from the class of admissible policies $\Pi$ *without loss of optimality*.

Since policies in $\Pi$ select every link $i$ repeatedly and each link activation results in a successful packet delivery with positive probability $p_i$, it follows that $I_i[m]$ and $R_i$ are finite

with probability one. Thus, in the limit as $T \to \infty$, we have $R_i^2/T \to 0$, $D_i(T) \to \infty$ and $R_i/D_i(T) \to 0$. Applying those limits to $J_T^\pi$ in (2.22) gives the *objective function of the infinite-horizon AoI problem*

$$\lim_{T\to\infty} J_T^\pi = \frac{1}{2N}\sum_{i=1}^{N} w_i \left[\frac{\bar{\mathbb{M}}[I_i^2]}{\bar{\mathbb{M}}[I_i]} + 1\right] \quad \text{w.p.1 .} \tag{2.23}$$

This insightful expression depicts the relationship between AoI and the moments of the inter-delivery time $I_i[m]$.

Denote the total number of packets *transmitted* by link $i$ up to and including slot $T$ as $\Upsilon_i(T) = \sum_{t=1}^{T} u_i(t)$. Since at most one link can be selected at any given slot, i.e. $\sum_{i=1}^{N} u_i(t) \leq 1, \forall t$, we have

$$\sum_{i=1}^{N} \Upsilon_i(T) = \sum_{t=1}^{T}\sum_{i=1}^{N} u_i(t) \leq T \quad \text{w.p.1 .} \tag{2.24}$$

Moreover, by the strong law of large numbers, we know that in the limit as $T \to \infty$ the ratio of the number of packet deliveries by the number of packet transmissions is

$$\lim_{T\to\infty} \frac{D_i(T)}{\Upsilon_i(T)} = p_i \quad \text{w.p.1 .} \tag{2.25}$$

With the definition of $\Upsilon_i(T)$ and the operator $\bar{\mathbb{V}}[I_i]$, we obtain $L_B$ by manipulating the objective function of the infinite-horizon AoI problem in (2.23) as follows

$$\begin{aligned}
\lim_{T\to\infty} J_T^\pi &= \frac{1}{2N}\sum_{i=1}^{N} w_i \left[\frac{\bar{\mathbb{V}}[I_i]}{\bar{\mathbb{M}}[I_i]} + \bar{\mathbb{M}}[I_i] + 1\right] \overset{(a)}{\geq} \frac{1}{2N}\sum_{i=1}^{N} w_i \bar{\mathbb{M}}[I_i] + \frac{1}{2N}\sum_{i=1}^{N} w_i \\
&\overset{(b)}{=} \lim_{T\to\infty} \frac{1}{2N}\sum_{i=1}^{N} w_i \frac{T}{D_i(T)} + \frac{1}{2N}\sum_{i=1}^{N} w_i \\
&\overset{(c)}{\geq} \lim_{T\to\infty} \frac{1}{2N}\left(\sum_{j=1}^{N} \Upsilon_j(T)\right)\left(\sum_{i=1}^{N} \frac{w_i}{D_i(T)}\right) + \frac{1}{2N}\sum_{i=1}^{N} w_i \\
&\overset{(d)}{\geq} \lim_{T\to\infty} \frac{1}{2N}\left(\sum_{i=1}^{N} \sqrt{\frac{w_i \Upsilon_i(T)}{D_i(T)}}\right)^2 + \frac{1}{2N}\sum_{i=1}^{N} w_i \\
&\overset{(e)}{=} \frac{1}{2N}\left(\sum_{i=1}^{N} \sqrt{\frac{w_i}{p_i}}\right)^2 + \frac{1}{2N}\sum_{i=1}^{N} w_i \quad w.p.1 \,,
\end{aligned} \tag{2.26}$$

where (a) uses the fact that $\bar{\mathbb{V}}[I_i] \geq 0$, (b) uses (2.16), (c) uses the inequality in (2.24), (d) uses Cauchy-Schwarz inequality and (e) uses the equality in (2.25). Notice that (2.26) holds for all $\pi \in \Pi$ and that it gives the expression for $L_B$ found in (2.15).

Finally, since $J_T^\pi$ in (2.17) is positive for every $\pi \in \Pi$ and for every $T$, we employ Fatou's lemma to (2.26) and establish that $\lim_{T\to\infty} \mathbb{E}[J_T^\pi] \geq \mathbb{E}[\lim_{T\to\infty} J_T^\pi] \geq L_B, \forall \pi \in \Pi$. ■

The sequence of inequalities in (2.26) that led to $\lim_{T\to\infty} \mathbb{E}[J_T^\pi] \geq L_B, \forall \pi \in \Pi$ could have rendered a loose lower bound. However, in the next section, we use $L_B$ to derive a performance guarantee $\rho^{MAF}$ for the Maximum Age First policy and show that $\rho^{MAF} \to 1$ for symmetric networks with large $N$, i.e. under these conditions the value of $L_B$ is as tight as possible. Furthermore, the numerical results in Sec. 2.3 suggest that the lower bound is tight in a variety of network configurations. In the upcoming sections, we derive performance guarantees for Maximum Age First, Randomized, Max-Weight, and Whittle's Index policies.

### 2.2.2 Maximum Age First Policy

In this section, we study the Maximum Age First (MAF) policy and show that under some conditions on the underlying network it is AoI-optimal. Moreover, for general networks, we obtain a closed-form expression for the performance guarantee $\rho^{MAF}$ as a function of $(N, w_i, p_i)$. We introduced the MAF policy in [52] with the name of Greedy policy.

**Definition 2.2** (Maximum Age First policy)**.** *The MAF policy selects, in each slot $t$, the link $i$ with highest value of $h_i(t)$, with ties being broken arbitrarily*[5].

Next, we discuss a few properties of the MAF policy that lead to the optimality result in Theorem 2.7.

**Remark 2.3.** *The MAF policy switches scheduling decisions only after a successful packet transmission.*

[5]Unless stated otherwise, we assume that ties are broken in favor of the link with the lowest index $i$.

In slot $t$, the MAF policy selects link $j = \arg\max_i\{h_i(t)\}$. Assume that this packet transmission fails. Then, in the next slot, the MAF policy selects $\arg\max_i\{h_i(t+1)\} = \arg\max_i\{h_i(t)+1\}$ which is again $j$. It is easy to see that the MAF policy will select the same link $j$, uninterruptedly, until the corresponding packet is successfully delivered.

**Remark 2.4** (Round Robin)**.** *Without loss of generality, reorder the index i in descending order of* $\vec{h}(1)$*, with link* 1 *having the highest* $h_i(1)$ *and link N the lowest* $h_i(t)$*. The MAF policy delivers packets according to the index sequence* $(1,2,\cdots,N,1,2,\cdots)$ *until the end of the time-horizon T, i.e. MAF follows a Round Robin pattern.*

Remark 2.4 follows directly from Remark 2.3 and it provides a complete description of the behavior of MAF.

**Remark 2.5** (Steady-State of MAF for error-free channels)**.** *Consider a wireless network with error-free channels,* $p_i = 1, \forall i$*. The MAF policy drives this network to a steady-state in which the sum of the elements of* $\vec{h}(t)$ *is constant and given by*

$$\sum_{i=1}^{N} h_i(t) = 1+2+\cdots+N = \frac{N(N+1)}{2}, \forall t \geq N+1\,. \tag{2.27}$$

Remark 2.5 follows directly from Remark 2.4. Figure 2-3 illustrates a network employing the MAF policy. It is easy to see that the steady-state is achieved at the beginning of slot $N+1$ and that the sum in (2.27) is independent of the initial AoI vector $\vec{h}(1)$.

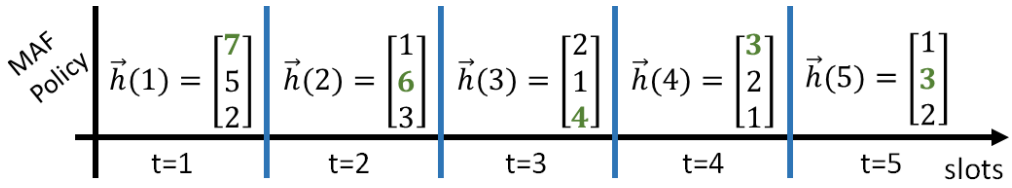



Figure 2-3: Evolution of $\vec{h}(t)$ when the MAF policy is employed in a network with $N = 3$ nodes, error-free channels, $p_i = 1, \forall i$, and $\vec{h}(1) = [7,5,2]^T$. In each slot, the MAF policy selects the link with highest $h_i(t)$. The selected link is represented in bold green. All elements in $\vec{h}(t)$ change according to (2.3): green elements are updated to 1 while black elements are incremented by 1. In this figure, the Round Robin pattern is evident.

In Theorem 2.7, we establish that MAF is AoI-optimal when the underlying network is *symmetric*, namely all links have the same channel reliability $p_i = p \in (0,1]$ and weight $w_i = w \geq 0$. Prior to the main result, we establish in Lemma 2.6 that MAF is AoI-optimal for a symmetric network with error-free channels.

**Lemma 2.6** (Optimality of MAF for error-free channels)**.** *Consider a symmetric wireless network with error-free channels $p_i = 1$ and weights $w_i = w > 0, \forall i$. Among the class of admissible policies $\Pi$, the MAF policy attains the minimum sum AoI for any given time-horizon $T$, namely*

$$J_T^{MAF} \leq J_T^{\pi} = \frac{w}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} h_i^{\pi}(t), \forall \pi \in \Pi, \forall T \geq 1 \, . \tag{2.28}$$

The complete proof of Lemma 2.6 is in [51, Appendix B]. Intuitively, MAF minimizes $\sum_{i=1}^{N} h_i(t)$ at every slot $t$ by reducing the highest element of $\vec{h}(t)$ to *unity*. Together, Lemma 2.6 and Remark 2.5 show that, when channels are error-free, MAF drives the network to a steady-state that is AoI-optimal. Next, we employ Lemma 2.6 to show that the MAF policy is AoI-optimal for any symmetric network.

**Theorem 2.7** (Optimality of MAF)**.** *Consider a symmetric wireless network with channel reliabilities $p_i = p \in (0,1]$ and weights $w_i = w > 0, \forall i$. Among the class of admissible policies $\Pi$, the MAF policy attains the minimum expected sum AoI for any given time-horizon $T$, namely*

$$\mathbb{E}\left[J_T^{MAF}\right] \leq \mathbb{E}\left[J_T^{\pi}\right] = \frac{w}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{E}\left[h_i^{\pi}(t)\right], \forall \pi \in \Pi, \forall T \geq 1 \, . \tag{2.29}$$

*Proof.* To show that the MAF policy minimizes the EWSAoI of any symmetric wireless network, we generalize Lemma 2.6 using a *stochastic ordering* argument [96] that compares the evolution of $\vec{h}(t)$ when MAF is employed to that when an arbitrary policy $\pi$

is employed. For the sake of simplicity and without loss of optimality, in this proof we assume that $\pi$ is work-conserving. There is no loss of optimality since for every non work-conserving policy, there is at least one work-conserving policy that is strictly dominant.

Denote the random variable that represents the sum of the elements of $\vec{h}(t)$ when $\pi$ is employed by $SH_t^\pi = \sum_{i=1}^N h_i^\pi(t)$. Using this notation and the symmetry assumptions of Theorem 2.7, the AoI optimization for a finite time-horizon $T$ in (2.5a) becomes

$$\mathbb{E}[J_T^*] = \frac{1}{TN} \min_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} w\, h_i^\pi(t)\right] = \frac{w}{TN} \min_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=1}^{T} SH_t^\pi\right]. \tag{2.30}$$

Next, we introduce the concept of stochastic ordering. Denote the stochastic process associated with the sequence $\{SH_t^\pi\}_{t=1}^T$ as $SH^\pi$ and its sample path as $sh^\pi$. Let $\mathbb{D}$ be the space of all sample paths $sh^\pi$. Define by $\mathscr{F}$ the set of measurable functions $f : \mathbb{D} \to \mathbb{R}^+$ such that $f(sh^{MAF}) \leq f(sh^\pi)$ for every $sh^{MAF}, sh^\pi \in \mathbb{D}$ which satisfy $sh_t^{MAF} \leq sh_t^\pi, \forall t$.

**Definition 2.8.** *(Stochastic ordering) We say that $SH^{MAF}$ is stochastically smaller than $SH^\pi$ and write $SH^{MAF} \leq_{st} SH^\pi$ if $\mathbb{P}\{f(SH^{MAF}) > z\} \leq \mathbb{P}\{f(SH^\pi) > z\}, \forall z \in \mathbb{R}, \forall f \in \mathscr{F}$.*

Since $f(SH^\pi)$ is positive, $SH^{MAF} \leq_{st} SH^\pi$ implies[6] $\mathbb{E}[f(SH^{MAF})] \leq \mathbb{E}[f(SH^\pi)], \forall f \in \mathscr{F}$. Knowing that one function that satisfies the conditions in $\mathscr{F}$ is $f(SH^\pi) = \sum_{t=1}^T SH_t^\pi$, it follows that if $SH^{MAF} \leq_{st} SH^\pi, \forall \pi \in \Pi$, then $\mathbb{E}[\sum_{t=1}^T SH_t^{MAF}] \leq \mathbb{E}[\sum_{t=1}^T SH_t^\pi], \forall \pi \in \Pi$, which is equivalent to the EWSAoI minimization in (2.30). Therefore, for establishing the optimality of $MAF$, it is sufficient to establish that $SH^{MAF}$ is stochastically smaller than $SH^\pi, \forall \pi \in \Pi$.

Stochastic ordering can be demonstrated using its definition directly. However, this is often complex for it involves comparing the probability distributions of $SH^{MAF}$ and $SH^\pi$. Instead, we use the following result from [96], which is also used in works such as [13, 24, 90]: for verifying that $SH^{MAF} \leq_{st} SH^\pi$, it is sufficient to show that there exists two stochastic processes $\widetilde{SH}^{MAF}$ and $\widetilde{SH}^\pi$ such that

[6] Recall that for any positive valued $X$, it follows that $\mathbb{E}[X] = \int_{x=0}^\infty (1 - \mathbb{P}\{X \leq x\})dx = \int_{x=0}^\infty \mathbb{P}\{X > x\}dx$.

(i) $SH^{\pi}$ and $\widetilde{SH}^{\pi}$ have the same probability distribution;

(ii) $\widetilde{SH}^{MAF}$ and $\widetilde{SH}^{\pi}$ are on a common probability space;

(iii) $SH^{MAF}$ and $\widetilde{SH}^{MAF}$ have the same probability distribution; and

(iv) $\widetilde{SH}_t^{MAF} \leq \widetilde{SH}_t^{\pi}$, with probability 1, $\forall t$.

This result allows us to establish stochastic ordering between $SH^{MAF}$ and $SH^{\pi}$ by properly designing the auxiliary processes $\widetilde{SH}^{MAF}$ and $\widetilde{SH}^{\pi}$. This design is achieved by utilizing stochastic coupling.

*Stochastic coupling* is a method utilized for comparing stochastic processes by imposing a common underlying probability space. We use stochastic coupling to construct $\widetilde{SH}^{\pi}$ and $\widetilde{SH}^{MAF}$ based on $SH^{\pi}$ and $SH^{MAF}$, respectively.

Let the process $\widetilde{SH}^{\pi}$ be identical to $SH^{\pi}$. Their (common) probability space is determined by the sequence of scheduling decisions $u_i(t)$ of policy $\pi$ and the sequence of channel states $c_i(t)$. Now, let us construct $\widetilde{SH}^{MAF}$ on the same probability space as $\widetilde{SH}^{\pi}$. For that, we couple $\widetilde{SH}^{MAF}$ to $\widetilde{SH}^{\pi}$ by dynamically connecting the channel state of MAF to the channel state of policy $\pi$ as follows. Suppose that in slot $t$, policy $\pi$ schedules link $j$ while MAF schedules link $i$, then, for the duration of that slot, we assign $c_i(t) \leftarrow c_j(t)$. As a result, if policy $\pi$'s transmission was successful, i.e. $c_j(t) = 1$, then we impose that MAF's transmission is also successful, i.e. $c_i(t) = 1$. This dynamic assignment imposes that, at every slot $t \in \{1, 2, \cdots, T\}$, the channel state of MAF is identical to the channel state of $\pi$. Notice that this assignment is only possible because the random variable $c_i(t)$ is i.i.d. with respect to the links and slots, which is the same reason for $\widetilde{SH}^{MAF}$ and $SH^{MAF}$ having the same probability distribution.

Returning to our four conditions, it follows directly from the coupling method described above that (i), (ii) and (iii) are satisfied. Thus, the only condition that remains to be shown is

$$\text{(iv)}\ \ \widetilde{SH}_t^{MAF} \leq \widetilde{SH}_t^{\pi}, \text{with probability 1, } \forall t.$$

Coupling between $\widetilde{SH}^{\pi}$ and $\widetilde{SH}^{MAF}$ is the key property to establish (iv). Assume that policy $\pi$ is employed and consider a sample path $\widetilde{sh}^{\pi}$ spanning the entire time-horizon.

Use the sequence of channel states from the links selected by $\pi$ during the evolution of $\widetilde{sh}^{\pi}$ to create the coupled sample path $\widetilde{sh}^{MAF}$, as illustrated in Figure 2-4. Notice that the scheduling decisions taken during slots in which the channel state is OFF cannot change the relationship ($\leq$ or $\geq$) between $\widetilde{sh}_t^{\pi}$ and $\widetilde{sh}_t^{MAF}$. As a result, they can be removed from the analysis and we can focus on slots with *error-free channels*. Lemma 2.6 established that, in a network with error-free channels, we have $\widetilde{sh}_t^{MAF} \leq \widetilde{sh}_t^{\pi}$, for every slot $t$ and for every policy $\pi \in \Pi$. Since this argument follows for every sample path $\widetilde{sh}^{\pi}$, we establish condition (iv) and the stochastic ordering argument. ■

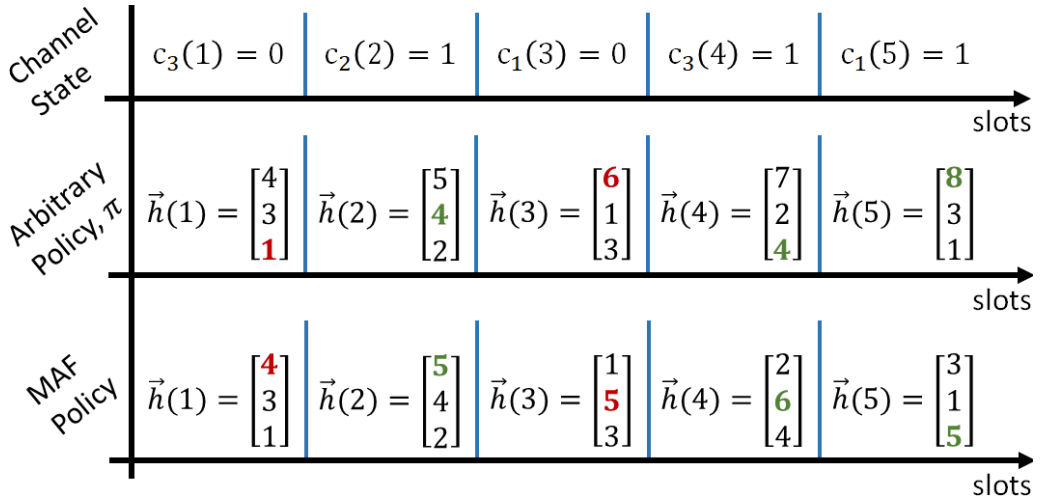



Figure 2-4: Evolution of $\widetilde{sh}^{\pi}$ and $\widetilde{sh}^{MAF}$ for a network with $N = 3$ nodes, $T = 5$ slots, initial AoI $\vec{h}(1) = [4,3,1]^T$ and unreliable channels. Recall that a channel is ON when $c_i(t) = 1$ and OFF when $c_i(t) = 0$. Successful transmissions are represented in green and failed transmissions in red. On the top, channel states associated with the sequence of scheduling decisions of the arbitrary policy $\pi$. Notice that, due to coupling, the MAF policy has identical channel states. On the middle, the evolution of $\vec{h}(t)$ when policy $\pi$ is employed. On the bottom, the evolution of $\vec{h}(t)$ when MAF is employed. Comparing the sum of $\vec{h}(t)$ over time for both policies, we have $\widetilde{sh}^{\pi} = \{8;11;10;13;12\}$ and $\widetilde{sh}^{MAF} = \{8;11;9;12;9\}$, and we see that $\widetilde{sh}_t^{MAF} \leq \widetilde{sh}_t^{\pi}, \forall t$.

Theorem 2.7 establishes that the MAF policy, which selects link $j = \arg\max_i \{h_i(t)\}$ in every slot $t$, is AoI-optimal when the network is symmetric. For general networks, with links possibly having different channel reliabilities $p_i$ and weights $w_i$, scheduling decisions based exclusively on $\vec{h}(t)$ may not be AoI-optimal.

Next, we derive a closed-form expression for the performance guarantee of MAF $\rho^{MAF}$. The expression for $\rho^{MAF}$ depends on the statistics of the set of values $\{1/p_i\}_{i=1}^N$, in particular of its *coefficient of variation*. Let the sample mean and sample variance of $\{1/p_i\}_{i=1}^N$ be

$$\bar{\mathbb{M}}\left[\frac{1}{p_i}\right] = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{p_j} \quad \text{and} \quad \bar{\mathbb{V}}\left[\frac{1}{p_i}\right] = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{p_j} - \bar{\mathbb{M}}\left[\frac{1}{p_i}\right]\right)^2. \tag{2.31}$$

Then, the coefficient of variation is given by

$$C_V = \frac{\sqrt{\bar{\mathbb{V}}\left[\frac{1}{p_i}\right]}}{\bar{\mathbb{M}}\left[\frac{1}{p_i}\right]}. \tag{2.32}$$

The coefficient of variation is a measure of how spread out are the values of $1/p_i$. The value of $C_V$ is large when $\{1/p_i\}_{i=1}^N$ are disperse and $C_V = 0$ if and only if $p_i = p$ for all links.

**Theorem 2.9** (Performance of MAF policy)**.** *Consider a wireless network with parameters $(N, p_i, w_i)$ and an infinite time-horizon. The MAF policy is $\rho^{MAF}$-optimal, where*

$$\rho^{MAF} = \frac{\left(\frac{N+1+C_V^2}{2}\right)\bar{\mathbb{M}}\left[\frac{1}{p_j}\right]\bar{\mathbb{M}}[w_i]}{\frac{N}{2}\left(\bar{\mathbb{M}}\left[\sqrt{\frac{w_i}{p_i}}\right]\right)^2 + \frac{1}{2}\bar{\mathbb{M}}[w_i]}. \tag{2.33}$$

*Proof.* The performance guarantee for MAF is given by $\rho^{MAF} = \mathbb{E}[J^{MAF}]/L_B$, where the denominator is the universal lower bound in (2.15) and the numerator is the infinite-horizon objective function $\mathbb{E}[J^{MAF}] = \lim_{T\to\infty}\mathbb{E}[J_T^{MAF}]$ which is derived next.

To analyze the evolution of $h_i(t)$ when the MAF policy is employed, we assume for this proof (without loss of generality) that the link index $i$ is in descending order of $\vec{h}(1)$, with link 1 having the highest $h_i(1)$ and link $N$ the lowest $h_i(t)$, as in Remark 2.4. Then, the

properties in Remarks 2.3 and 2.4 can be summarized as follows:

(i) MAF *selects* the same link $i$, uninterruptedly, until the corresponding packet is successfully delivered; and

(ii) MAF *delivers* packets according to the index sequence $(1,2,\cdots,N,1,2,\cdots)$ until the end of the time-horizon $T$.

Based on property (i), define $X_i[m]$ as the number of successive transmission attempts by link $i$ that precede the $m$th packet delivery to the same link, with $X_i[0]=0,\forall i$. For a given $i$, the random variables $X_i[m]$ are i.i.d. with geometric distribution. Moreover, transmissions to different links are independent. Hence, we have

$$\mathbb{E}[X_i[m]] = \frac{1}{p_i} \tag{2.34a}$$

$$\mathbb{E}[X_i[m]X_j[m-1]] = \frac{1}{p_i p_j} \tag{2.34b}$$

$$\mathbb{E}\left[X_i^2[m]\right] = \frac{2-p_i}{p_i^2}\,. \tag{2.34c}$$

According to property (ii), packets are delivered following a Round Robin pattern. Thus, as illustrated in Fig. 2-5, the total number of slots in the interval between the $(m-1)$th and $m$th packet deliveries by a target link $i$ *when the MAF policy is employed* can be written as

$$I_i[m] = \sum_{j=i+1}^{N} X_j[m-1] + \sum_{j=1}^{i} X_j[m]\,. \tag{2.35}$$

By employing (2.34a)-(2.34c), the first and second moments of $I_i[m]$ can be expressed as

$$\mathbb{E}[I_i[m]] = \sum_{i=1}^{N} \frac{1}{p_i}\,; \tag{2.36a}$$

$$\mathbb{E}\left[I_i^2[m]\right] = \sum_{j=1}^{N} \frac{2-p_j}{p_j^2} + 2\sum_{j=1}^{N}\sum_{k=j+1}^{N} \frac{1}{p_j p_k}\,. \tag{2.36b}$$

Evidently, when the MAF policy is employed, the sequence of packet deliveries via link $i$ is a renewal process with i.i.d. inter-delivery times $I_i[m]$. Therefore, using the generalization of the elementary renewal theorem for renewal-reward processes [23, Sec. 5.7], we

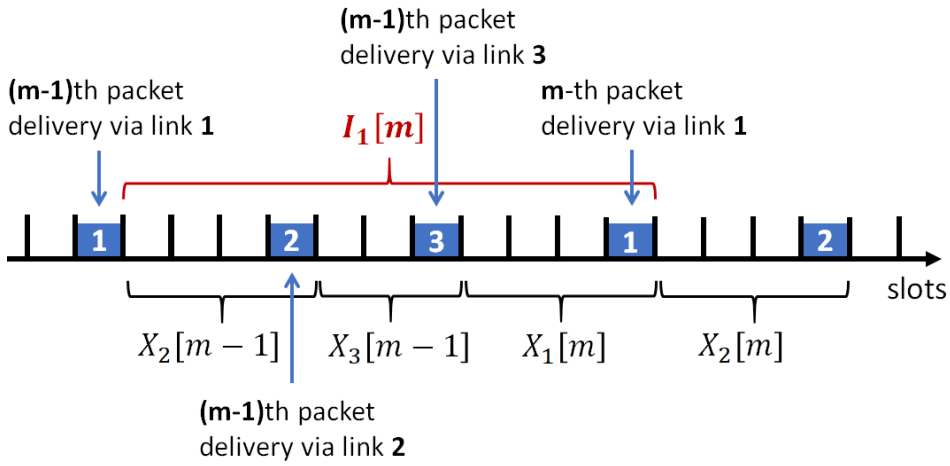



Figure 2-5: Illustration of the time period between the (m-1)th and mth packet deliveries by link $i$ when the MAF policy is employed in a network with $N = 3$ nodes. A blue box with index $j$ represents a packet delivery by link $j$. Notice the Round Robin pattern described in Remark 2.4.

have

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[h_i^{MAF}(t)] = \frac{\mathbb{E}[reward]}{\mathbb{E}[interval]} = \frac{\mathbb{E}[1+2+\cdots+I_i[m]]}{\mathbb{E}[I_i[m]]} = \frac{\mathbb{E}[I_i[m]^2]}{2\mathbb{E}[I_i[m]]} + \frac{1}{2}\,. \tag{2.37}$$

Applying (2.36a)-(2.36b) into (2.37) yields:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[h_i^{MAF}(t)] = \frac{N\left(\frac{1}{N}\sum_{j=1}^{N}\frac{1}{p_j}\right)^2 + \frac{1}{N}\sum_{j=1}^{N}\frac{1}{p_j^2}}{\frac{2}{N}\sum_{i=1}^{N}\frac{1}{p_i}}\,, \tag{2.38}$$

and then substituting (2.31) and (2.32), we obtain

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[h_i^{MAF}(t)] = \frac{1}{2}\bar{\mathbb{M}}\left[\frac{1}{p_j}\right]\left(N+1+C_V^2\right)\,. \tag{2.39}$$

Employing (2.39) into the objective function in (2.5a), we get

$$\mathbb{E}[J^{MAF}] = \frac{1}{N}\sum_{i=1}^{N} w_i \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[h_i^{MAF}(t)\right] = \left(\frac{N+1+C_V^2}{2}\right)\bar{\mathbb{M}}\left[\frac{1}{p_j}\right]\bar{\mathbb{M}}[w_i]\,. \tag{2.40}$$

Dividing (2.40) by the lower bound in (2.15) yields the performance guarantee $\rho^{MAF}$ in (2.33). ■

Next, we use the performance guarantee in (2.33) to obtain the necessary and sufficient conditions for the optimality of the MAF policy when $N \to \infty$.

**Corollary 2.10.** *In the limit as $N \to \infty$, the performance guarantee of MAF $\rho^{MAF} \to 1$ if and only if the network is symmetric.*

*Proof.* In the limit as $N \to \infty$, the performance guarantee of MAF can be written as

$$\rho^{MAF} = \frac{\left(\frac{N+C_V^2}{2}\right) \bar{\mathbb{M}}\left[\frac{1}{p_j}\right] \bar{\mathbb{M}}[w_i]}{\frac{N}{2}\left(\bar{\mathbb{M}}\left[\sqrt{\frac{w_i}{p_i}}\right]\right)^2}. \tag{2.41}$$

Consider two inequalities: (i) Cauchy-Schwarz

$$\left(\bar{\mathbb{M}}\left[\sqrt{\frac{w_i}{p_i}}\right]\right)^2 \leq \bar{\mathbb{M}}\left[\frac{1}{p_j}\right] \bar{\mathbb{M}}[w_i]\ ; \tag{2.42}$$

and (ii) Positive coefficient of variation: $C_V \geq 0$. It is evident from (2.41) that $\rho^{MAF} = 1$ if and only if both inequalities (i) and (ii) hold with *equality* and this is true if and only if $w_i = w$ and $p_i = p$ for all links. ■

The two main results in this section are Theorem 2.7, which established that MAF is AoI-optimal when the network is symmetric, and Theorem 2.9, which characterized the performance guarantee of MAF for general networks. Corollary 2.10 shows that (as expected) when $N \to \infty$ and the network is symmetric, we have $\rho^{MAF} \to 1$. Notice that $\rho^{MAF} \to 1$ implies that $\mathbb{E}[J^{MAF}] \to L_B$, *which suggests that the lower bound $L_B$ is tight.*

By leveraging the knowledge of $h_i(t)$, but disregarding the values of $w_i$ and $p_i$, the MAF policy achieves optimal AoI performance in symmetric networks. In the next section, we discuss a class of scheduling policies that use the knowledge of $w_i$ and $p_i$, but neglect $h_i(t)$.

### 2.2.3 Stationary Randomized Policy

Consider the class of Stationary Randomized policies in which scheduling decisions are made randomly, according to *fixed probabilities* $\beta_i / \sum_{j=1}^{N} \beta_j$ for positive values of $\{\beta_i\}_{i=1}^{N}$.

**Definition 2.11** (Randomized policy)**.** *The Randomized policy selects, in each slot $t$, link $i$ with probability $\beta_i / \sum_{j=1}^{N} \beta_j$, for every link $i$.*

Denote the Randomized policy as $R$. Observe that this simple policy is stationary and it does not use information about the current AoI $h_i(t)$. We assume that $R$ has knowledge[7] of $(N, p_i, w_i)$ and may use these parameters to tune $\beta_i$. Next, we derive a closed-form expression for the performance guarantee $\rho^R$ and find a Randomized policy that is 2-optimal *for all network configurations* $(N, p_i, w_i)$.

**Theorem 2.12** (Performance of Randomized policy)**.** *Consider a wireless network with parameters $(N, p_i, w_i)$ and an infinite time-horizon. The Randomized policy with positive values of $\{\beta_i\}_{i=1}^{N}$ is $\rho^R$-optimal, where*

$$\rho^R = 2 \, \frac{\bar{\mathbb{M}}\left[\beta_i\right] \bar{\mathbb{M}}\left[\dfrac{w_i}{p_i \beta_i}\right]}{\left(\bar{\mathbb{M}}\left[\sqrt{\dfrac{w_i}{p_i}}\right]\right)^2 + \dfrac{1}{N} \bar{\mathbb{M}}\left[w_i\right]} \, . \tag{2.43}$$

*Proof.* The performance guarantee is defined as $\rho^R = \mathbb{E}[J^R]/L_B$, where the denominator is the universal lower bound in (2.15) and the numerator is the infinite-horizon objective function $\mathbb{E}[J^R] = \lim_{T \to \infty} \mathbb{E}[J_T^R]$ which is derived next.

Recall that $u_i(t)$ indicates that link $i$ was selected in slot $t$ and $d_i(t) = c_i(t) u_i(t)$ indicates a successful packet transmission in the same slot. When the Randomized policy is

[7] The assumption of fixed and known channel reliabilities $\{p_i\}_{i=1}^{N}$ is used in chapters 2, 3 and 4. This assumption is not used in chapter 5 when the deployed system learns $p_i$ over time.

employed, we have $\mathbb{E}[u_i(t)] = \beta_i / \sum_{j=1}^{N} \beta_j, \forall t$ and

$$\mathbb{E}\left[d_i(t)\right] = \mathbb{E}\left[d_i\right] = \frac{\beta_i}{\sum_{j=1}^{N} \beta_j} p_i, \forall t \ . \tag{2.44}$$

Clearly, the sequence of packet deliveries by link $i$ is a renewal process with geometric inter-delivery times $I_i[m]$ with mean $(\mathbb{E}[d_i])^{-1}$. Thus, using the generalization of the elementary renewal theorem for renewal-reward processes [23, Sec. 5.7] yields

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[h_i^R(t)] = \frac{\mathbb{E}[I_i[m]^2]}{2\mathbb{E}[I_i[m]]} + \frac{1}{2} = \frac{1}{\mathbb{E}[d_i]} \ , \tag{2.45}$$

and substituting (2.44) and (2.45) into the objective function in (2.5a) gives

$$\mathbb{E}\left[J^R\right] = \frac{1}{N} \sum_{i=1}^{N} \frac{w_i}{\mathbb{E}\left[d_i\right]} = \frac{1}{N} \sum_{j=1}^{N} \beta_j \sum_{i=1}^{N} \frac{w_i}{p_i \beta_i} = N \bar{\mathbb{M}}\left[\beta_i\right] \bar{\mathbb{M}}\left[\frac{w_i}{p_i \beta_i}\right] \ . \tag{2.46}$$

Finally, dividing (2.46) by the lower bound in (2.15) gives $\rho^R$ in (2.43). ■

**Corollary 2.13.** *The Stationary Randomized policy with $\beta_i = \sqrt{w_i/p_i}, \forall i$ has $\rho^R < 2$ for all network configurations $(N, p_i, w_i)$.*

*Proof.* The assignment $\beta_i = \sqrt{w_i/p_i}, \forall i \in \{1, \cdots, N\}$ is the necessary condition for the Cauchy-Schwarz inequality

$$\left(\bar{\mathbb{M}}\left[\sqrt{\frac{w_i}{p_i}}\right]\right)^2 \leq \bar{\mathbb{M}}\left[\beta_i\right] \bar{\mathbb{M}}\left[\frac{w_i}{p_i \beta_i}\right] \ , \tag{2.47}$$

to hold with equality. Applying this condition to (2.43) results in $\rho^R < 2$. Notice that $\beta_i = \sqrt{w_i/p_i}$ minimizes the RHS of (2.47) and the expression in (2.46). ■

Theorem 2.12 gives a closed-form expression for $\rho^R$ and Corollary 2.13 shows that, by using only the knowledge of $w_i$ and $p_i$, a Randomized policy can *achieve 2-optimal performance in all network setups* $(N, p_i, w_i)$. Next, we develop a Max-Weight policy that leverages the knowledge of $w_i$, $p_i$, and $h_i(t)$ in making scheduling decisions.

### 2.2.4 Max-Weight Policy

In this section, we use concepts from Lyapunov Optimization [85] to derive a Max-Weight policy. The Max-Weight policy is obtained by minimizing the drift of a Lyapunov Function of the network state at every slot $t$. Consider the linear Lyapunov Function

$$L(\vec{h}(t)) = \frac{1}{N}\sum_{i=1}^{N} \tilde{\alpha}_i h_i(t) \ , \tag{2.48}$$

where $\tilde{\alpha}_i > 0$ are auxiliary parameters used to tune the performance of the Max-Weight policy, and consider the one-slot Lyapunov Drift

$$\Delta(\vec{h}(t)) = \mathbb{E}\left[ L(\vec{h}(t+1)) - L(\vec{h}(t)) \middle| \vec{h}(t) \right] . \tag{2.49}$$

The Lyapunov Function $L(\vec{h}(t))$ depicts how large is the AoI of the network during slot $t$, while the Lyapunov Drift $\Delta(\vec{h}(t))$ represents the growth of $L(\vec{h}(t))$ from one slot to the next. Intuitively, by minimizing the drift, the Max-Weight policy reduces the value of $L(\vec{h}(t))$ and, consequently, keeps the AoI of the network low.

To find the policy that minimizes the one-slot drift $\Delta(\vec{h}(t))$, we first need to analyze the RHS of (2.49). Consider a scheduling policy $\pi$ making a scheduling decision during slot $t$ based on its knowledge of $\vec{h}(t)$ and $\tilde{\alpha}_i$. Recall that $d_i(t)$ indicates a packet delivery by link $i$ during slot $t$. An alternative way to represent the evolution of $h_i(t)$ defined in (2.3) is $h_i(t+1) = d_i(t) + (h_i(t)+1)[1-d_i(t)]$. Taking the conditional expectation of $h_i(t+1)$ yields

$$\mathbb{E}\left[ h_i(t+1) - h_i(t) | \vec{h}(t) \right] = -\mathbb{E}\left[ d_i(t) | \vec{h}(t) \right] h_i(t) + 1 \ . \tag{2.50}$$

Substituting (2.48) into (2.49) and then using (2.50) gives the following expression for the Lyapunov Drift

$$\Delta(\vec{h}(t)) = -\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}\left[ d_i(t) | \vec{h}(t) \right] \tilde{\alpha}_i h_i(t) + \frac{1}{N}\sum_{i=1}^{N} \tilde{\alpha}_i \ . \tag{2.51}$$

Recall that $\mathbb{E}\left[d_i(t)|\vec{h}(t)\right] = p_i\mathbb{E}\left[u_i(t)|\vec{h}(t)\right]$. Observe that the choice of $u_i(t)$ affects only the first term on the RHS of (2.51). During slot $t$, the scheduling policy that maximizes the sum $\sum_{i=1}^{N}\mathbb{E}\left[u_i(t)|\vec{h}(t)\right]p_i\tilde{\alpha}_i h_i(t)$ also minimizes $\Delta(\vec{h}(t))$. The Max-Weight policy minimizes the drift at every slot $t$. Denote the Max-Weight policy as $MW$.

**Definition 2.14** (Max-Weight policy)**.** *The MW policy selects, in each slot $t$, the link $i$ with highest value of $p_i\tilde{\alpha}_i h_i(t)$, with ties being broken arbitrarily.*

**Remark 2.15.** *The Max-Weight policy is AoI-optimal for symmetric wireless networks with parameters $w_i = w$, $p_i = p$ and $\tilde{\alpha}_i = \tilde{\alpha}, \forall i$.*

Observe that when the network is symmetric, prioritizing according to $p_i\tilde{\alpha}_i h_i(t)$ is identical to prioritizing according to $h_i(t)$, i.e. Max-Weight is identical to MAF. Thus, from Theorem 2.7 (Optimality of MAF), we conclude that Max-Weight is AoI-optimal.

**Theorem 2.16** (Performance of Max-Weight policy)**.** *Consider a wireless network with parameters $(N, p_i, w_i)$ and an infinite time-horizon. The Max-Weight policy with $\tilde{\alpha}_i = \sqrt{w_i/p_i}$ has $\rho^{MW} < 2$ for all network configurations $(N, p_i, w_i)$.*

*Proof.* The performance guarantee is defined as $\rho^{MW} = U_B^{MW}/L_B$, where the denominator is the universal lower bound in (2.15) and the numerator is an upper bound to the infinite-horizon objective function $U_B^{MW} \geq \lim_{T\to\infty}\mathbb{E}[J_T^{MW}]$ which is derived next by manipulating the expression of the one-slot Lyapunov drift in (2.51).

Recall that the Max-Weight policy minimizes $\Delta(\vec{h}(t))$ by choosing $u_i(t)$ such that the sum $\sum_{i=1}^{N}\mathbb{E}\left[d_i(t)|\vec{h}(t)\right]\tilde{\alpha}_i h_i(t)$ is maximized. Employing any other policy $\pi \in \Pi$ yields a lower (or equal) sum. Consider a Stationary Randomized policy as defined in Sec. 2.2.3. It chooses $u_i(t)$ randomly, independently of the value of $\vec{h}(t)$, and yields $\mathbb{E}\left[d_i(t)|\vec{h}(t)\right] =$

$\mathbb{E}[d_i]$. Substituting $\mathbb{E}[d_i]$ into the equation of the one-slot Lyapunov Drift gives

$$\Delta(\vec{h}(t)) \leq -\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[d_i]\,\tilde{\alpha}_i h_i(t) + \frac{1}{N}\sum_{i=1}^{N}\tilde{\alpha}_i\,. \tag{2.52}$$

Now, taking the expectation with respect to $\vec{h}(t)$ yields

$$\mathbb{E}\left[L(\vec{h}(t+1)) - L(\vec{h}(t))\right] \leq -\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[d_i]\,\tilde{\alpha}_i\mathbb{E}[h_i(t)] + \frac{1}{N}\sum_{i=1}^{N}\tilde{\alpha}_i\,. \tag{2.53}$$

Summing over $t \in \{1,2,\cdots,T\}$ and dividing by $T$ results in

$$\frac{\mathbb{E}\left[L(\vec{h}(T+1))\right]}{T} - \frac{\mathbb{E}\left[L(\vec{h}(1))\right]}{T} \leq -\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[d_i]\,\tilde{\alpha}_i\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[h_i(t)] + \frac{1}{N}\sum_{i=1}^{N}\tilde{\alpha}_i\,. \tag{2.54}$$

Manipulating the expression and knowing that $L(\vec{h}(T+1))$ is always positive gives

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[d_i]\,\tilde{\alpha}_i\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[h_i(t)] \leq \frac{\mathbb{E}\left[L(\vec{h}(1))\right]}{T} + \frac{1}{N}\sum_{i=1}^{N}\tilde{\alpha}_i\,. \tag{2.55}$$

Knowing that $L(\vec{h}(1))$ is finite, assigning $\tilde{\alpha}_i = w_i/\mathbb{E}[d_i]$ and taking the limit as $T \to \infty$, gives

$$\lim_{T\to\infty}\frac{1}{TN}\sum_{i=1}^{N}\sum_{t=1}^{T}w_i\mathbb{E}[h_i(t)] \leq \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{\mathbb{E}[d_i]}\,. \tag{2.56}$$

The LHS of (2.56) is the definition of EWSAoI for the Max-Weight policy and the RHS is the upper bound $U_B^{MW}$. Notice that the RHS is identical to the EWSAoI for the Stationary Randomized policy $\mathbb{E}\left[J^R\right]$ in (2.46). Hence, we can use Corollary 2.13 to conclude that $\rho^{MW} < 2$ for the Max-Weight policy with $\tilde{\alpha}_i = \sqrt{w_i/p_i}/(\sum_{j=1}^{N}\sqrt{w_j/p_j})$. Notice that the term $\sum_{j=1}^{N}\sqrt{w_j/p_j}$ is a constant that appears in every $\tilde{\alpha}_i$. Thus, it can be removed from $\tilde{\alpha}_i$ without affecting the Max-Weight policy. ■

The choice in (2.48) for a linear Lyapunov Function with auxiliary parameters $\tilde{\alpha}_i$ allowed us to obtain the performance guarantee $\rho^{MW} < 2$. Choosing a different Lyapunov

Function yields a different Max-Weight policy with a different performance guarantee. The standard choice of Lyapunov Function is quadratic [85, Chapter 3]. The quadratic Lyapunov Function is studied in [51, Sec. IV.D]. This choice results in a somewhat similar Max-Weight policy with a performance guarantee $\rho^{MW'} < 4$.

In contrast to the MAF and Randomized policies, the Max-Weight policy uses all available information, namely $\tilde{\alpha}_i = \sqrt{w_i/p_i}$ and $h_i(t)$, in making scheduling decisions. As expected, numerical results in Sec. 2.3 demonstrate that Max-Weight outperforms both MAF and Randomized in every network setup simulated. In fact, the performance of Max-Weight is comparable to the universal lower bound $L_B$.

However, by comparing the performance guarantees derived for each policy, namely $\rho^{MAF}$, $\rho^{R}$, and $\rho^{MW}$, it might seem that Max-Weight does not provide better performance. The reason for this is the challenge to obtain a tight upper bound $U_B^{MW}$ for Max-Weight. As opposed to MAF and Randomized, the performance of Max-Weight cannot be evaluated using Renewal Theory and does not have properties that simplify the analysis, such as packets being delivered following a Round Robin pattern or links being selected according to fixed probabilities.

Next, we consider the AoI minimization problem from a different perspective and propose an Index policy [110], also known as Whittle's Index policy. This policy is surprisingly similar to the Max-Weight policy and also yields a strong performance.

### 2.2.5 Whittle's Index Policy

Whittle's Index policy is the optimal solution to a relaxation of the Restless Multi-Armed Bandit (RMAB) problem. This low-complexity heuristic policy has been extensively used in the literature [78,90,93] and is known to have a strong performance in a range of applications [73, 109]. The challenge associated with this approach is that the Index policy is only defined for problems that are *indexable*, a condition which is often difficult to establish. A detailed introduction to the relaxed RMAB problem can be found in [25, 110].

To find the Whittle's Index, we transform the AoI optimization (2.6a)-(2.6b) into a relaxed Restless Multi-Armed Bandit (RMAB) problem. This is possible because the AoI associated with each link in the network evolves as a restless bandit. To obtain the re-

laxed RMAB problem, we first substitute the $T$ interference constraints $\sum_{i=1}^{N} u_i(t) \leq 1, \forall t$ in (2.6b) by the single time-averaged constraint

$$\frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{E}[u_i(t)] \leq \frac{1}{N} \,. \tag{2.57}$$

Then, we relax this time-averaged constraint, by placing (2.57) into the objective function (2.6a) together with the associated Lagrange Multiplier $C \geq 0$ and, as a result, we obtain the relaxed RMAB problem.

**Relaxed RMAB problem**

$$\min_{\pi \in \Pi} \left\{ \lim_{T \to \infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \left( w_i \mathbb{E}\left[h_i^{\pi}(t)\right] + C\, \mathbb{E}\left[u_i^{\pi}(t)\right] \right) \right\} - \frac{C}{N} \,, \tag{2.58a}$$

$$\text{s.t. } C \geq 0 \,. \tag{2.58b}$$

Notice that the relaxed RMAB problem is separable and thus can be solved for each individual link. The *relaxed RMAB problem associated with each link is called the Decoupled Model*. The solution to the Decoupled Model lays the foundation for the design of the Index policy. Next, we solve the Decoupled Model, establish that the relaxed AoI optimization is indexable and obtain a closed-form expression for the Whittle's Index.

## Decoupled Model

The Decoupled Model adheres to the network model in Sec. 2.1 with $N = 1$, except for the addition of a *service charge*. The service charge is a fixed cost per transmission $C \geq 0$ that is incurred by the network every time the BS selects a link, i.e. $u(t) = 1$. Observe that a scheduling policy running on a network with a *single link* can only choose between selecting this link or idling.

**Decoupled Model**

$$\min_{\pi \in \Pi} \left\{ \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \left( w \mathbb{E}\left[h^{\pi}(t)\right] + C\, \mathbb{E}\left[u^{\pi}(t)\right] \right) \right\} \,, \tag{2.59a}$$

$$\text{s.t. } C \geq 0 \,. \tag{2.59b}$$

The Decoupled Model is formulated as a Dynamic Program (DP). The components of the DP are: the *network state* $h(t)$, the *control variable* $u(t)$, the *state transitions* and the *cost function*. The *state transitions* are divided in two cases: i) when the scheduling policy transmits, namely $u(t) = 1$, then the state transition from slot $t$ to slot $t+1$ depends on the channel condition as follows

$$\mathbb{P}\left(h(t+1) = h(t)+1 | u(t) = 1, h(t)\right) = 1-p\,; \quad \text{[channel OFF]} \tag{2.60a}$$

$$\mathbb{P}\left(h(t+1) = 1 | u(t) = 1, h(t)\right) = p\,, \quad \text{[channel ON]} \tag{2.60b}$$

and ii) when the scheduling policy idles, namely $u(t) = 0$, then the state transition is deterministic

$$\mathbb{P}\left(h(t+1) = h(t)+1 | u(t) = 0, h(t)\right) = 1\,. \tag{2.61}$$

The *cost function* at the transition from slot $t$ to slot $t+1$ is given by

$$g_t(h(t), u(t)) = wh(t) + Cu(t)\,. \tag{2.62}$$

With the four components of the DP formulation described, next we present Bellman equations and the differential cost-to-go function.

Consider the *Decoupled Model in steady-state* with network state $h$ and control variable $u$. Then, Bellman equations are given by

$$S(1) = 0 \quad \text{and} \tag{2.63a}$$

$$S(h) + \lambda = \min_{u\in\{0,1\}} \left\{ wh + S(h+1)\,;\; wh + C + (1-p)S(h+1) + pS(1) \right\}, \tag{2.63b}$$

for all $h \in \{1, 2, \cdots\}$, where $\lambda$ is the *optimal average cost* and $S(h)$ is the *differential cost-to-go function*. Notice that the first part of the minimization in (2.63b) is associated with idling, i.e. choosing $u = 0$, and the second part is associated with transmitting, i.e. choosing $u = 1$, with ties being broken in favor to idling.

The stationary scheduling policy that solves Bellman equations and the Decoupled

Model[8] is given in Proposition 2.17.

**Proposition 2.17** (Threshold policy)**.** *The stationary scheduling policy that solves Bellman equations* (2.63a)-(2.63b) *is a threshold policy that, in each slot $t$, transmits when $h > H-1$ and idles when $1 \leq h \leq H-1$. For positive fixed values of $C > 0$, the expression for the threshold is*

$$H = \left\lfloor \frac{3}{2} - \frac{1}{p} + \sqrt{\left(\frac{1}{p} - \frac{1}{2}\right)^2 + \frac{2C}{wp}} \right\rfloor . \tag{2.64}$$

The complete proof of Proposition 2.17 is provided in Appendix 2.A. Intuitively, we expect that the optimal scheduling decision is to idle during slots in which $h$ is low (avoiding the service charge $C$) and to transmit when $h$ is high (attempting to reduce the value of $h$). Moreover, if the optimal decision is to transmit when the state is $h = H$, for a given $H$, then it is natural to expect that for all $h > H$ the optimal decision is also to transmit, which is characteristic of threshold policies.

The outline of the proof in Appendix 2.A follows: 1) assume that the optimal policy is a threshold policy with an unknown $H$; 2) using this assumption, find a solution to Bellman equations (2.63a)-(2.63b); 3) show that the solution and the assumption are consistent; and 4) obtain the threshold $H$ in (2.64), which is the minimum integer $H$ for which the optimal decision is to transmit. Next, we define the condition for indexability and establish that the relaxed AoI optimization is indexable.

## Indexability and Closed-form Index

Consider the Decoupled Model and its associated optimal threshold policy. For a given value of $C$, let $\mathscr{I}(C) = \{h \in \mathbb{N} | h < H\}$ be the set of states $h$ in which the threshold policy

[8]The Decoupled Model is an Expected Average Cost problem over an infinite-horizon and with *countably infinite state space*. In general, these problems are challenging to address. For the Decoupled Model in (2.59a)-(2.59b), it can be shown that [12, Proposition 5.6.1] is satisfied under some additional conditions on $\Pi$. The results in [12, Proposition 5.6.1] and Proposition 2.17 are sufficient to establish the optimality of the stationary scheduling policy.

*idles*. The definition of indexability is given next.

**Definition 2.18** (Indexability)**.** *The Decoupled Model is indexable if the set* $\mathscr{I}(C)$ *increases monotonically from* $\emptyset$ *to* $\mathbb{N}$*, as the value of* $C$ *increases from* $0$ *to* $+\infty$*. The AoI optimization* (2.5a) *with a relaxed interference constraint* (2.57) *is indexable if the Decoupled Model is indexable for all links* $i$*.*

The indexability of the Decoupled Model follows directly from the expression of $H$ in (2.64). Clearly, the threshold $H$ is monotonically increasing with $C$. Substituting $C = 0$ yields $H = 1$, which implies $\mathscr{I}(C) = \emptyset$, and the limit $C \to +\infty$ gives $H \to +\infty$, which implies $\mathscr{I}(C) = \mathbb{N}$. Since this is true for the Decoupled Model associated with every link $i$, we conclude that the relaxed AoI optimization is indexable. Given indexability, we define Whittle's Index next.

**Definition 2.19** (Index)**.** *Denote by* $C(h)$ *the Whittle's Index in state* $h$*. Then,* $C(h)$ *is the* infimum value of $\mathbf{C}$ *that makes both scheduling decisions (transmit or idle) equally desirable to the optimal policy in state* $h$*.*

Consider Proposition 2.17. For both scheduling decisions to be equally desirable in state $h$, the threshold should be $H = h + 1$. Substituting (2.64) we can solve this equation for $C$ which gives the following closed-form expression for the Index

$$C(h) = \frac{wph}{2}\left[h + \frac{2}{p} - 1\right]. \tag{2.65}$$

## Whittle's Index Policy

After establishing indexability and obtaining the expression for the Index, *we return to our original problem in Sec. 2.1*, with the BS scheduling links $i \in \{1,2,\cdots,N\}$ over time. We denote the Whittle's Index policy for the wireless network as $WI$.

**Definition 2.20** (Whittle's Index policy)**.** *The $WI$ policy selects, in each slot $t$, the link $i$ with highest value of*

$$C_i(h_i(t)) = \frac{w_i p_i h_i(t)}{2}\left[h_i(t) + \frac{2}{p_i} - 1\right], \tag{2.66}$$

*with ties being broken arbitrarily.*

Notice that in the original problem in Sec. 2.1, *no service charge is incurred by the network* for scheduling packet transmissions. The index $C_i(h_i(t))$ is utilized to prioritize links. By construction, the index $C_i(h_i(t))$ represents the service charge that the network *would be willing to pay* in order to transmit a packet using link $i$ during slot $t$. Intuitively, by selecting the link with highest $C_i(h_i(t))$, the Whittle's Index policy is transmitting the most valuable packet.

Notice that the Whittle's Index policy is similar to the Max-Weight policy despite the fact that they were developed using different methods. Whittle's Index and Max-Weight[9] policies prioritize link $i$ according to

$$w_i p_i h_i^2(t) + w_i p_i \left(\frac{2}{p_i} - 1\right) \quad \text{and} \quad w_i p_i h_i^2(t)\,, \text{ respectively.}$$

Moreover, both $WI$ and $MW$ are equivalent to the $MAF$ policy when the network is symmetric, implying that $WI$ and $MW$ are AoI-optimal when $w_i = w$ and $p_i = p$. Next, we derive the performance guarantee $\rho^{WI}$ for the Whittle's Index policy in general networks, with possibly different values of $w_i$ and $p_i$.

[9] Max-Weight with $\tilde{\alpha}_i = \sqrt{w_i/p_i}$ prioritize link $i$ according to $\tilde{\alpha}_i p_i h_i(t)$, which is equivalent to $w_i p_i h_i^2(t)$.

**Theorem 2.21** (Performance of Whittle's Index policy)**.** *Consider a wireless network with parameters* $(N, p_i, w_i)$ *and an infinite time-horizon. The Whittle's Index policy is* $\rho^{WI}$*-optimal, where*

$$\rho^{WI} = 4 \frac{\left( \bar{\mathbb{M}} \left[ \sqrt{\frac{w_i}{p_i}} \left( \frac{\sqrt{2}}{p_i} + \frac{1}{\sqrt{2}} \right) \right] \right)^2}{\left( \bar{\mathbb{M}} \left[ \sqrt{\frac{w_i}{p_i}} \right] \right)^2 + \frac{1}{N} \bar{\mathbb{M}} [w_i]}. \tag{2.67}$$

The complete proof of Theorem 2.21 is provided in [51, Appendix H]. The key idea of the proof is to transform the Whittle's Index policy into a Max-Weight policy and then use arguments similar to the ones in the proof of Theorem 2.16 (Performance of Max-Weight policy). Next, we evaluate the performance of the low-complexity scheduling policies discussed in this section using MATLAB simulations.

## 2.3 Simulation Results

In this section, we evaluate the performance of the scheduling policies in terms of the Expected Weighted Sum Age of Information in (2.4). We compare five scheduling policies: i) Maximum Age First policy; ii) Randomized policy with $\beta_i = \sqrt{w_i/p_i}$ ; iii) Max-Weight policy with $\tilde{\alpha}_i = \sqrt{w_i/p_i}$ ; iv) Whittle's Index policy and v) Dynamic Program. The numerical results associated with the first four policies are *simulations*, while the results associated with the Dynamic Program are *computations* of the EWSAoI obtained by applying the recursion in (2.11). By definition, the Dynamic Program yields the optimal performance.

Figures. 2-6, 2-7 and 2-8 evaluate the scheduling policies in a variety of network settings. In Fig. 2-6, we consider a two-user *symmetric* network with a total of $T = 500$ slots and both links having the same weight $w_1 = w_2 = 1$ and channel reliability $p_1 = p_2 = p \in \{1/15, \cdots, 14/15\}$. In Fig. 2-7, we consider a two-user *general* network with a total of $T = 500$ slots, both links having the same channel reliability $p_1 = p_2 = p \in \{1/15, \cdots, 14/15\}$

but different weights $w_1 = 10$ and $w_2 = 1$. In Fig. 2-8, we consider larger networks. Due to the high computation complexity associated with the Dynamic Program, we show the Lower Bound $L_B$ from (2.15) instead. We consider a network with an increasing number of links $N \in \{5, 10, \cdots, 45, 50\}$, a total of $T = 100,000$ slots, channel reliability $p_i = i/N$, $\forall i$ and all links having the same weight $w_i = 1$. The initial AoI vector is $\vec{h}_1 = [1, 1, \cdots, 1]^T$ in all simulations.

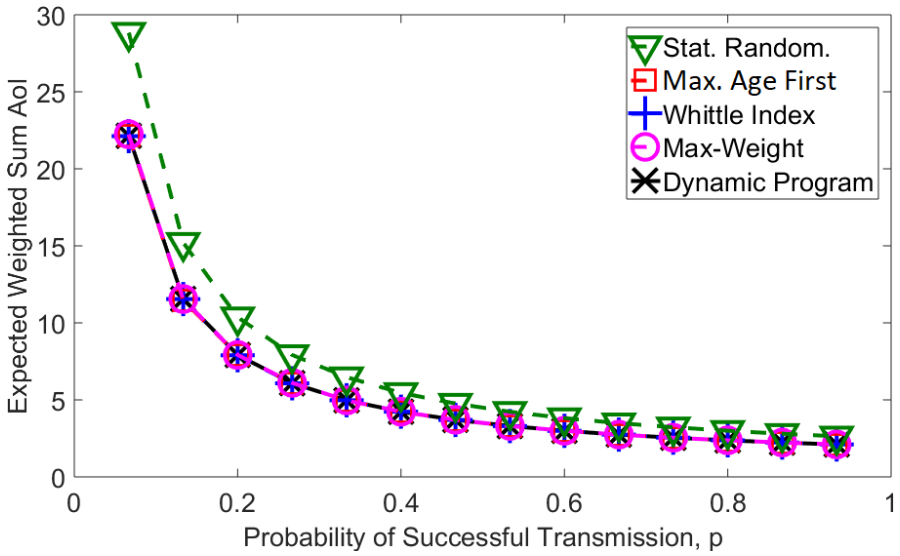



Figure 2-6: Two-user symmetric network with $T = 500, w_i = 1, p_i = p$, $\forall i$. The simulation result for each policy and for each value of $p$ is an average over $1,000$ runs.

Our results in Figs. 2-6, 2-7 and 2-8 show that the performances of the Max-Weight and Whittle's Index policies are comparable to the optimal performance in every network setting considered. The results in Fig. 2-6 support the optimality of the Maximum Age First, Max-Weight, and Whittle's Index policies for any symmetric network. Figs. 2-7 and 2-8 suggest that, in general, the Max-Weight and Whittle's Index policies outperform Maximum Age First and Randomized. An important feature of Maximum Age First, Randomized, Max-Weight, and Whittle's Index policies is that they require low computational resources even for networks with a large number of nodes.

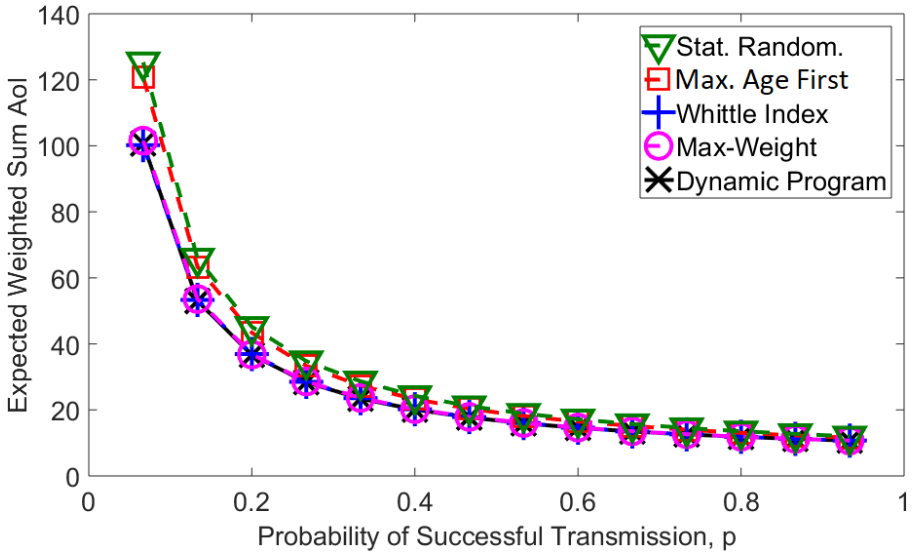



Figure 2-7: Two-user general network with $T = 500, w_1 = 10, w_2 = 1, p_i = p, \forall i$. The simulation result for each policy and for each value of $p$ is an average over $1,000$ runs.

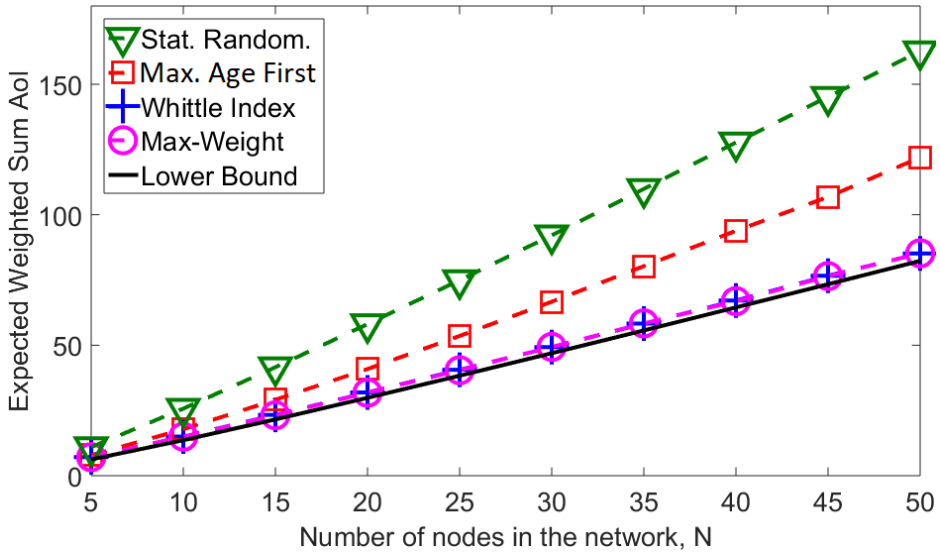



Figure 2-8: Network with $T = 100,000, w_i = 1, p_i = i/N, \forall i$. The simulation result for each policy and for each value of $N$ is an average over 10 runs.

## 2.4 Summary

In this chapter, we considered a broadcast single-hop wireless network with sources that generate fresh packets *on demand* and transmit them via unreliable communication links. We formulated the problem of optimizing transmission scheduling decisions with respect

to the Expected Weighted Sum AoI in the network.

First, we obtained the AoI-optimal policy using Dynamic Programming. We showed that the computational complexity of such solution grows exponentially with the number of sources $N$, making it suitable for small networks. For large networks, we developed four low-complexity scheduling policies and derived performance guarantees for each of them as a function of network parameters, in particular the network size $N$, the channel reliabilities $\{p_i\}_{i=1}^N$ and the weights $\{w_i\}_{i=1}^N$. A summary of the main results follows:

- Maximum Age First policy is AoI-optimal for the case of symmetric networks, when all links have the same channel reliability $p_i = p$ and weight $w_i = w$. The performance guarantee for general networks is given in Theorem 2.9;
- Stationary Randomized policy with $\beta_i = \sqrt{w_i/p_i}$ is 2-optimal for any network configuration $(N, p_i, w_i)$;
- Max-Weight policy with $\tilde{\alpha}_i = \sqrt{w_i/p_i}$ is AoI-optimal for symmetric networks and 2-optimal for general networks; and
- Whittle's Index policy is AoI-optimal for symmetric networks. The performance guarantee for general networks is given in Theorem 2.21.

Simulation results show that both Max-Weight and Whittle's Index policies outperform the other scheduling policies in every configuration simulated, and achieve near optimal information freshness.

In [51, 52], we generalize the results in this chapter to the case of nodes generating packets *periodically*. In chapter 4 of this thesis, we generalize the results in this chapter to the case of nodes generating packets *stochastically*.

# Appendices

## 2.A Proof of Proposition 2.17 (Threshold Policy)

**Proposition 2.17** (Threshold Policy). The stationary scheduling policy that solves Bellman equations (2.63a)-(2.63b) is a threshold policy that, in each slot $t$, transmits when $h > H-1$ and idles when $1 \leq h \leq H-1$. For positive fixed values of $C > 0$, the expression for the threshold is

$$H = \left\lfloor \frac{3}{2} - \frac{1}{p} + \sqrt{\left(\frac{1}{p} - \frac{1}{2}\right)^2 + \frac{2C}{wp}} \right\rfloor .$$

*Proof.* Consider the Decoupled Model in (2.59a)-(2.59b). During slot $t$, the scheduling policy $\pi$ must decide between transmitting and idling. If $\pi$ transmits, the network incurs a service charge $C$ and the value of $h$ reduces to unity with probability $p$ and increases to $h+1$ with probability $1-p$. On the other hand, if $\pi$ idles, the value of $h$ increases to $h+1$ and there is no service charge. Intuitively, we expect that the optimal scheduling decision is to transmit when $h$ is high and idle when $h$ is low. In particular, if the optimal scheduling decision is to transmit when $h = H$, we expect that it is optimal to transmit for all $h \geq H$. This behavior is characteristic of threshold policies.

We start the proof by assuming that the optimal policy $\pi^*$ is a threshold policy that idles when $1 \leq h \leq H-1$ and transmits when $h > H-1$, for a given value of $H \geq 1$. Then, using this assumption, we solve Bellman equations (2.63a)-(2.63b) and show that the solution is consistent with the assumption. For convenience, we rewrite Bellman equations below as

$$S(1) = 0 \quad \text{and} \tag{2.68a}$$

$$S(h) = S(h+1) - \lambda + wh + \min_{u \in \{0,1\}} \{\, 0 \,;\, C - pS(h+1) \,\} . \tag{2.68b}$$

First, we *analyze the case* $\mathbf{h \geq H}$. According to (2.68b), the condition for the threshold

policy $\pi^*$ to transmit in a slot with state $h \geq H$ is

$$S(h+1) > \frac{C}{p} \,. \tag{2.69}$$

Assuming that condition (2.69) holds, it follows from (2.68b) that

$$S(h) = -\lambda + wh + C + (1-p)S(h+1) \,.$$

Since this expression is valid for all $h \geq H$, we can substitute $S(h+1)$ above and get

$$S(h) = -\lambda + wh + C + (1-p)\left[-\lambda + w(h+1) + C\right] + (1-p)^2 S(h+2) \,.$$

Repeating this procedure $n$ times, yields

$$\begin{aligned} S(h) =& [wh + C - \lambda][1 + (1-p) + \cdots + (1-p)^n] + \\ &+ w[(1-p) + 2(1-p)^2 + \cdots + n(1-p)^n] + \\ &+ (1-p)^{n+1} S(h+n+1) \,, \end{aligned}$$

and in the limit as $n \to \infty$ we have

$$S(h) = \frac{wh + C - \lambda}{p} + \frac{w(1-p)}{p^2} \,, \text{ for } h \geq H \,. \tag{2.70}$$

Notice from (2.70) that $(1-p)^{n+1} S(h+n+1) \to 0$ as $n \to \infty$.

Next, we *analyze the case* $\mathbf{1} \leq \mathbf{h} \leq \mathbf{H-1}$. According to (2.68b), the condition for the threshold policy $\pi^*$ to idle in a slot with state $h \in \{1, \cdots, H-1\}$ is

$$S(h+1) \leq \frac{C}{p} \,. \tag{2.71}$$

Assuming that condition (2.71) holds, it follows from (2.68b) that

$$S(h) = S(h+1) - \lambda + wh \,.$$

In particular, for $h = H-1$, we have:

$$S(H-1) = S(H) - \lambda + w(H-1)\,,$$

where $S(H)$ is given in (2.70). Next, for $h = H-2$, we have:

$$\begin{aligned} S(H-2) &= S(H-1) - \lambda + w(H-2) \\ &= S(H) - 2\lambda + 2wH - w(1+2)\,. \end{aligned}$$

Repeating this process $n$ times, yields

$$\begin{aligned} S(H-n) &= S(H) - n\lambda + nwH - w(1+2+\cdots+n) \\ &= S(H) - n\lambda + nwH - \frac{w(n+1)n}{2}\,, \text{ for } n \in \{1,\cdots,H-1\}\,. \end{aligned} \tag{2.72}$$

and substituting $n = H-h$ we have

$$S(h) = S(H) - (H-h)\left[\lambda - wH + \frac{w(H-h+1)}{2}\right]\,, \text{ for } h \in \{1,\cdots,H-1\}\,. \tag{2.73}$$

In (2.70) and (2.73), we have expressions for the differential cost-to-go $S(h)$ as a function of the threshold $H$ and the optimal average cost $\lambda$. To find $H$ and $\lambda$, we analyze $S(h)$ at two points: 1) when $h$ is in the vicinity of the threshold; and 2) when $h = 1$.

**Analysis of S(h) in the vicinity of the threshold.** Policy $\pi^*$ idles when $h = H-1$ and transmits when $h = H$. Merging the conditions in (2.69) and (2.71) with the appropriate values of $h$ yields

$$S(H) \leq \frac{C}{p} < S(H+1)\,. \tag{2.74}$$

From the monotonicity of $S(h)$ in (2.70), it follows that there exists $h' = H + \gamma$ with $H \in \{1,2,3,\cdots\}$ and $\gamma \in [0,1)$ for which

$$S(h' = H+\gamma) = \frac{C}{p}\,. \tag{2.75}$$

Substituting (2.70) into (2.75) yields

$$\begin{aligned}\frac{C}{p} &= \frac{w(H+\gamma)+C-\lambda}{p}+\frac{w(1-p)}{p^2}\\ \lambda &= w(H+\gamma)-w+\frac{w}{p}\,.\end{aligned} \tag{2.76}$$

**Analysis of $\mathbf{S(h)}$ when $\mathbf{h=1}$.** From Bellman equations in (2.63a) we have $S(1)=0$. Substituting the expression for $S(h)$ in (2.73) gives

$$S(H)-(H-1)\left[\lambda-wH+\frac{wH}{2}\right]=0\,.$$

and substituting $S(H)$ from (2.70) yields

$$\frac{C+wH-\lambda}{p}+\frac{w(1-p)}{p^2}=(H-1)\left[\lambda-wH+\frac{wH}{2}\right]\,, \tag{2.77}$$

and substituting $\lambda$ from (2.76) gives

$$\frac{C-w\gamma}{p}=(H-1)\left[w\gamma-w+\frac{w}{p}+\frac{wH}{2}\right]\,. \tag{2.78}$$

Manipulating this quadratic equation on $H$ gives the unique positive solution

$$H=\frac{3}{2}-\frac{1}{p}-\gamma+\sqrt{\frac{2C}{wp}+\left(\frac{1}{p}-\frac{1}{2}\right)^2-\gamma(1-\gamma)}\,. \tag{2.79}$$

It is easy to see from (2.79) that the derivative $dH/d\gamma<0$ when $\gamma\in[0,1)$, implying that $H$ is monotonically decreasing. Hence, in the range $\gamma\in[0,1)$, the value of $H$ decreases from

$$\begin{aligned}H(\gamma=0)&=\frac{3}{2}-\frac{1}{p}+\sqrt{\frac{2C}{wp}+\left(\frac{1}{p}-\frac{1}{2}\right)^2}\;;\;\text{to}\\ H(\gamma=1)&=\frac{1}{2}-\frac{1}{p}+\sqrt{\frac{2C}{wp}+\left(\frac{1}{p}-\frac{1}{2}\right)^2}\,.\end{aligned}$$

Since $H(0)-H(1)=1$, there exists a unique $\gamma^*\in[0,1)$ such that $H(\gamma^*)$ is integer-valued

and the expression for the threshold $H$ is given by

$$H = H(\gamma^*) = \left\lfloor \frac{3}{2} - \frac{1}{p} + \sqrt{\frac{2C}{wp} + \left(\frac{1}{p} - \frac{1}{2}\right)^2} \right\rfloor . \tag{2.80}$$

With the expression of $H$, we can obtain the optimal average cost by isolating $\lambda$ in (2.77) as follows

$$\lambda = \frac{w}{p} + \frac{\frac{C}{p} + \frac{wH(H-1)}{2}}{H + \frac{(1-p)}{p}} . \tag{2.81}$$

Finally, with the closed-form expressions for the differential cost-to-go $S(h)$, threshold $H$ and optimal average cost $\lambda$, it is possible to validate the consistency between the solution and the (initial) assumption of a threshold policy. For the solution of Bellman equations (2.68a)-(2.68b) to be a threshold policy that idles when $h \in \{1, \cdots, H-1\}$ and transmits when $h \geq H$, the following condition should hold:

$$S(h^{(-)}+1) \leq \frac{C}{p} < S(h^{(+)}+1) ,$$

for all $h^{(-)} \in \{1, \cdots, H-1\}$ and $h^{(+)} \in \{H, H+1, \cdots\}$. Since $S(h)$ in (2.70) and (2.73) is monotonically increasing it is sufficient to show that

$$S(H) \leq \frac{C}{p} < S(H+1) . \tag{2.82}$$

Recall from (2.75) that for $H$ in (2.80) there exists $\gamma \in [0,1)$ such that

$$S(H+\gamma) = \frac{C}{p} .$$

From the monotonicity of $S(h)$ in (2.70), it follows that condition (2.82) is satisfied. Thus, the solution to Bellman equations is consistent. ■

Chapter 3

# Throughput Constrained AoI Optimization

The scheduling policies discussed in chapter 2 attempt to minimize AoI by dynamically allocating resources to the different links in the network. Depending on the network configuration and scheduling policy, some links can be activated frequently, while others less often. The frequency at which information is delivered to the destination is of particular importance in sensor networks. Clearly, a sensor that measures the quantity of fuel requires a lower update frequency (i.e. throughput) than a sensor that is measuring the proximity to obstacles in order to avoid collisions. For capturing this attribute, we associate a minimum throughput requirement with each node in the network. Hence, in addition to providing information freshness, the scheduling policies should also fulfill throughput constraints from the individual nodes.

It is important to emphasize that high throughput does not guarantee low AoI. Low packet delay and service regularity are also necessary to achieve low AoI. *In this chapter, we consider the problem of minimizing the AoI in the network while simultaneously satisfying throughput requirements from the individual nodes*. First, we derive a lower bound on the AoI performance achievable by any given network. Then, we develop two low-complexity transmission scheduling policies, namely Randomized and Drift-Plus-Penalty, and show that both are guaranteed to be within a factor of two away from the lower bound,

while simultaneously satisfying any feasible throughput requirements. To the best of our knowledge, this is the first work[1] to consider AoI-based policies that provably satisfy throughput constraints of multiple destinations simultaneously.

The remainder of this chapter is organized as follows. In Sec. 3.1, we formulate the throughput constrained AoI optimization problem. In Sec. 3.2.1 we derive an analytical lower bound on the AoI optimization. In Sec. 3.2.2, we find the scheduling policy that solves the AoI optimization over the class of Stationary Randomized policies. In Sec. 3.2.3, we develop the Drift-Plus Penalty policy and obtain performance guarantees in terms of AoI and throughput. In Sec. 3.3, both policies are simulated and compared to the state-of-the-art in the literature. A summary of results is provided in Sec. 3.4.

## 3.1 System Model

Consider a single-hop wireless network with a base station (BS) and $N$ nodes sharing time-sensitive information through unreliable communication links, as described in Sec. 2.1 and illustrated in Fig. 2-1. The transmission scheduling policy $\pi \in \Pi$ controls the decision $\{u_i(t)\}_{i=1}^N$ of the BS in each slot $t$. Recall that $u_i(t) \in \{0,1\}$ is 1 when the BS selects link $i$ during slot $t$, $c_i(t) \in \{0,1\}$ is 1 when the channel associated with link $i$ is ON, and $d_i(t) \in \{0,1\}$ is 1 when a packet is successfully transmitted via link $i$. The *interference constraint* associated with the wireless channel imposes that $\sum_{i=1}^N u_i(t) \leq 1$, $\forall t \in \{1,\cdots,T\}$, meaning that, at any given slot $t$, the scheduling policy can select at most one link for transmission. The channel state process is i.i.d. over time and independent across different links, with $\mathbb{P}(c_i(t)=1) = p_i, \forall i,t$. It follows from $d_i(t) = c_i(t)u_i(t)$ that $\mathbb{E}[d_i(t)] = p_i\mathbb{E}[u_i(t)]$.

Let $q_i$ be a strictly positive real value that represents the minimum throughput requirement of link $i$. Using the random variable $d_i^\pi(t)$, we define the *long-term throughput* of link $i$ when policy $\pi$ is employed as

$$\hat{q}_i^\pi := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[d_i^\pi(t)] \,. \tag{3.1}$$

[1] This work was first published in [49] and [50].

Then, we express the *minimum throughput constraint* of each link as $\hat{q}_i^{\pi} \geq q_i \, , \forall i \in \{1, \cdots, N\}$.

We assume that $\{q_i\}_{i=1}^N$ is a *feasible* set of minimum throughput requirements, i.e. there exists a policy $\pi \in \Pi$ that satisfies all $T$ interference constraints and all $N$ throughput constraints simultaneously.

**Remark 3.1.** *The inequality*

$$\sum_{i=1}^{N} \frac{q_i}{p_i} \leq 1 \, , \tag{3.2}$$

*is a necessary and sufficient condition for the feasibility of* $\{q_i\}_{i=1}^N$.

*Proof.* The necessary condition is shown by substituting $\mathbb{E}\left[d_i^{\pi}(t)\right] = p_i \mathbb{E}\left[u_i^{\pi}(t)\right]$ into (3.1), manipulating the expression and summing over all $i$. Then, by applying the interference constraints, yields

$$\sum_{i=1}^{N} \frac{q_i}{p_i} \leq \sum_{i=1}^{N} \frac{\hat{q}_i^{\pi}}{p_i} = \lim_{T \to \infty} \frac{1}{T} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbb{E}[u_i^{\pi}(t)] \leq 1 \, . \tag{3.3}$$

Sufficiency is shown by constructing a policy that fulfills the throughput requirements $\{q_i\}_{i=1}^N$ when (3.2) is satisfied. One example of such policy is given in Sec. 3.2.2 ■

Throughout this chapter, we assume that (3.2) is satisfied with strict inequality. With the definitions of AoI in Sec. 2.1 and throughput in Sec. 3.1, we present the AoI optimization problem subject to the minimum throughput requirements.

**Throughput constrained AoI optimization**

$$\mathbb{E}[J^*] = \min_{\pi \in \Pi} \left\{ \lim_{T \to \infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} w_i \mathbb{E}\left[h_i(t)\right] \right\} \tag{3.4a}$$

$$\text{s.t. } \hat{q}_i^{\pi} \geq q_i \, , \forall i \, ; \tag{3.4b}$$

$$\textstyle\sum_{i=1}^{N} u_i(t) \leq 1 \, , \forall t \, . \tag{3.4c}$$

The scheduling policy that results from (3.4a)-(3.4c) is referred to as *AoI-optimal*. To compare the performance of different scheduling policies, we employ the optimality ratio $\rho^{\pi} = U_B^{\pi} / L_B$, where $L_B$ is an universal lower bound to the optimization problem in (3.4a)-(3.4c) and $U_B^{\pi}$ is an upper bound to the performance of the admissible policy $\pi \in \Pi$. Admissible policies are non-anticipative and satisfy both constraints (3.4b) and (3.4c).

## 3.2 Scheduling Policies

In this section, we obtain a lower bound $L_B$ to the AoI optimization (3.4a)-(3.4c). Then, we propose two low-complexity scheduling policies with strong AoI performances that provably satisfy the throughput constraints *for every feasible set* $\{q_i\}_{i=1}^N$. To evaluate the AoI performance of each policy, we find their corresponding optimality ratio $\rho^\pi$. Moreover, in Sec. 3.3, we simulate and compare these policies to the state-of-the-art in the literature.

### 3.2.1 Universal Lower Bound

In this section, we derive a lower bound to the optimization problem in (3.4a)-(3.4c).

**Theorem 3.2** (Lower Bound)**.** *The optimization problem in* (3.5a)-(3.5c) *provides a lower bound $L_B$ to the AoI optimization* (3.4a)-(3.4c)*, namely $L_B \leq \mathbb{E}[J^*]$ for every network setup $(N, p_i, q_i, w_i)$.*

Lower Bound

$$L_B = \min_{\pi\in\Pi} \left\{ \frac{1}{2N} \sum_{i=1}^{N} w_i \left( \frac{1}{\hat{q}_i^\pi} + 1 \right) \right\} \tag{3.5a}$$

$$\text{s.t. } \hat{q}_i^\pi \geq q_i \,, \forall i \,; \tag{3.5b}$$

$$\textstyle\sum_{i=1}^{N} u_i(t) \leq 1 \,, \forall t \,. \tag{3.5c}$$

*Proof Outline.* From (2.26), which was derived for the proof of Theorem 2.1, we know that the RHS in (3.4a) can be written as

$$\begin{aligned} \lim_{T\to\infty} J_T^\pi &= \lim_{T\to\infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} w_i h_i(t) = \\ &= \frac{1}{2N} \sum_{i=1}^{N} w_i \left[ \frac{\bar{\mathbb{M}}[I_i^2]}{\bar{\mathbb{M}}[I_i]} + 1 \right] \geq \frac{1}{2N} \sum_{i=1}^{N} w_i \left[ \bar{\mathbb{M}}[I_i] + 1 \right] \quad \text{w.p.1} \,. \end{aligned} \tag{3.6}$$

Intuitively, we know that the inter-delivery time $\bar{\mathbb{M}}[I_i]$ is inversely proportional to the throughput $\hat{q}_i^{\pi}$ what gives (3.5a). The complete proof is in Appendix 3.A. ■

Notice that the lower bound in (3.5a)-(3.5c) depends only on the long-term throughput associated with policy $\pi$. In turn, the throughput $\hat{q}_i^{\pi} \in (0,1]$ depends only on the total number of packets delivered to node $i$ during the time-horizon $T$. The unique solution to this minimization problem is given by Corollary 3.6 and Algorithm 1. In the next section, we use this lower bound to obtain a tight optimality ratio, $\rho^R < 2$, for a Stationary Randomized policy.

### 3.2.2 Stationary Randomized Policy

Denote by $\Pi_R$ the class of Stationary Randomized Policies and let $R \in \Pi_R$ be a Randomized policy that makes scheduling decisions randomly, according to fixed probabilities $\{\mu_i\}_{i=1}^N$, where $\mu_i = \mathbb{E}[u_i(t)] \in (0,1], \forall i, \forall t$, and $\mu_{idle} = 1 - \sum_{i=1}^N \mu_i$.

**Definition 3.3** (Randomized policy)**.** *The Randomized policy selects, in each slot t, link i with probability $\mu_i$, or idles with probability $\mu_{idle}$.*

Observe that each policy in $\Pi_R$ is fully characterized by the set of scheduling probabilities $\{\mu_i\}_{i=1}^N$.Next, we find the Optimal Stationary Randomized policy $R^*$ that solves the AoI optimization (3.4a)-(3.4c) over the class $\Pi_R \subset \Pi$ and derive the associated optimality ratio $\rho^R$.

**Remark 3.4.** *Consider a policy $R \in \Pi_R$ with scheduling probabilities $\{\mu_i\}_{i=1}^N$. The long-term throughput and the expected time-average AoI of link i can be expressed as*

$$\hat{q}_i^R = p_i \mu_i \,; \tag{3.7}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[h_i^R(t)] = \frac{1}{p_i \mu_i} \,. \tag{3.8}$$

*Proof.* In any given slot $t$, the BS receives a packet from link $i$ if this link is scheduled and the corresponding packet transmission is successful. It follows that $\mathbb{E}[d_i^R(t)] = p_i\mu_i, \forall i,t$. Hence, by the definition of throughput in (3.1) and the renewal-reward result in (2.45), we obtain (3.7) and (3.8), respectively. ■

Substituting both expressions from Remark 3.4 into the AoI optimization (3.4a)-(3.4c) gives the equivalent optimization problem over the class $\Pi_R$ presented below.

Optimization over Randomized policies

$$\mathbb{E}[J^{R*}] = \min_{R\in\Pi_R} \left\{ \frac{1}{N} \sum_{i=1}^{N} \frac{w_i}{p_i\mu_i} \right\} \tag{3.9a}$$

$$\text{s.t. } p_i\mu_i \geq q_i \,, \forall i \,; \tag{3.9b}$$

$$\textstyle\sum_{i=1}^{N} \mu_i \leq 1 \,. \tag{3.9c}$$

Notice that under the class $\Pi_R$, conditions (3.9c) and (3.4c) are equivalent. The Optimal Stationary Randomized policy $R^*$ is characterized by the set $\{\mu_i^*\}_{i=1}^N$ that solves (3.9a)-(3.9c).

**Theorem 3.5** (Optimality Ratio for $R^*$)**.** *The optimality ratio of $R^*$ is such that $\rho^R < 2$, i.e. the Optimal Stationary Randomized policy is 2-optimal for any wireless network.*

*Proof.* Let $\hat{q}_i^L$ be the throughput associated with the policy that solves the Lower Bound (3.5a)-(3.5c). Consider the policy $R \in \Pi_R$ with long-term throughput $\hat{q}_i^R = p_i\mu_i = \hat{q}_i^L$ for each link $i$. Since $\hat{q}_i^R = \hat{q}_i^L$, it follows that $R$ satisfies all throughput constraints. Comparing $L_B$ in (3.5a) with the objective function associated with $R$, namely $\mathbb{E}[J^R]$, yields

$$\frac{\mathbb{E}[J^R]}{2} < L_B \;\rightarrow\; \rho^R = \frac{\mathbb{E}[J^{R*}]}{\mathbb{E}[J^*]} \leq \frac{\mathbb{E}[J^R]}{L_B} < 2 \,, \tag{3.10}$$

where $\mathbb{E}[J^*]$ comes from (3.4a) and $\mathbb{E}[J^{R*}]$ from (3.9a). Recall that $L_B \leq \mathbb{E}[J^*] \leq \mathbb{E}[J^{R*}] \leq \mathbb{E}[J^R]$. ■

**Corollary 3.6.** *The Optimal Stationary Randomized policy $R^*$ is also the solution for the Lower Bound problem* (3.5a)-(3.5c).

*Proof.* Using the same argument as in the proof of Theorem 3.5, in particular $\hat{q}_i^R = p_i\mu_i = \hat{q}_i^L$, it follows that the scheduling policy that solves the Optimization over Randomized policies (3.9a)-(3.9c) also solves the Lower Bound (3.5a)-(3.5c). ■

**Theorem 3.7** (Optimal Stationary Randomized policy)**.** *The scheduling probabilities $\{\mu_i^*\}_{i=1}^N$ that result from Algorithm 1 are the unique solution to* (3.9a)-(3.9c) *and, thus, characterize the Optimal Stationary Randomized policy $R^*$.*

**Algorithm 1** Unique solution to Karush–Kuhn–Tucker (KKT) conditions

1: $\gamma_i \leftarrow w_i p_i / N q_i^2$ , $\forall i \in \{1,2,\cdots,N\}$
2: $\gamma \leftarrow \max_i\{\gamma_i\}$
3: $\mu_i \leftarrow (q_i/p_i)\max\{1\,;\sqrt{\gamma_i/\gamma}\}$ , $\forall i$
4: $S \leftarrow \mu_1 + \mu_2 + \cdots + \mu_N$
5: **while** $S < 1$ **do**
6: decrease $\gamma$ slightly
7: repeat steps 3 and 4 to update $\mu_i$ and $S$
8: **end while**
9: $\mu_i^* = \mu_i, \forall i$, and $\gamma^* = \gamma$
10: **return** $(\mu_1^*, \mu_2^*, \cdots, \mu_N^*, \gamma^*)$

*Proof.* To find the set of scheduling probabilities $\{\mu_i^*\}_{i=1}^N$ that solve the optimization problem (3.9a)-(3.9c), we analyze the KKT Conditions. Let $\{\lambda_i\}_{i=1}^N$ be the KKT multipliers associated with the relaxation of (3.9b) and $\gamma$ be the multiplier associated with the relaxation of (3.9c). Then, for $\lambda_i \geq 0, \forall i$, $\gamma \geq 0$ and $\mu_i \in [q_i/p_i, 1], \forall i$, we define

$$\mathscr{L}(\mu_i,\lambda_i,\gamma) = \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{p_i\mu_i} + \sum_{i=1}^{N}\lambda_i(q_i - p_i\mu_i) + \gamma\left(\sum_{i=1}^{N}\mu_i - 1\right), \tag{3.11}$$

and, otherwise, we define $\mathscr{L}(\mu_i,\lambda_i,\gamma) = +\infty$. Then, the KKT Conditions are

(i) Stationarity: $\nabla_{\mu_i}\mathscr{L}(\mu_i,\lambda_i,\gamma)=0$;

(ii) Complementary Slackness: $\gamma\left(\sum_{i=1}^{N}\mu_i-1\right)=0$;

(iii) Complementary Slackness: $\lambda_i(q_i-p_i\mu_i)=0,\forall i$;

(iv) Primal Feasibility: $p_i\mu_i\geq q_i$ ,$\forall i$, and $\sum_{i=1}^{N}\mu_i\leq 1$; and

(v) Dual Feasibility: $\lambda_i\geq 0,\forall i$, and $\gamma\geq 0$.

Since $q_i$ is strictly positive, the function $\mathscr{L}(\mu_i,\lambda_i,\gamma)$ is *convex* on the interval of interest $\mu_i\in[q_i/p_i,1]$. Therefore, if there exists a vector $(\{\mu_i^*\}_{i=1}^N,\{\lambda_i^*\}_{i=1}^N,\gamma^*)$ that satisfies all KKT Conditions, then this vector is unique. As a result, the scheduling policy $R^*\in\Pi_R$ that optimizes (3.9a)-(3.9c) is also unique and is characterized by $\{\mu_i^*\}_{i=1}^N$. Next, we find the vector $(\{\mu_i^*\}_{i=1}^N,\{\lambda_i^*\}_{i=1}^N,\gamma^*)$.

To assess stationarity, $\nabla_{\mu_i}\mathscr{L}(\mu_i,\lambda_i,\gamma)=0$, we calculate the partial derivative of $\mathscr{L}(\mu_i,\lambda_i,\gamma)$ with respect to $\mu_i$. It follows from the derivative that

$$\frac{w_i}{Np_i\mu_i^2}+\lambda_i p_i=\gamma\,,\forall i\,. \tag{3.12}$$

From complementary slackness, $\gamma(\sum_{i=1}^N\mu_i-1)=0$, we know that either $\gamma=0$ or $\sum_{i=1}^N\mu_i=1$. Equation (3.12) shows that the value of $\gamma$ can only be zero if $\lambda_i=0$ and $\mu_i\to\infty$, which violates $\mu_i\in[q_i/p_i,1]$. Hence, we obtain

$$\gamma>0\quad\text{and}\quad\sum_{i=1}^{N}\mu_i=1\,. \tag{3.13}$$

Notice that $\sum_{i=1}^N\mu_i=1$ implies in $\mu_{idle}=0$.

Based on dual feasibility, $\lambda_i\geq 0$, we can separate links $i\in\{1,\cdots,N\}$ into two categories: links with $\lambda_i>0$ and links with $\lambda_i=0$.

*Category 1)* links $i$ with $\lambda_i>0$. It follows from complementary slackness, $\lambda_i(q_i-p_i\mu_i)=0$, that

$$\mu_i=\frac{q_i}{p_i}\,. \tag{3.14}$$

Plugging this value of $\mu_i$ into (3.12) gives the inequality $\lambda_i p_i = \gamma - \gamma_i > 0$, where we defined the constant

$$\gamma_i := \frac{w_i p_i}{N q_i^2} \,. \tag{3.15}$$

*Category 2)* links $i$ with $\lambda_i = 0$. It follows from (3.12) that

$$\gamma = \gamma_i \left( \frac{q_i}{p_i \mu_i} \right)^2 \;\rightarrow\; \mu_i = \frac{q_i}{p_i} \sqrt{\frac{\gamma_i}{\gamma}} \,. \tag{3.16}$$

In summary, for any fixed value of $\gamma > 0$, the scheduling probability of link $i$ is

$$\mu_i = \frac{q_i}{p_i} \max \left\{ 1, \sqrt{\frac{\gamma_i}{\gamma}} \right\} \,. \tag{3.17}$$

Notice that for a decreasing value of $\gamma$, the probability $\mu_i$ remains fixed or increases. Our goal is to find the value of $\gamma^*$ that gives $\{\mu_i^*\}_{i=1}^N$ satisfying the condition $\sum_{i=1}^N \mu_i^* = 1$.

*Proposed algorithm to find* $\gamma^*$: start with $\gamma = \max\{\gamma_i\}$. Then, according to (3.17), all links have $\mu_i = q_i/p_i$ and, by the feasibility condition in (3.2), it follows that

$$\sum_{i=1}^{N} \mu_i = \sum_{i=1}^{N} \frac{q_i}{p_i} \leq 1 \,. \tag{3.18}$$

Now, by gradually decreasing $\gamma$ and adjusting $\{\mu_i\}_{i=1}^N$ according to (3.17), we can find the unique $\gamma^*$ that fulfills $\sum_{i=1}^N \mu_i^* = 1$. The solution $\gamma^*$ exists since $\gamma \to 0$ implies in $\sum_{i=1}^N \mu_i \to \infty$. The uniqueness of $\gamma^*$ follows from the monotonicity of $\mu_i$ with respect to $\gamma$. This process is described in Algorithm 1 and illustrated in Fig. 3-1.

Algorithm 1 outputs the set of scheduling probabilities $\{\mu_i^*\}_{i=1}^N$ and the parameter $\gamma^*$. The set $\{\lambda_i^*\}_{i=1}^N$ is obtained using (3.12). Hence, the unique vector $(\{\mu_i^*\}_{i=1}^N, \{\lambda_i^*\}_{i=1}^N, \gamma^*)$ that solves the KKT Conditions is found. ■

In order to fulfill the throughput constraints (3.9b), every scheduling policy in $\Pi_R$ must allocate at least $\mu_i \geq q_i/p_i$ to each link $i$. What differentiates policies in $\Pi_R$ is how they distribute the remaining resources, $1 - \sum_{i=1}^N q_i/p_i$, between links. According to Algorithm 1, the Optimal Stationary Randomized policy $R^*$ supplies additional resources, $\mu_i^* > q_i/p_i$, to links with high value of $\gamma_i$, namely links with a high priority $w_i$ or a low value of $q_i/p_i$.

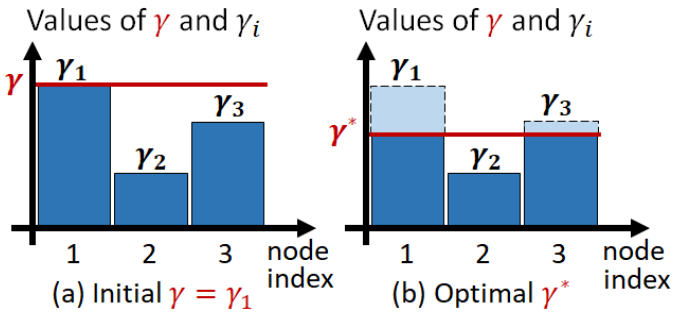



Figure 3-1: Illustration of Algorithm 1 in a network with three links. On the left, the initial configuration with $\gamma = \max\{\gamma_i\}$. On the right, the outcome $\gamma^*$ implies that under policy $R^*$ link 2 will operate with minimum required scheduling probability $\mu_2 = q_2/p_2$, while the other two links will operate with a scheduling probability that is larger than the minimum.

Notice that if a link with low $q_i/p_i$ was given the minimum required amount of resources, it would rarely transmit and its AoI would be high. In contrast, policy $R^*$ allocates the minimum required, $\mu_i^* = q_i/p_i$, to links with low priority $w_i$ or high $q_i/p_i$.

Notice that in the limit as $q_i \to 0, \forall i$, Algorithm 1 gives the same result as Corollary 2.13. For arbitrarily low values of $q_i$, $\gamma_i$ are high and all links have

$$\mu_i = \frac{q_i}{p_i} \max\left\{1; \sqrt{\frac{\gamma_i}{\gamma}}\right\} = \frac{q_i}{p_i}\sqrt{\frac{\gamma_i}{\gamma}} = \sqrt{\frac{w_i}{p_i}}\sqrt{\frac{1}{N\gamma}}\,. \tag{3.19}$$

Then, the sum of scheduling probabilities is $\sum_{i=1}^N \mu_i^* = 1$ only when

$$\mu_i^* = \sqrt{\frac{w_i}{p_i}} \Bigg/ \sum_{j=1}^{N} \sqrt{\frac{w_j}{p_j}}, \forall i\,. \tag{3.20}$$

The scheduling probabilities in (3.20) are identical to the ones in Corollary 2.13.

The policies $R \in \Pi_R$ discussed in this section are as simple as possible. They select links randomly, according to fixed scheduling probabilities $\{\mu_i\}_{i=1}^N$ calculated offline by Algorithm 1. Despite their simplicity, $R^*$ satisfies the throughput constraints for every feasible set $\{q_i\}_{i=1}^N$ and is 2-optimal in terms of information freshness, regardless of the network configuration $(N, p_i, q_i, w_i)$. In the following section, we develop scheduling policies that take advantage of additional information, such as the current $h_i(t)$ of each link, for selecting links in an adaptive manner.

### 3.2.3 Drift-Plus-Penalty Policy

The Drift-Plus-Penalty policy is derived using a similar technique as the Max-Weight policy in Sec. 2.2.4. The main difference between these two policies is that the Drift-Plus-Penalty is designed to reduce the sum of the Lyapunov Drift and a Penalty Function, while the Max-Weight policy reduces only the Lyapunov Drift. As we will see, this difference allows the Drift-Plus-Penalty policy to simultaneously achieve low AoI performance and satisfy the throughput requirements. Prior to presenting the Drift-Plus-Penalty policy, we introduce the notions of throughput debt, augmented network state, Lyapunov Function and Lyapunov Drift. Notice that they differ from the definitions in Sec. 2.2.4.

Let $x_i(t)$ be the throughput debt associated with link $i$ at the beginning of slot $t$. The throughput debt evolves as

$$x_i(t+1) = tq_i - \sum_{\tau=1}^{t} d_i(\tau) \, . \tag{3.21}$$

The value of $tq_i$ can be interpreted as the minimum number of packets that link $i$ should have delivered by slot $t+1$ and $\sum_{\tau=1}^{t} d_i(\tau)$ is the total number of packets actually delivered in the same interval. Define the operator $(.)^+ = \max\{(.),0\}$ that computes the positive part of a scalar. Then, the positive part of the throughput debt is given by $x_i^+(t) = \max\{x_i(t);0\}$. A large debt $x_i^+(t)$ indicates to the scheduling policy $\pi \in \Pi$ that link $i$ is lagging behind in terms of throughput. In fact, strong stability of the process $x_i^+(t)$, namely

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[x_i^+(t)] < \infty \, , \tag{3.22}$$

is sufficient to establish that the minimum throughput constraint, $\hat{q}_i^{\pi} \geq q_i$, is satisfied [85, Theorem 2.8].

Denote by $S(t) = (h_i(t), x_i^+(t))_{i=1}^N$ the augmented network state at the beginning of slot $t$ and define the Penalty Function as follows

$$P'(S(t)) := \frac{1}{2} \sum_{i=1}^{N} \tilde{\alpha}_i \mathbb{E}[h_i(t+1)|S(t)] \, , \tag{3.23}$$

where $\tilde{\alpha}_i > 0$ are auxiliary parameters used to tune the performance of the Drift-Plus-

Penalty policy. Observe that $P'(S(t))$ is large when links have high AoI. Then, define the Lyapunov Function

$$L'(S(t)) := \frac{V'}{2}\sum_{i=1}^{N}\left[x_i^+(t)\right]^2 , \tag{3.24}$$

where $V'$ is a strictly positive real value that represents the importance of the throughput constraints. Notice that, as opposed to the Lyapunov Function in (2.48), the expression in (3.24) does not contain an AoI term. This is because the term with $h_i(t)$ is already present in the Penalty Function. Similarly to (2.49), we define the one-slot Lyapunov Drift as

$$\Delta'(S(t)) := \mathbb{E}\left[L'\left(S(t+1)\right) - L'\left(S(t)\right)|S(t)\right] . \tag{3.25}$$

The Drift-Plus-Penalty policy is designed to minimize an upper bound on the sum $\Delta'(S(t)) + P'(S(t))$ at every slot $t$. To obtain this upper bound, we analyze both the Lyapunov Drift and the Penalty Function. Substituting the Lyapunov Function (3.24) into the Drift gives

$$\Delta'(S(t)) = \frac{V'}{2}\sum_{i=1}^{N}\mathbb{E}\left\{[x_i^+(t+1)]^2 - [x_i^+(t)]^2\,|S(t)\right\} . \tag{3.26}$$

To find an expression for $[x_i^+(t+1)]^2 - [x_i^+(t)]^2$, we use the recursion

$$x_i(t+1) = x_i(t) - d_i(t) + q_i \text{ , with } x_i(1) = 0 . \tag{3.27}$$

Notice that (3.27) follows from (3.21). Squaring $x_i^+(t+1)$, yields

$$\begin{aligned}\left[x_i^+(t+1)\right]^2 &= \left[\max\{x_i(t) - d_i(t) + q_i; 0\}\right]^2 \\ &\leq \left[\max\{x_i^+(t) - d_i(t) + q_i; 0\}\right]^2 \\ &\leq [x_i^+(t) - d_i(t) + q_i]^2 .\end{aligned} \tag{3.28}$$

Manipulating (3.28), gives

$$[x_i^+(t+1)]^2 - [x_i^+(t)]^2 \leq -2x_i^+(t)[d_i(t) - q_i] + 1 . \tag{3.29}$$

Taking the conditional expectation of (3.29) and applying $\mathbb{E}[d_i(t)|S(t)] = p_i\mathbb{E}[u_i(t)|S(t)]$

from (2.2), gives

$$\mathbb{E}\left\{ [x_i^+(t+1)]^2 - [x_i^+(t)]^2 \middle| S(t) \right\} \leq -2x_i^+(t)\left(p_i\mathbb{E}\{u_i(t)|S(t)\} - q_i\right) + 1 \,. \tag{3.30}$$

Substituting (3.30) into the Lyapunov Drift in (3.26), yields

$$\Delta'(S(t)) \leq -V' \sum_{i=1}^{N} x_i^+(t)\left(p_i\mathbb{E}\{u_i(t)|S(t)\} - q_i\right) + V'N/2 \,. \tag{3.31}$$

Next, we analyze the Penalty Function (3.23) by utilizing the evolution of $h_i(t)$ in (2.3)

$$\begin{aligned} P'(S(t)) &:= \frac{1}{2}\sum_{i=1}^{N} \tilde{\alpha}_i \mathbb{E}[h_i(t+1)|S(t)] \\ &= \frac{1}{2}\sum_{i=1}^{N} \tilde{\alpha}_i \{h_i(t) + 1 - h_i(t)\mathbb{E}[d_i(t)|S(t)]\} \\ &= \frac{1}{2}\sum_{i=1}^{N} \tilde{\alpha}_i \{h_i(t) + 1 - p_i h_i(t)\mathbb{E}[u_i(t)|S(t)]\} \,. \end{aligned} \tag{3.32}$$

Substituting (3.31) and (3.32) into the sum $\Delta'(S(t)) + P'(S(t))$ yields the expression for the upper bound

$$\Delta'(S(t)) + P'(S(t)) \leq -\sum_{i=1}^{N} \mathbb{E}\left[u_i(t)\,|S(t)\right] W_i'(t) + B'(t) \,, \tag{3.33}$$

where $W_i'(t)$ and $B'(t)$ are given by

$$W_i'(t) = \frac{\tilde{\alpha}_i p_i}{2} h_i(t) + V' p_i x_i^+(t) \,; \tag{3.34}$$

$$B'(t) = \sum_{i=1}^{N} \left\{ \frac{\tilde{\alpha}_i}{2}\left[h_i(t)+1\right] + V' x_i^+(t) q_i + \frac{V'}{2} \right\} \,. \tag{3.35}$$

The values of $W_i'(t)$ and $B'(t)$ can be easily calculated by any admissible policy and thus can be used for making scheduling decisions. However, notice that the term $B'(t)$ in (3.35) is not affected by the choice of $u_i(t)$. The Drift-Plus-Penalty policy minimizes the upper bound in (3.33) at every slot $t$. Denote the Drift-Plus-Penalty policy as *DPP*.

**Definition 3.8** (Drift-Plus-Penalty policy)**.** *The DPP policy selects, in each slot $t$, the link $i$ with highest value of $W_i'(t)$, with ties being broken arbitrarily.*

**Theorem 3.9.** *The DPP policy satisfies any feasible set of minimum throughput requirements $\{q_i\}_{i=1}^N$.*

**Theorem 3.10** (Optimality Ratio for *DPP*)**.** *For any wireless network with parameters $(N, p_i, q_i, w_i)$, by choosing the constants $\tilde{\alpha}_i = w_i/\mu_i^* p_i, \forall i$, the optimality ratio of DPP is such that*

$$\rho^{DPP} \leq 2 + \frac{1}{L_B}\left[V' - \frac{1}{N}\sum_{i=1}^{N} w_i\right] . \tag{3.36}$$

*Moreover, if $V' \leq \sum_{i=1}^N w_i/N$, then the DPP policy is 2-optimal.*

The proofs of Theorems 3.9 and 3.10 are provided in Appendices 3.B and 3.C, respectively. Notice from the expression of $W_i'(t)$ in (3.34) that increasing $V'$ prioritizes the throughput debt minimization over the AoI minimization. This effect is also captured in the expression for $\rho^{DPP}$ in (3.36), where a larger $V'$ loosens the performance guarantee in terms of information freshness.

The Lyapunov Function in (3.24) with a quadratic term in $x_i^+(t)$ has a central role in showing that the *DPP* policy satisfies any feasible requirements $\{q_i\}_{i=1}^N$. The Penalty Function (3.23) with a linear term in $h_i(t)$ is central to show that the *DPP* policy is 2-optimal. Recall that the *MW* policy in Sec. 2.2.4 was also designed around a linear term on $h_i(t)$ and was shown to be 2-optimal. Comparing Theorems 2.16 and 3.10, we can clearly see the similarities. In particular, notice that in the limit as $q_i \to 0, \forall i$, the throughput debt becomes $x_i^+(t) = 0, \forall i, t$. As a result, the *DPP* policy selects links according to $W_i'(t) = \tilde{\alpha}_i p_i h_i(t)/2$ which is equivalent to $W_i'(t) = \sqrt{w_i p_i} h_i(t)$. Hence, the *DPP* policy becomes

identical to the *MW* policy, as expected.

The Optimal Stationary Randomized policy $R^*$ selects links randomly, according to fixed scheduling probabilities $\{\mu_i^*\}_{i=1}^N$. In contrast, the Drift-Plus-Penalty policy *DPP* uses feedback from the network, namely $h_i(t)$ and $x_i^+(t)$, to guide scheduling decisions. We expect the *DPP* policy to outperform $R^*$. In the next section, we simulate $R^*$ and *DPP*, and compare their performance against the state-of-the-art in the literature.

## 3.3 Simulation Results

In this section, we simulate four transmission scheduling policies: 1) Optimal Randomized, $R^*$; 2) Drift-Plus-Penalty, *DPP*; 3) Whittle's Index, *WI*, without throughput constraints; and 4) Largest Weighted-Debt First, *LD*. The first two policies are developed in Sec. 3.2 and the last two are proposed in Sec. 2.2.5 and [34], respectively. Policy *WI* was proposed in Sec. 2.2.5 for minimizing the AoI in broadcast wireless networks *without throughput constraints*. Policy *LD* selects, in each slot $t$, the node with highest value of $x_i(t)/p_i$, where $x_i(t)$ is the throughput debt (3.21). It was shown in [34] that *LD* satisfies any set of feasible throughput requirements $\{q_i\}_{i=1}^N$. Notice that *LD* does not account for AoI.

We simulate a network with $N$ nodes, each having different parameters. Node $i$ has weight $w_i = (N+1-i)/N$, channel reliability $p_i = i/N$ and minimum throughput requirement $q_i = \varepsilon p_i/N$, where $\varepsilon \in [0,1)$ represents the hardness of satisfying the throughput constraints $\hat{q}_i^\pi \geq q_i$. The larger the value of $\varepsilon$, the more challenging are the constraints. Notice that $\varepsilon < 1$ is necessary for the feasibility of $\{q_i\}_{i=1}^N$. The value of $V'$ represent the importance of the throughput constraints for *DPP*. A lower value of $V'$ reduces the priority of the throughput debt and increases the priority of AoI minimization. Recall that for any positive $V'$, the *DPP* policy is guaranteed to satisfy any feasible throughput requirements *in the long run*. Policies $R^*$, *WI* and *LD* are not affected by $V'$.

Two performance metrics are used to evaluate scheduling policies. Figures 3-2, 3-4 and 3-6 measure the Expected Weighted Sum AoI, $\mathbb{E}[J_T^\pi]$, defined in (2.4) and compare it with the lower bound $L_B$ in (3.5a). Figs. 3-3 and 3-5 measure the maximum normalized throughput debt, defined as $\max_i\{x_i^+(T+1)/Tq_i\}$. Figs. 3-2 and 3-3 show simulations of networks

with increasing time-horizon, namely $T \in \{10^4, 5\times10^4, 10^5, 5\times10^5, 10^6, 15\times10^6\}$, and fixed $N = 15$, $\varepsilon = 0.9$, and $V' = 1$. Each data point in Figs. 3-2 and 3-3 is an average over the results of $10^8/T$ simulations. Figs. 3-4 and 3-5 show simulations of networks with increasing size, namely $N \in \{5, 10, \cdots, 25, 30\}$, and fixed $\varepsilon = 0.9$ and $V' = N^2$. Fig. 3-6 shows networks with varying throughput constraints, namely $\varepsilon \in \{0.7, 0.75, \cdots, 0.95, 0.999\}$, and fixed $N = 30$ and $V' = N^2$. Each data point in Figs. 3-4, 3-5 and 3-6 is an average over the results of 10 simulations and each simulation runs for a total of $T = N \times 10^6$ slots.

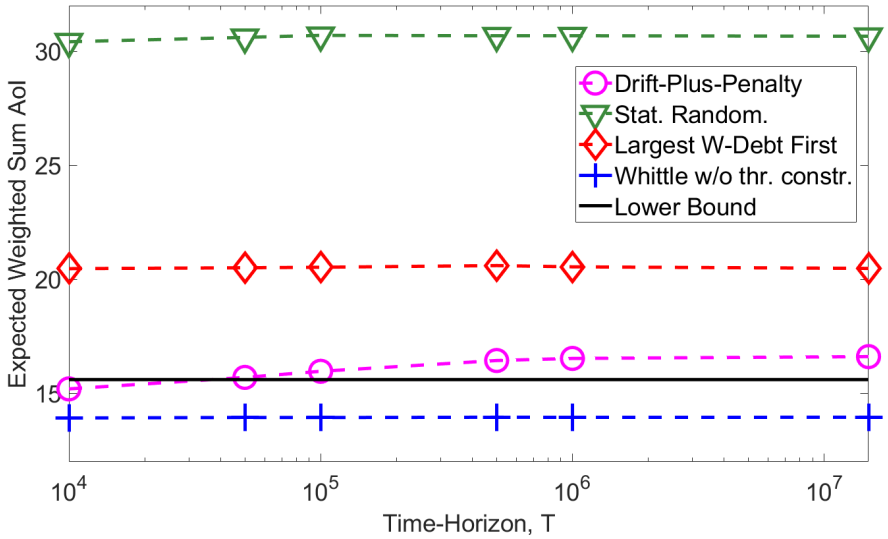



Figure 3-2: Network with increasing time-horizon $T$, $N = 15$, $\varepsilon = 0.9$, and $V' = 1$. The simulation result for each policy and for each value of $T$ is an average over $10^8/T$ runs.

Figs. 3-2 and 3-3 show the effects of low $V'$ on *DPP*. A lower value of $V'$ gives lower priority to the throughput debt and, as a result, the network may take longer to achieve the desired throughput, especially when the number of nodes $N$ and/or $\varepsilon$ are large. This convergence time is illustrated in Fig. 3-3. The advantage of having low $V'$ is the (slight) improvement in EWSAoI. Examining *DPP* in Figs. 3-2 and 3-4, it can be seen that when $V'$ goes from $15^2$ to 1, the EWSAoI of *DPP* decreases from 17.26 to 16.61, i.e. less than 5% improvement when $V'$ decreases by a factor of 225.

As expected, simulations clearly support the theoretical results in Sec. 3.2. Specifically, Figs. 3-2 to 3-6 show that: 1) the AoI performance of $R^*$ is a factor of 2 away from the lower bound; 2) the AoI performance of *DPP* is comparable to the lower bound in every network configuration simulated; 3) $R^*$, *LD*, and *DPP* satisfy the feasible throughput requirements,

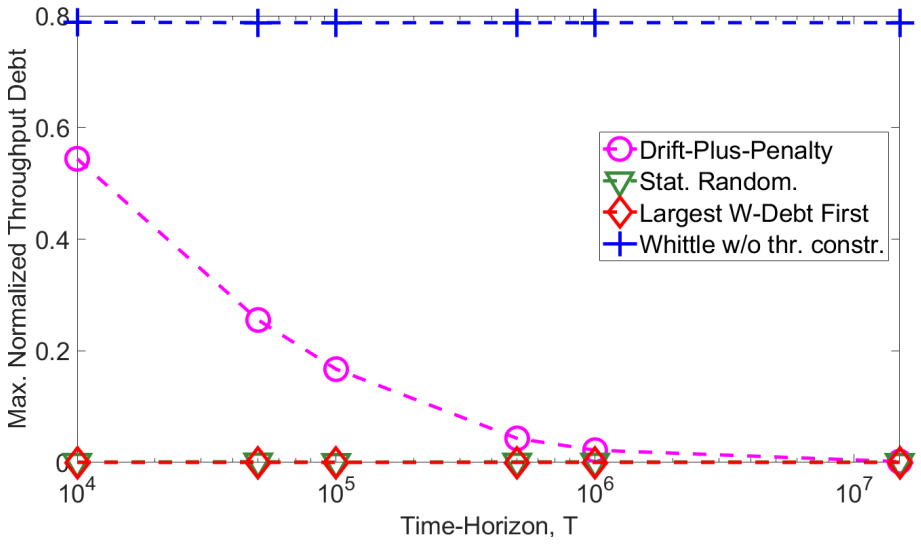



Figure 3-3: Network with increasing time-horizon $T$, $N = 15$, $\varepsilon = 0.9$, and $V' = 1$. The simulation result for each policy and for each value of $T$ is an average over $10^8/T$ runs.

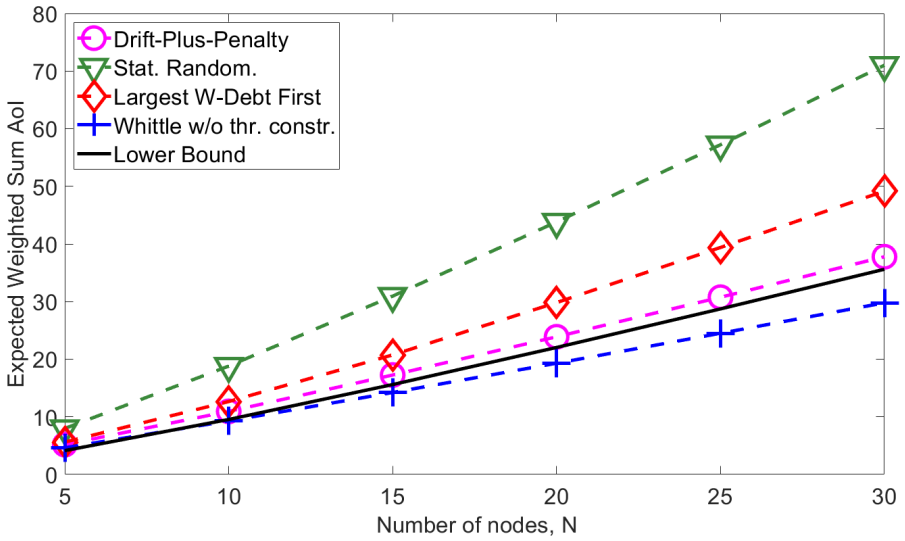



Figure 3-4: Network with increasing size $N$, $T = N \times 10^6$, $\varepsilon = 0.9$, and $V' = N^2$. The simulation result for each policy and for each value of $N$ is an average over 10 runs.

while $WI$ does not satisfy the throughput requirements; and 4) the AoI performance of $WI$ is superior to the lower bound, which is only possible because $WI$ developed in Sec. 2.2.5 does not satisfy the throughput requirements. The performance of $WI$ shows the impact of the throughput requirements on the AoI performance. *We conclude that DPP has superior performance in terms of both AoI and throughput.*

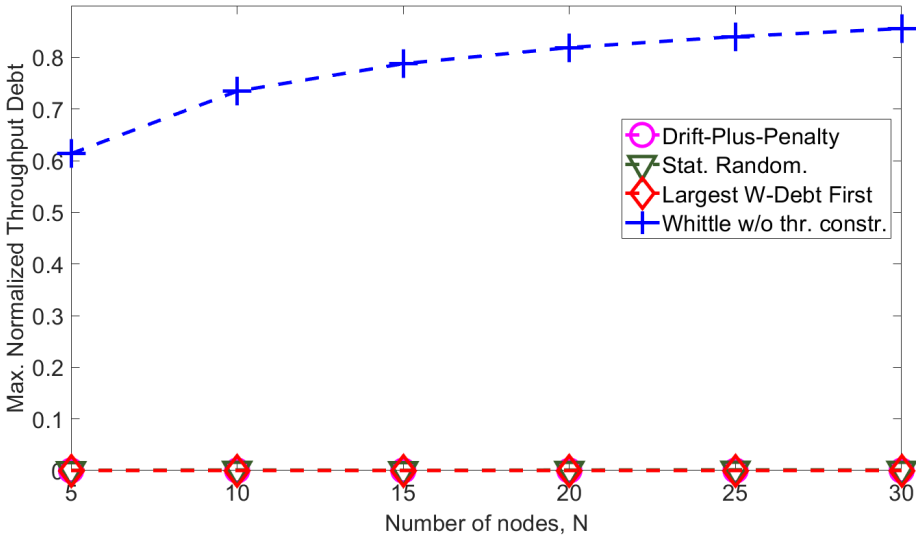



Figure 3-5: Network with increasing size $N$, $T = N \times 10^6$, $\varepsilon = 0.9$, and $V' = N^2$. The simulation result for each policy and for each value of $N$ is an average over 10 runs.

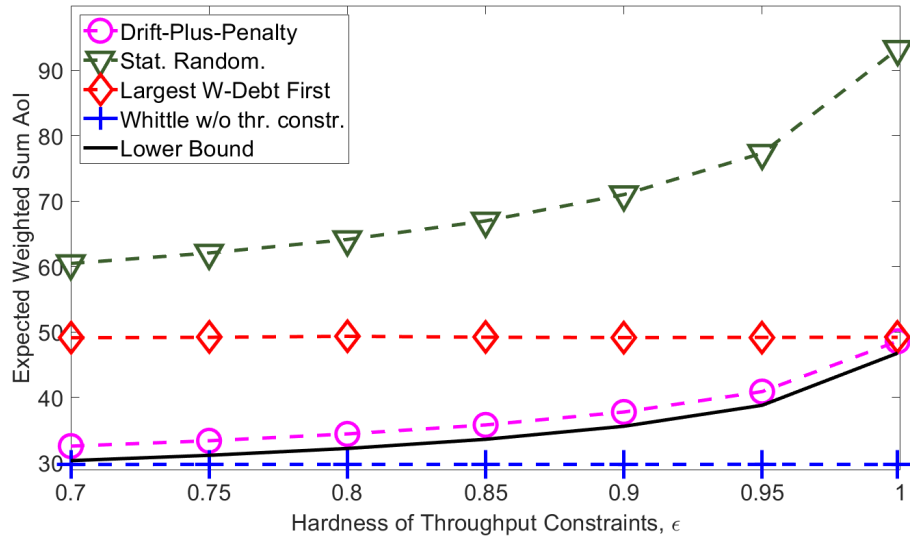



Figure 3-6: Network with increasing $\varepsilon$, $N = 30$, $T = N \times 10^6$, and $V' = N^2$. The simulation result for each policy and for each value of $\varepsilon$ is an average over 10 runs.

## 3.4 Summary

In this chapter, we considered a broadcast single-hop wireless network with sources that generate fresh packets *on demand* and transmit them via unreliable communication links. We addressed the problem of minimizing the Expected Weighted Sum AoI in the network while simultaneously satisfying throughput requirements from the individual nodes.

Throughput requirements can either capture an attribute of the nodes or be used to enforce fair allocation of resources in the network.

First, we derived a lower bound on the AoI performance achievable by any given network. Then, we developed two low-complexity transmission scheduling policies, namely Stationary Randomized and Drift-Plus-Penalty, and showed that both are 2-optimal for any network configuration, while simultaneously satisfying any feasible throughput requirements. Simulation results show that the Drift-Plus-Penalty policy outperforms other scheduling policies in every configuration simulated, and achieves near optimal information freshness.

**Remark 3.11** (Whittle's Index policy for wireless networks with throughput constraints)**.** *Using similar arguments as in Sec. 2.2.5, we can transform the* throughput constrained AoI optimization *in* (3.4a)-(3.4c) *into a relaxed Restless Multi-Armed Bandit (RMAB) problem. Unfortunately, it can be shown that* due to the throughput constraints, $\hat{q}_i^{\pi} \geq q_i$*, this relaxed RMAB problem is not indexable. To overcome this challenge, we can relax the throughput constraints in* (3.4b)*, place them into the objective function of* (3.4a)-(3.4c)*, and propose a Whittle's Index policy associated with the "doubly relaxed" problem. The drawback of this approach is that the resulting Whittle's Index policy does not guarantee that feasible throughput constraints are satisfied. For this reason, the development of the Whittle's Index policy is not included in the body of this chapter. The Whittle's Index policy is developed and discussed in Appendix 3.D. Numerical results can be found in [50, Sec. IV].*

# Appendices

## 3.A Proof of Theorem 3.2 (Lower Bound)

**Theorem 3.2** (Lower Bound). The optimization problem in (3.5a)-(3.5c) provides a lower bound $L_B$ to the AoI optimization (3.4a)-(3.4c), namely $L_B \leq \mathbb{E}[J^*]$ for every network setup $(N, p_i, q_i, w_i)$.

Lower Bound

$$L_B = \min_{\pi \in \Pi} \left\{ \frac{1}{2N} \sum_{i=1}^{N} w_i \left( \frac{1}{\hat{q}_i^{\pi}} + 1 \right) \right\}$$
$$\text{s.t. } \hat{q}_i^{\pi} \geq q_i \ , \forall i \ ;$$
$$\textstyle\sum_{i=1}^{N} u_i(t) \leq 1 \ , \forall t \ .$$

*Proof.* Consider a scheduling policy $\pi \in \Pi$ that satisfies all throughput and interference constraints running on a network for the time-horizon of $T$ slots. Let $\Omega$ be the sample space associated with this network and let $\omega \in \Omega$ be a sample path. For a given sample path $\omega$, the total number of packets delivered by link $i$ during the $T$ slots is denoted $D_i(T) = \sum_{t=1}^{T} d_i(t)$ and the inter-delivery time associated with each of those deliveries is denoted $I_i[m]$. In particular, let $I_i[m]$ be the number of slots between the $(m-1)$th and $m$th packet deliveries from link $i$, $\forall m \in \{1, \cdots, D_i(T)\}$[2]. After the last packet delivery from link $i$, the number of remaining slots is $R_i$. Hence, the time-horizon can be written as

$$T = \sum_{m=1}^{D_i(T)} I_i[m] + R_i \ , \forall i \in \{1, 2, \cdots, N\} \ . \tag{3.37}$$

According to the evolution of $h_i(t)$ in (2.3), the slot that follows the $(m-1)$th packet delivery from link $i$ has an AoI of $h_i(t) = 1$. Since the $m$th packet is delivered only after $I_i[m]$ slots, we know that $h_i(t)$ evolves as $\{1, 2, \cdots, I_i[m]\}$. This pattern is repeated throughout

[2]Naturally, $I_i[1]$ is the number of slots between the first packet delivery from link $i$ and the first slot $t = 1$.

the entire time-horizon, including the last $R_i$ slots. As a result, the time-average Age of Information of link $i$ can be expressed as

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} h_i(t) &= \frac{1}{T}\left[\sum_{m=1}^{D_i(T)} \frac{(I_i[m]+1)I_i[m]}{2} + \frac{(R_i+1)R_i}{2}\right] \\ &= \frac{1}{2}\left[\frac{D_i(T)}{T}\left(\frac{1}{D_i(T)}\sum_{m=1}^{D_i(T)} I_i^2[m]\right) + \frac{R_i^2}{T} + 1\right], \end{aligned} \tag{3.38}$$

where the last equality uses (3.37) to replace the two linear terms by $T$.

Define the operator $\bar{\mathbb{M}}[\mathbf{x}]$ that computes the sample mean of any set $\mathbf{x}$. In particular, let the sample mean of $I_i[m]$ and $I_i^2[m]$ be

$$\bar{\mathbb{M}}[I_i] = \frac{1}{D_i(T)}\sum_{m=1}^{D_i(T)} I_i[m] \quad \text{and} \quad \bar{\mathbb{M}}[I_i^2] = \frac{1}{D_i(T)}\sum_{m=1}^{D_i(T)} I_i^2[m]\,. \tag{3.39}$$

Substituting $\bar{\mathbb{M}}[I_i^2]$ into (3.38) and then applying Jensen's inequality, yields

$$\frac{1}{T}\sum_{t=1}^{T} h_i(t) \geq \frac{1}{2}\left(\frac{D_i(T)}{T}\left(\bar{\mathbb{M}}[I_i]\right)^2 + \frac{R_i^2}{T} + 1\right), \tag{3.40}$$

combining (3.37) and (3.39), and then substituting the result into (3.40), gives

$$\frac{1}{T}\sum_{t=1}^{T} h_i(t) \geq \frac{1}{2}\left(\frac{1}{T}\frac{(T-R_i)^2}{D_i(T)} + \frac{R_i^2}{T} + 1\right). \tag{3.41}$$

By minimizing the RHS of (3.41) analytically with respect to the variable $R_i$, we have

$$\frac{1}{T}\sum_{t=1}^{T} h_i(t) \geq \frac{1}{2}\left(\frac{T}{D_i(T)+1} + 1\right). \tag{3.42}$$

Taking the expectation of (3.42) and applying Jensen's inequality, yields

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[h_i(t)\right] \geq \frac{1}{2}\left(\frac{1}{\mathbb{E}\left[\frac{D_i(T)}{T}\right] + \frac{1}{T}} + 1\right). \tag{3.43}$$

Applying the limit $T \to \infty$ to (3.43) and using the definition of throughput in (3.1), gives

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[h_i(t)\right] \geq \frac{1}{2}\left(\frac{1}{\hat{q}_i^{\pi}} + 1\right) . \tag{3.44}$$

Combining (3.44) and the objective function in (3.4a), yields

$$\begin{aligned} \lim_{T\to\infty} \mathbb{E}\left[J_T^{\pi}\right] &= \lim_{T\to\infty} \frac{1}{N} \sum_{i=1}^{N} \frac{w_i}{T} \sum_{t=1}^{T} \mathbb{E}\left[h_i(t)\right] \\ &\geq \frac{1}{2N} \sum_{i=1}^{N} w_i \left(\frac{1}{\hat{q}_i^{\pi}} + 1\right) . \end{aligned} \tag{3.45}$$

Finally, substituting (3.45) into the AoI optimization (3.4a)-(3.4c) gives the Lower Bound (3.5a)-(3.5c). ■

## 3.B Proof of Theorem 3.9

**Theorem 3.9**. The DPP policy satisfies any feasible set of minimum throughput requirements $\{q_i\}_{i=1}^N$.

*Proof.* The expression of the upper bound in (3.33) is central to the analysis in this appendix and is rewritten below for convenience.

$$\Delta'(S(t))+P'(S(t)) \leq -\sum_{i=1}^{N} \mathbb{E}\left[u_i(t)\,|S(t)\right] W_i'(t) + B'(t)\,,$$

where $W_i'(t)$ and $B'(t)$ are given by

$$W_i'(t) = \frac{\tilde{\alpha}_i p_i}{2} h_i(t) + V' p_i x_i^+(t)\,;$$
$$B'(t) = \sum_{i=1}^{N} \left\{ \frac{\tilde{\alpha}_i}{2}\left[h_i(t)+1\right] + V' x_i^+(t) q_i + \frac{V'}{2} \right\}.$$

Recall that the Drift-Plus-Penalty policy is designed to minimize the RHS of (3.33). Hence, a Stationary Randomized Policy $R \in \Pi_R$ that, in each slot $t$, selects node $i$ with probability $\mu_i \in (0,1]$ yields a lower (or equal) RHS, i.e.

$$\sum_{i=1}^{N} \mathbb{E}\left[u_i(t)\,|S(t)\right] W_i'(t) \geq \sum_{i=1}^{N} \mu_i W_i'(t)\,. \tag{3.46}$$

Substituting (3.46) into the RHS of (3.33) gives

$$\begin{aligned}\Delta'(S(t)) + P'(S(t)) &\leq -\sum_{i=1}^{N} \mu_i W_i'(t) + B'(t) \\ &\leq -\sum_{i=1}^{N} V' x_i^+(t)[\mu_i p_i - q_i] + \frac{1}{2}\sum_{i=1}^{N}[V' + \tilde{\alpha}_i] + \frac{1}{2}\sum_{i=1}^{N} \tilde{\alpha}_i h_i(t)[1-\mu_i p_i]\,,\end{aligned} \tag{3.47}$$

and by substituting the expression of $P'(S(t))$ in (3.23) and rearranging the terms, we get

$$\sum_{i=1}^{N} V' x_i^+(t)[\mu_i p_i - q_i] + \frac{1}{2}\sum_{i=1}^{N} \tilde{\alpha}_i h_i(t)\mu_i p_i \leq \\ \leq \frac{1}{2}\sum_{i=1}^{N}[V' + \tilde{\alpha}_i] - \Delta'(S(t)) - \frac{1}{2}\sum_{i=1}^{N} \tilde{\alpha}_i \mathbb{E}[h_i(t+1) - h_i(t)|S(t)] \,. \tag{3.48}$$

For simplicity of exposition, we divide inequality (3.48) into five terms $LHS'_1 + LHS'_2 \leq RHS'_1 + RHS'_2 + RHS'_3$. Taking their expectation with respect to $S(t)$, summing them over $t \in \{1, 2, \cdots, T\}$ and then dividing them by $TN$, gives

$$LHS'_1 = \frac{1}{N}\sum_{i=1}^{N}(\mu_i p_i - q_i)\frac{V'}{T}\sum_{t=1}^{T}\mathbb{E}[x_i^+(t)] \,; \tag{3.49a}$$

$$LHS'_2 = \frac{1}{2N}\sum_{i=1}^{N}(\tilde{\alpha}_i \mu_i p_i)\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[h_i(t)] \,; \tag{3.49b}$$

$$RHS'_1 = \frac{1}{2N}\sum_{i=1}^{N}[V' + \tilde{\alpha}_i] \,; \tag{3.49c}$$

$$RHS'_2 = \frac{V'}{2NT}\sum_{i=1}^{N}\mathbb{E}\left\{[x_i^+(1)]^2 - [x_i^+(T+1)]^2\right\} \,; \tag{3.49d}$$

$$RHS'_3 = \frac{1}{2NT}\sum_{i=1}^{N}\tilde{\alpha}_i \mathbb{E}[h_i(1) - h_i(T+1)] \,. \tag{3.49e}$$

Notice that the expression of the Lyapunov Drift (3.26) was utilized in $RHS'_2$. Since $h_i(T+1)$ and $x_i^+(T+1)$ are non-negative, the expression of $RHS'_2 + RHS'_3$ can be simplified as follows

$$RHS'_2 + RHS'_3 \leq \frac{1}{2NT}\sum_{i=1}^{N}\mathbb{E}\left\{V'[x_i^+(1)]^2 + \tilde{\alpha}_i h_i(1)\right\} \tag{3.50}$$

Recall that $h_i(1) = 1$ and $x_i(1) = 0$. Hence, in the limit $T \to \infty$, we have $RHS'_2 + RHS'_3 \leq 0$.

Since $LHS'_2$ is non-negative, it follows that the inequality can be reduced to $LHS'_1 \leq RHS'_1 + RHS'_2 + RHS'_3$. Applying the limit $T \to \infty$ and using (3.50) yields

$$\sum_{i=1}^{N}(\mu_i p_i - q_i)\lim_{T\to\infty}\frac{V'}{T}\sum_{t=1}^{T}\mathbb{E}[x_i^+(t)] \leq \frac{1}{2}\sum_{i=1}^{N}[V' + \tilde{\alpha}_i] \tag{3.51}$$

By rearranging the terms, it is easy to see that strong stability holds for any given node $i$, i.e.

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[x_i^+(t)\right] < \infty\,, \tag{3.52}$$

what establishes condition (3.22). ■

## 3.C Proof of Theorem 3.10 (Optimality Ratio for DPP)

**Theorem 3.10** (Optimality Ratio for *DPP*). For any wireless network with parameters $(N, p_i, q_i, w_i)$, by choosing the constants $\tilde{\alpha}_i = w_i/\mu_i^* p_i, \forall i$, the optimality ratio of *DPP* is such that

$$\rho^{DPP} \leq 2 + \frac{1}{L_B}\left[V' - \frac{1}{N}\sum_{i=1}^{N} w_i\right] .$$

Moreover, if $V' \leq \sum_{i=1}^{N} w_i/N$, then the *DPP* policy is 2-optimal.

*Proof.* Consider the analysis in Appendix 3.B. In particular, the inequality $LHS'_1 + LHS'_2 \leq RHS'_1 + RHS'_2 + RHS'_3$ presented in (3.49a)-(3.49e). Given that $LHS'_1$ is non-negative, it follows that the inequality can be reduced to $LHS'_2 \leq RHS'_1 + RHS'_2 + RHS'_3$. Then, applying the limit $T \to \infty$ and using (3.50) yields

$$\sum_{i=1}^{N} (\tilde{\alpha}_i \mu_i p_i) \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[h_i(t)] \leq \sum_{i=1}^{N} \left[V' + \tilde{\alpha}_i\right] \tag{3.53}$$

Analogously to the proof of Theorem 3.5, let $\hat{q}_i^L$ be the long-term throughput associated with the policy that solves the Lower Bound optimization (3.5a)-(3.5c). Then, evaluating $L_B$ from (3.5a) gives

$$L_B = \frac{1}{2N}\sum_{i=1}^{N} \frac{w_i}{\hat{q}_i^L} + \frac{1}{2N}\sum_{i=1}^{N} w_i \, . \tag{3.54}$$

Now, for each node $i$, we impose the following scheduling probability $\mu_i = \hat{q}_i^L/p_i$ and constant $\tilde{\alpha}_i = w_i/\hat{q}_i^L$. Then, evaluating (3.53) gives

$$\mathbb{E}\left[J^{DPP}\right] \leq \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{\hat{q}_i^L} + V' \, . \tag{3.55}$$

Comparing (3.54) and (3.55), yields

$$L_B \leq \mathbb{E}\left[J^{DPP}\right] \leq 2L_B + \left[V' - \frac{1}{N}\sum_{i=1}^{N} w_i\right] ; \tag{3.56}$$

$$\rho^{DPP} \leq 2 + \frac{1}{L_B}\left[V' - \frac{1}{N}\sum_{i=1}^{N} w_i\right] . \tag{3.57}$$

Recall from Corollary 3.6 that $\hat{q}_i^L = \mu_i^* p_i$. Hence, we know that $\tilde{\alpha}_i = w_i/\mu_i^* p_i, \forall i$. The proof is complete. ■

## 3.D Whittle's Index policy

In this appendix, we develop a Whittle's Index policy by transforming the throughput constrained AoI optimization (3.4a)-(3.4c) into a relaxed Restless Multi-Armed Bandit (RMAB) problem. This is possible because the AoI associated with each link in the network evolves as a restless bandit. To obtain the relaxed RMAB problem, we substitute the $T$ interference constraints $\sum_{i=1}^{N} u_i(t) \leq 1, \forall t$, in (3.4c) by the single time-averaged constraint

$$\frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{E}[u_i(t)] \leq \frac{1}{N} \,. \tag{3.58}$$

Then, we relax this time-averaged constraint, by placing (3.58) into the objective function (3.4a) together with the associated Lagrange Multiplier $C \geq 0$. The resulting optimization problem is called relaxed RMAB and its solution lays the foundation for the design of Whittle's Index. A detailed description of this method can be found in [25, 110].

One of the challenges associated with this method is that Whittle's Index is only defined for problems that are *indexable*. Unfortunately, it can be shown that *due to the throughput constraints, $\hat{q}_i^{\pi} \geq q_i$, the relaxed RMAB resulting from the transformation of the throughput constrained AoI optimization* (3.4a)-(3.4c) *is not indexable*.

To overcome this challenge, we first relax the throughput constraints (3.4b), placing them into the objective function of (3.4a)-(3.4c) as follows

**Relaxed AoI Optimization**

$$\mathbb{E}\left[\widetilde{J}^{*}\right] = \min_{\pi \in \Pi} \left\{ \lim_{T \to \infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \left[ w_i \mathbb{E}\left[h_i(t)\right] + \theta_i \left( \frac{q_i}{p_i} - \mathbb{E}[u_i(t)] \right) \right] \right\} \tag{3.59a}$$

$$\text{s.t.} \;\; \theta_i \geq 0 \,, \forall i \,; \tag{3.59b}$$

$$\sum_{i=1}^{N} u_i(t) \leq 1 \,, \forall t \,. \tag{3.59c}$$

Each Lagrange Multiplier $\theta_i$ is associated with a relaxation of $\hat{q}_i^{\pi} \geq q_i$. These multipliers are called *throughput incentives* for they represent the penalty incurred by scheduling policies that deviate from the corresponding throughput constraint. Now, applying the

transformation described at the beginning of this section to the relaxed AoI optimization (3.59a)-(3.59c) yields

**Doubly relaxed RMAB**

$$\mathbb{E}\left[J^{DR*}\right] = \min_{\pi\in\Pi}\left\{\lim_{T\to\infty}\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}\left[w_i\mathbb{E}\left[h_i(t)\right]+(C-\theta_i)\mathbb{E}\left[u_i(t)\right]-\frac{C}{N}+\frac{\theta_i q_i}{p_i}\right]\right\} \tag{3.60a}$$

$$\text{s.t. } \theta_i \geq 0\,, \forall i\,; \tag{3.60b}$$

$$C \geq 0\,. \tag{3.60c}$$

Next, we solve the doubly relaxed RMAB, establish that the relaxed AoI optimization is indexable and obtain a closed-form expression for the Whittle's Index.

The doubly relaxed RMAB is separable and thus can be solved for each individual link. Observe that a scheduling policy running on a network with a single link $i$ can only choose between selecting link $i$ for transmission during slot $t$ or idling. The scheduling policy that optimizes (3.60a)-(3.60c) for a given link $i$ is characterized next.

**Proposition 3.12** (Threshold policy)**.** *Consider the doubly relaxed RMAB problem* (3.60a)-(3.60c) *associated with a single link i. The optimal scheduling policy is a threshold policy that, in each slot t, selects link i when* $h_i(t) \geq H_i$ *and idles when* $1 \leq h_i(t) < H_i$*. For positive fixed values of C and* $\theta_i$*: 1) if* $C > \theta_i$*, then the expression for the threshold is*

$$H_i = \left\lfloor \frac{3}{2} - \frac{1}{p_i} + \sqrt{\left(\frac{1}{p_i}-\frac{1}{2}\right)^2 + \frac{2(C-\theta_i)}{w_i p_i}} \right\rfloor\,; \tag{3.61}$$

*2) otherwise, if* $C \leq \theta_i$*, then the threshold is* $H_i = 1$*.*

The proof of Proposition 3.12 follows similar arguments as the proof of Proposition 2.17. Next, we define the condition for indexability and establish that the relaxed AoI optimization is indexable. For a given value of $C$, let $\mathscr{I}_i(C) = \{h_i(t) \in \mathbb{N} | h_i(t) < H_i\}$ be the set of

states $h_i(t)$ in which the threshold policy idles. The doubly relaxed RMAB associated with link $i$ is indexable if the set $\mathscr{I}_i(C)$ increases monotonically from $\emptyset$ to $\mathbb{N}$, as the value of $C$ increases from 0 to $+\infty$. Furthermore, the relaxed AoI optimization is indexable if this condition holds for all links. The condition on $\mathscr{I}_i(C)$ follows directly from Proposition 3.12 and is true for all links $i$. *Thus, we establish that the relaxed AoI optimization is indexable.*

Given indexability, we define Whittle's Index. Let $C_i(h_i(t))$ be the Whittle's Index associated with link $i$ in state $h_i(t)$. By definition, $C_i(h_i(t))$ is the *infimum value of* $\mathbf{C}$ that makes both scheduling decisions (transmit or idle) equally desirable to the threshold policy in state $h_i(t)$. The scheduling decisions are equally desirable in state $h_i(t)$ when the multiplier $C$ is such that $H_i = h_i(t)+1$. Using (3.61) to solve this equation for the value of $C$ gives the following expression for the Index

$$C_i(h_i(t)) = \frac{w_i p_i h_i(t)}{2}\left[h_i(t) + \frac{2}{p_i} - 1\right] + \theta_i \,. \tag{3.62}$$

After establishing indexability and obtaining the expression for $C_i(h_i(t))$, we define Whittle's Index policy.

**Definition 3.13** (Whittle's Index policy)**.** *The Whittle's Index policy selects, in each slot $t$, the link with highest value of $C_i(h_i(t))$, with ties being broken arbitrarily.*

Denote the Whittle's Index policy as $WI$. Next, we derive the performance guarantee $\rho^{WI}$.

**Theorem 3.14** (Optimality Ratio for $WI$)**.** *For any given network setup $(N, p_i, q_i, w_i)$, the optimality ratio of $WI$ is such that*

$$\rho^{WI} \leq 8 + \frac{1}{L_B}\left[\frac{1}{N}\sum_{i=1}^{N}\theta_i - \frac{7}{2N}\sum_{i=1}^{N} w_i\right] \,. \tag{3.63}$$

*In particular, for every network with $\sum_{i=1}^{N}\theta_i \leq 7\sum_{i=1}^{N} w_i/2$, the Whittle's Index policy is 8-optimal.*

The proof of Theorem 3.14 is provided in [50, Appendix G]. The arguments used for deriving $\rho^{WI}$ are similar to the ones for deriving $\rho^{DPP}$ in Theorem 3.10. Those similarities come from the fact that policies *DPP* and *WI* are alike. Comparing the expressions for $W_i(t)$ and $C_i(h_i(t))$, in (3.34) and (3.62), respectively, we can see that: 1) both have an AoI term, $W_i(t)$ has $\tilde{\alpha}_i p_i h_i(t)/2$ and $C_i(h_i(t))$ has $w_i p_i h_i(t)\left[h_i(t)+\frac{2}{p_i}-1\right]/2$; and 2) both have a throughput term, $W_i(t)$ has $V' p_i x_i^+(t)$ and $C_i(h_i(t))$ has $\theta_i$. Naturally, we expect the performance of both policies to be similar in terms of AoI. *The key difference between DPP and WI lies in the throughput term. While the term $V' p_i x_i^+(t)$ guarantees that DPP satisfies the throughput constraint, $\hat{q}_i^{\pi} \geq q_i$, the positive scalar $\theta_i$ represents an incentive for WI to comply with the constraint, but provides no guarantee.* The benefit of using a fixed $\theta_i$ is that there is no need to keep track of $x_i^+(t)$ for each link and at every slot $t$.

The results in this section *hold for any given set of positive throughput incentives* $\{\theta_i\}_{i=1}^N$. Next, we propose an algorithm that finds the values of $\theta_i$ which maximize a lower bound on the Lagrange Dual problem associated with the relaxed AoI optimization (3.59a)-(3.59c). Observe that $\mathbb{E}\left[J^{DR*}\right]$ in (3.60a) is the Lagrange Dual function associated with (3.59a)-(3.59c). Thus, we can define the Lagrange Dual problem as $\max_{C,\theta_i}\{\mathbb{E}\left[J^{DR*}\right]\}$ subject to $C \geq 0$ and $\theta_i \geq 0, \forall i$. Since this dual problem is challenging to address, we consider a lower bound:

$$\max_{C,\chi_i}\{\widetilde{\mathscr{L}}(C,\chi_i)\} \leq \max_{C,\theta_i}\{\mathbb{E}\left[J^{DR*}\right]\} \leq \mathbb{E}\left[J^*\right] . \tag{3.64}$$

subject to $\chi_i = C - \theta_i$, $C \geq 0$ and $\theta_i \geq 0$ for all links $i$, where

$$\widetilde{\mathscr{L}}(C,\chi_i) = \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{p_i} - \frac{C}{N}\left[1-\sum_{i=1}^{N}\frac{q_i}{p_i}\right] + \sum_{i=1}^{N}\frac{w_i}{N}\left[\sqrt{\frac{2\chi_i}{w_i p_i}+\left[\frac{1}{p_i}-\frac{1}{2}\right]^2} - \frac{\chi_i q_i}{w_i p_i} - \frac{1}{p_i} - \frac{1}{2}\right].$$

The throughput incentives $\theta_i$ that result from the maximization of $\widetilde{\mathscr{L}}(C,\chi_i)$ are given by Algorithm 2. Simulation results in [50] show that the values of $\{\theta_i^*\}_{i=1}^N$ from Algorithm 2 reduce the throughput debt when compared to $\theta_i = 0$.

**Algorithm 2** Throughput Incentives

1: $\chi_i \leftarrow w_i p_i[(1/q_i)^2 - (1/p_i - 1/2)^2]/2 \ , \forall i$
2: $C \leftarrow \max_i\{\chi_i\}$
3: $\phi_i^{-1} \leftarrow p_i\sqrt{2\min\{C;\chi_i\}/(w_i p_i) + (1/p_i - 1/2)^2} \ , \forall i$
4: $S \leftarrow \phi_1 + \phi_2 + \cdots + \phi_N$
5: **while** $S < 1$ **do**
6: decrease $C$ slightly
7: repeat steps 3 and 4 to update $\phi_i$ and $S$
8: **end while**
9: $C^* = C$ and $\chi_i^* = \min\{C^*;\chi_i\}$ and $\theta_i^* = C^* - \chi_i^*, \forall i$
10: **return** $(\theta_1^*, \theta_2^*, \cdots, \theta_N^*)$

Chapter 4

# AoI in Wireless Networks with Stochastic Arrivals

In this chapter, we consider a broadcast single-hop wireless network with a base station (BS) and a number of nodes sharing time-sensitive information through unreliable communication links, as illustrated in Fig. 4-1. A key difference from chapters 2 and 3 is that *fresh packets cannot be generated on demand*. Packets from each stream arrive according to a stochastic process and are enqueued in a separate (per stream) queue. The queueing discipline controls which packet within each queue is available for transmission. The scheduling policy decides, at every time $t$, which stream to serve to the corresponding destination. Our goal is to develop scheduling policies that keep the information fresh at every destination, i.e. that minimize the average AoI in the network.

In Sec. 2.2.2, it was shown that when the network is symmetric[1] and streams can generate fresh packets on demand, the optimal scheduling policy serves the stream associated with the largest AoI, namely $\arg\max\{h_i(t)\}$. This policy is optimal for it gives the largest reduction in AoI over all streams. However, when packet generation is random, streams may not have fresh packets available for transmission. Thus, a scheduling policy must account both for the AoI at the destinations and the time-stamps of the packets available for transmission in each queue. For example, consider a simple network with two streams

[1] A network is symmetric when all nodes have identical channel reliability $p_i = p$ and weight $w_i = w$.

and two destinations. Assume that at time $t$, each stream has a single packet in its queue. The packet from stream 1 was generated 30 msecs ago and the packet from stream 2 was generated 10 msecs ago. Assume that the current AoI at destinations 1 and 2 are $h_1(t) = 50$ msecs and $h_2(t) = 40$ msecs, respectively. A policy that serves the stream associated with the largest AoI would select stream 1 and yield an AoI reduction of $50 - 30 = 20$ msecs. Alternatively, serving stream 2 would result in a reduction of $40 - 10 = 30$ msecs. Hence, to minimize the average AoI, it is optimal to schedule stream 2. In this simple example, the optimal scheduling decision was easily determined. In general, designing a transmission scheduling policy that keeps information fresh over time is a challenging task that needs to take into account the packet arrival process, the queueing discipline, and the conditions of the wireless channels.

*In this chapter, we address the problem of optimizing link scheduling in networks with stochastic packet arrivals and unreliable channels operating under three common queueing disciplines.* In particular, we derive a lower bound on the AoI performance achievable by any given network operating under any queueing discipline. Then, we consider three common queueing disciplines and develop both a Randomized policy and a Max-Weight policy under each discipline. *Our approach allows us to evaluate the combined impact of the stochastic arrivals, queueing discipline and scheduling policy on AoI.* We evaluate the AoI performance both analytically and using simulations. Numerical results show that the performance of the Max-Weight policy with Last-Come First-Served queues is close to the analytical lower bound.

This chapter generalizes chapter 2. The main difference is that in previous chapters we assume that when the BS selects a stream, a new packet with fresh information is generated and then transmitted to the corresponding destination in the same time-slot. In that case, the packet delay is always 1 slot and the AoI is reduced to $h(t) = 1$ slot after every packet delivery. In contrast, in this chapter, we consider a network in which packets are generated according to a stochastic process and are enqueued before being transmitted. This seemingly modest distinction affects the packet delay and the evolution of AoI over time, which in turn affects the results and proofs throughout the chapter significantly. To illustrate the technical differences, we briefly compare the approaches taken for analyzing

Stationary Randomized policies and Max-Weight policies.

- Under the assumptions in previous chapters, the AoI evolution is stochastically renewed after every packet delivery, since $h(t) = 1$, and thus the AoI can be analyzed by directly applying the elementary renewal theorem for renewal-reward processes. In contrast, in this chapter, the evolution of AoI may be dependent across consecutive inter-delivery intervals and, thus, the same approach is not applicable. To analyze the AoI, we obtain the stationary distribution of a two-dimensional Markov Chain with countably-infinite state space in Proposition 4.5.

- The Max-Weight policy and Drift-Plus-Penalty policy in previous chapters make scheduling decisions based on AoI only. In contrast, in this chapter, the Max-Weight policy selects streams based on the "AoI reduction" accrued from a successful packet delivery, which is a function of both AoI and packet delay.

Beyond the fact that chapter 2 represents a special case of this chapter, in particular a network with LCFS queues and fresh packet arrivals at every decision time, the results are different in themselves and required different proof techniques due to the challenges imposed by the stochastic arrivals, queueing disciplines and packet delay.

The remainder of this chapter is organized as follows. In Sec. 4.1, we describe the network model. In Sec. 4.2 we derive an analytical lower bound on the AoI minimization problem. In Sec. 4.3, we develop the Optimal Stationary Randomized policy for each queueing discipline and characterize their AoI performance. In Sec. 4.4, we develop the Max-Weight policy and obtain performance guarantees in terms of AoI. In Sec. 4.5, we provide numerical results. A summary of results is provided in Sec. 4.6.

## 4.1 System Model

Consider a wireless network with a base station (BS) and $N$ nodes sharing time-sensitive information through unreliable communication links, as illustrated in Fig. 4-1. At the beginning of every slot $t$, a new packet from stream $i \in \{1, 2, \cdots, N\}$ arrives to the system with probability $\lambda_i \in (0, 1], \forall i$. Let $a_i(t) \in \{0, 1\}$ be the indicator function that is equal to 1 when

a packet from stream $i$ arrives in slot $t$, and $a_i(t) = 0$ otherwise. This Bernoulli arrival process is i.i.d. over time and independent across different streams, with $\mathbb{P}(a_i(t) = 1) = \lambda_i, \forall i, t$.

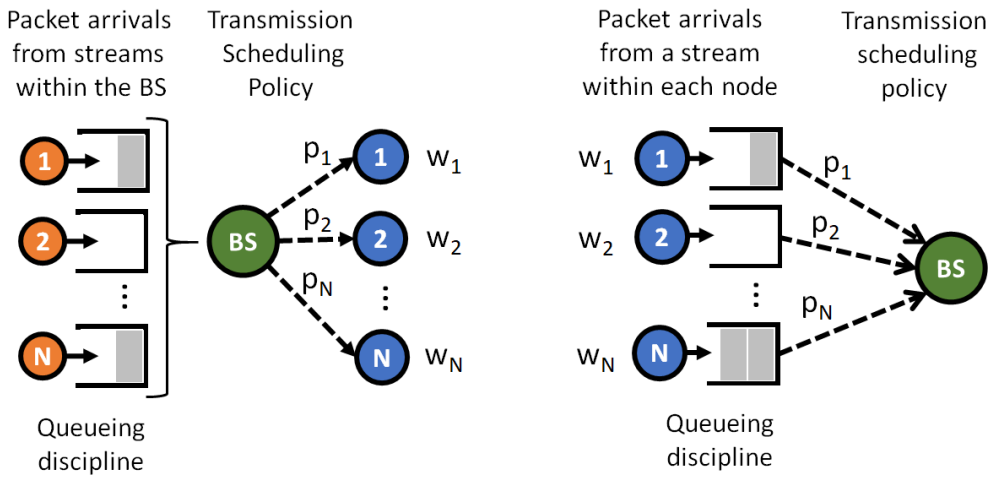



Figure 4-1: Illustration of the single-hop wireless network. On the left, we have a downlink network with the BS serving multiple traffic streams to different destinations. On the right, we have an uplink network with multiple sources transmitting different traffic streams to the BS. The network model in this section comprehends both scenarios.

Packets from stream $i$ are enqueued in queue $i$. Denote by Head-of-Line (HoL) packets the set of packets *from all queues* that are available to the BS for transmission in a given slot $t$. Depending on the queueing discipline employed by the network, queues can be of three types:

(i) *First-Come First-Served (FCFS) queues*: packets are served in order of arrival. The HoL packets in slot $t$ are the oldest packets in each queue. This is a standard queueing discipline, widely deployed in communication systems.

(ii) *Single packet queues*: when a new packet arrives, older packets from the same stream are dropped from the queue. The HoL packets in slot $t$ are the freshest (i.e. most recently generated) packets in each queue. This queueing discipline is known to minimize the AoI in a variety of contexts. From the perspective of the AoI, Single packet queues are equivalent to Last-Come First-Served (LCFS) queues;

(iii) *No queues*: packets can be transmitted only duing the slot in which they arrive. The HoL packets in slot $t$ are given by the set $\{i|a_i(t) = 1\}$.

Let $z_i(t)$ represent the system time of the HoL packet in queue $i$ at the beginning of slot $t$. By definition, we have $z_i(t) := t - \tau_i^A(t)$, where $\tau_i^A(t)$ is the arrival time of the HoL packet in queue $i$. Naturally, the value of $\tau_i^A(t)$ changes only when the HoL packet changes, namely when the current HoL packet is served or dropped and there is another packet in the same queue; or when the queue is empty and a new packet arrives. Notice that $z_i(t)$ is undefined when queue $i$ is empty.

We denote by $z_i^F(t)$, $z_i^S(t)$ and $z_i^N(t)$, the system times associated with *FCFS queues*, *Single packet queues* and *No queues*, respectively. For all three cases, whenever the system time is defined, it evolves according to the definition $z_i(t) := t - \tau_i^A(t)$. Moreover, it follows from the description of the queueing disciplines that the evolution of $z_i^S(t)$ can be written as

$$z_i^S(t) = \begin{cases} 0 & \text{if } a_i(t) = 1; \\ z_i^S(t-1) + 1 & \text{otherwise,} \end{cases} \tag{4.1}$$

and the evolution of $z_i^N(t)$ is such that $z_i^N(t) = 0$ whenever an arrival occurs, i.e. $a_i(t) = 1$, and is undefined otherwise. In contrast, the evolution of $z_i^F(t)$ cannot be simplified for it depends on both the arrival times and service times of packets in the queue.

In each slot $t$, the BS either idles or selects a stream and transmits its HoL packet to the corresponding destination over the wireless channel. Let $u_i(t) \in \{0,1\}$ be the indicator function that is equal to 1 when the BS transmits the HoL packet from stream $i$ during slot $t$, and $u_i(t) = 0$ otherwise[2]. The BS can transmit at most one packet at any given time-slot $t$. Hence, we have

$$\sum_{i=1}^{N} u_i(t) \leq 1, \forall t \ . \tag{4.2}$$

The transmission scheduling policy governs the sequence of decisions $\{u_i(t)\}_{i=1}^N$ of the BS.

Let $c_i(t) \in \{0,1\}$ represent the channel state associated with destination $i$ during slot $t$. When the channel is *ON*, we have $c_i(t) = 1$, and when the channel is *OFF*, we have $c_i(t) = 0$. The channel state process is i.i.d. over time and independent across different destinations, with $\mathbb{P}(c_i(t) = 1) = p_i, \forall i,t$.

[2]In previous chapters, when the BS selects stream $i$ a fresh packet is generated and transmitted over the associated link. However, when packets are generated randomly, the BS can select streams with no packets available for transmission. Notice that the indicator is $u_i(t) = 1$ only when a packet from stream $i$ is *transmitted* during slot $t$.

Let $d_i(t) \in \{0,1\}$ be the indicator function that is equal to 1 when destination $i$ successfully receives a packet during slot $t$, and $d_i(t) = 0$ otherwise. A successful reception occurs when the HoL packet is transmitted and the associated channel is ON, implying that $d_i(t) = c_i(t)u_i(t), \forall i,t$. Moreover, since the BS does not know the channel states prior to making scheduling decisions, $u_i(t)$ and $c_i(t)$ are independent, and

$$\mathbb{E}[d_i(t)] = p_i \mathbb{E}[u_i(t)], \forall i,t \ . \tag{4.3}$$

The scheduling policies considered in this chapter are non-anticipative, i.e. policies that do not use future information in making scheduling decisions. Let $\Pi$ be the class of non-anticipative policies and let $\pi \in \Pi$ be an arbitrary admissible policy. Our goal is to develop scheduling policies $\pi$ that minimize the average AoI in the network. Next, we formulate the AoI minimization problem.

Let $h_i(t)$ be the AoI associated with destination $i$ at the beginning of slot $t$. By definition, we have $h_i(t) := t - \tau_i^D(t)$, where $\tau_i^D(t)$ is the arrival time of the freshest packet delivered to destination $i$ before slot $t$. If during slot $t$ destination $i$ receives a packet with system time $z_i(t) = t - \tau_i^A(t)$ such that $\tau_i^A(t) > \tau_i^D(t)$, then in the next slot we have $h_i(t+1) = z_i(t) + 1$. Alternatively, if during slot $t$ destination $i$ does not receive a *fresher packet*, then the information gets one slot older, which is represented by $h_i(t+1) = h_i(t) + 1$. Notice that the three queueing disciplines considered in this chapter select HoL packets with increasing freshness, implying that $\tau_i^A(t) > \tau_i^D(t)$ holds[3] for every received packet. Hence, the AoI evolves as follows:

$$h_i(t+1) = \begin{cases} z_i(t) + 1 & \text{if } d_i(t) = 1; \\ h_i(t) + 1 & \text{otherwise,} \end{cases} \tag{4.4}$$

for simplicity, and without loss of generality, we assume that $h_i(1) = 1$ and $z_i(0) = 0, \forall i$. Substituting $z_i^F(t)$, $z_i^S(t)$ and $z_i^N(t)$ into (4.4) we obtain the AoI associated with *FCFS queues*, *Single packet queues* and *No queues*, respectively. In Fig. 4-2 we illustrate the evolution of $h_i(t)$ and $z_i(t)$ in a network employing *Single packet queues*.

[3]One example of a queueing discipline that can violate $\tau_i^A(t) > \tau_i^D(t)$ is the Last-Come First-Served (LCFS) queue. When an older packet with $\tau_i^A(t) \leq \tau_i^D(t)$ is delivered, the associated AoI does not decrease and the network runs as if no packet was delivered. It follows that, from the perspective of the AoI, *LCFS queues* are equivalent to *Single packet queues*.

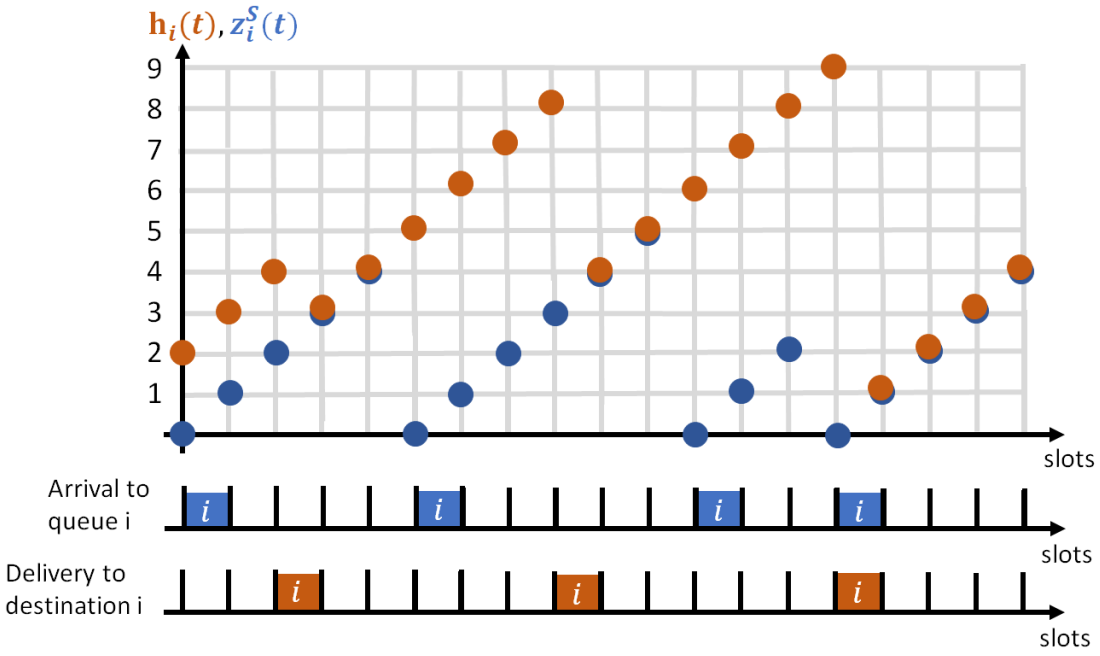



Figure 4-2: The blue and orange rectangles represent a packet arrival to queue $i$ and a successful packet delivery to destination $i$, respectively. The blue circles shows the evolution of $z_i(t)$ for the *Single packet queue* and the orange circles shows the AoI associated with destination $i$.

The time-average AoI associated with destination $i$ is given by $\mathbb{E}\left[\sum_{t=1}^{T} h_i(t)\right]/T$. For capturing the freshness of the information of a network employing scheduling policy $\pi \in \Pi$, we define the Expected Weighted Sum AoI (EWSAoI) in the limit as the time-horizon grows to infinity as

$$\mathbb{E}\left[J^{\pi}\right] = \lim_{T\to\infty} \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} w_i \mathbb{E}\left[h_i^{\pi}(t)\right] , \tag{4.5}$$

where $w_i$ is a positive real number that represents the priority of stream $i$. We denote by AoI-optimal, the scheduling policy $\pi^* \in \Pi$ that achieves minimum EWSAoI, namely

$$\mathbb{E}[J^*] = \min_{\pi\in\Pi} \mathbb{E}\left[J^{\pi}\right] , \tag{4.6}$$

where the expectation is with respect to the randomness in the channel state $c_i(t)$, scheduling decisions $u_i(t)$ and arrival process $a_i(t)$. Next, we introduce the long-term throughput and discuss the stability of FCFS queues.

### 4.1.1 Long-term Throughput

Recall from Sec. 3.1 that the long-term throughput associated with destination $i$ is defined as

$$\hat{q}_i^{\pi} := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[d_i^{\pi}(t)\right] . \tag{4.7}$$

Throughout this section, we assume that $\hat{q}_i^{\pi} > 0, \forall i$. Since packets from stream $i$ are generated at a rate $\lambda_i$, the long-term throughput provided to destination $i$ cannot be higher than $\lambda_i$. Hence, the long-term throughput satisfies

$$\hat{q}_i^{\pi} \leq \lambda_i, \forall i . \tag{4.8}$$

The shared and unreliable wireless channel further restricts the set of achievable values of long-term throughput $\{\hat{q}_i^{\pi}\}_{i=1}^{N}$. By employing (4.3) and (4.2) into the definition of long-term throughput in (4.7), we obtain

$$\sum_{i=1}^{N} \frac{\hat{q}_i^{\pi}}{p_i} \leq 1 . \tag{4.9}$$

Inequalities (4.8) and (4.9) are necessary conditions for the long-term throughput $\{\hat{q}_i^{\pi}\}_{i=1}^{N}$ of any admissible scheduling policy $\pi \in \Pi$, regardless of the queueing discipline. *Notice that conditions* (4.8) *and* (4.9) *are not throughput requirements imposed by the nodes as in chapter 3. Both inequalities are necessary conditions that follow naturally from the stochastic arrivals and interference constraints of the network.* Next, we discuss the stability of FCFS queues and its impact on the AoI minimization problem.

### 4.1.2 Queue Stability

Let $Q_i^{\pi}(t)$ be the number of packets in queue $i$ at the beginning of slot $t$ when policy $\pi$ is employed. Then, we say that queue $i$ is *stable* if

$$\lim_{T\to\infty} \mathbb{E}\left[Q_i^{\pi}(T)\right] < \infty . \tag{4.10}$$

A network is stable under policy $\pi$ when all of its queues are stable. For networks with *Single packet queues* and *No queues*, stability is trivial since the backlogs are such that $Q_i^{\pi}(t) \in \{0,1\}, \forall t$, regardless of the scheduling policy. The discussion about queue stability that follows is meaningful only for the case of *FCFS queues*.

**Definition 4.1** (Stability Region)**.** *A set of arrival rates $\{\lambda_i\}_{i=1}^N$ is within the stability region of a given wireless network if there exists an admissible scheduling policy $\pi \in \Pi$ that stabilizes all queues.*

When the network is unstable under a policy $\eta \in \Pi$, then the expected backlog of at least one of its queues grows indefinitely over time. An infinitely large backlog leads to packets with infinitely large system times, i.e. $z_i(t) \to \infty$. It follows from the evolution of $h_i(t)$ in (4.4) that the AoI also increases indefinitely and, as a result, the Expected Weighted Sum AoI diverges, namely $\mathbb{E}[J^{\eta}] \to \infty$. Clearly, instability is a critical disadvantage for *FCFS queues*. Hence, we are interested in scheduling policies that can stabilize the network whenever the arrival rates $\{\lambda_i\}_{i=1}^N$ are within the stability region. Prior to introducing the policies, we derive a lower bound to the AoI minimization problem.

## 4.2 Universal Lower Bound

In this section, we derive an alternative (and more insightful) expression for the AoI objective function $J^{\pi}$ in (4.5) in terms of packet delay and inter-delivery times. Then, we use this expression to obtain a lower bound to the AoI minimization problem, namely $L_B \leq \mathbb{E}[J^*]$, for any given network operating under an arbitrary queueing discipline.

Consider a network employing policy $\pi$ during the time-horizon $T$. Let $\Omega$ be the sample space associated with this network and let $\omega \in \Omega$ be a sample path. For a given sample path $\omega$, let $t_i[m]$ be the index of the time-slot in which the $m$th (fresher[4]) packet was delivered to destination $i$, $\forall m \in \{1, \cdots, D_i(T)\}$, where $D_i(T)$ is the total number of packets delivered

[4]Recall that the delivery of an older packet with $\tau_i^A(t) \leq \tau_i^D(t)$ does not change the associated AoI and, thus, should not be counted.

up to (and including) slot $T$. Then, we define $I_i[m] := t_i[m] - t_i[m-1]$ as the *inter-delivery time*, with $I_i[1] = t_i[1]$ and $t_i[0] = 0$.

The *packet delay* associated with the $m$th packet delivery to destination $i$ is given by $z_i(t_i[m])$. Notice that $z_i(t_i[m])$ is the system time of the HoL packet at the time it is delivered to the destination, which is the definition of *packet delay*. To simplify notation, we use $z_i[m]$ instead of $z_i(t_i[m])$.

Recall from (2.14) that $\bar{\mathbb{M}}[\mathbf{x}]$ is the operator that calculates the sample mean of a set of values $\mathbf{x}$. Using this operator, the sample mean of $I_i[m]$ for a fixed destination $i$ is given by

$$\bar{\mathbb{M}}[I_i] = \frac{1}{D_i(T)} \sum_{m=1}^{D_i(T)} I_i[m] \, . \tag{4.11}$$

For simplicity of notation, the time-horizon $T$ is omitted in the sample mean operator $\bar{\mathbb{M}}$.

**Proposition 4.2.** *The infinite-horizon AoI objective function $J^\pi$ can be expressed as follows*

$$J^\pi = \lim_{T\to\infty} \sum_{i=1}^{N} \frac{w_i}{2N} \left[ \frac{\bar{\mathbb{M}}[I_i^2]}{\bar{\mathbb{M}}[I_i]} + \frac{2\bar{\mathbb{M}}[z_i I_i]}{\bar{\mathbb{M}}[I_i]} + 1 \right] \; w.p.1 \, , \tag{4.12}$$

*where $I_i[m]$ is the inter-delivery time, $z_i[m]$ is the packet delay and*

$$\bar{\mathbb{M}}[z_i I_i] = \frac{1}{D_i(T)} \sum_{m=1}^{D_i(T)} z_i[m-1] I_i[m] \, . \tag{4.13}$$

*Proof Outline.* Consider the interval between the $(m-1)$th and $m$th packet delivery to destination $i$. The time slots in this interval are $t \in \{t_i[m-1]+1, t_i[m-1]+2, \cdots, t_i[m-1]+I_i[m]\}$, where $t_i[m-1]+I_i[m] = t_i[m]$ is the time slot in which the $m$th packet is delivered to destination $i$. According to the evolution of $h_i(t)$ in (4.4), in slot $t_i[m-1]+1$, we have $h_i(t_i[m-1]+1) = z_i[m-1]+1$. Then, in the inter-delivery period $I_i[m]$, the AoI increases by unity at every slot, until it reaches $h_i(t_i[m]) = z_i[m-1]+I_i[m]$. Hence, the expected sum

AoI in the interval $I_i[m]$ is given by

$$\sum_{t=t_i[m-1]+1}^{t_i[m-1]+I_i[m]} h_i(t) = z_i[m-1]I_i[m] + \frac{I_i[m]\,(I_i[m]+1)}{2}\,. \tag{4.14}$$

Recall that for $m=1$ we have $t_i[0]=0$ and $z_i(0)=0, \forall i$. An equivalent expression can be obtained for the interval $R_i$.

Using a similar approach as in Theorem 2.1, we substitute the sum (4.14) into the objective function in (4.5) to obtain the equivalent form in (4.12). The complete proof is provided in Appendix 4.A. ■

Equation (4.12) is valid for networks operating under an *arbitrary queueing discipline* and employing *any scheduling policy* $\pi \in \Pi$. This equation provides useful insights into the AoI minimization. The first term on the RHS of (4.12), namely $\bar{\mathbb{M}}[I_i^2]/2\bar{\mathbb{M}}[I_i]$, depends only on the service regularity provided by the scheduling policy. The second term on the RHS of (4.12) depends on both the packet delay $z_i[m-1]$ and the inter-delivery time $I_i[m]$, as follows

$$\frac{\bar{\mathbb{M}}[z_i I_i]}{\bar{\mathbb{M}}[I_i]} = \sum_{m=1}^{D_i(T)} \frac{I_i[m]}{\sum_{j=1}^{D_i(T)} I_i[j]} z_i[m-1]\,. \tag{4.15}$$

Notice that (4.15) is a weighted sample mean of the packet delays. Intuitively, for minimizing this term, both the queueing discipline and the scheduling policy should attempt to deliver packets with low delay $z_i[m-1]$ and, when the delay is high, they should deliver the next packet as soon as possible in order to reduce the weight $I_i[m]$ on the weighted mean (4.15).

The expression in (4.12) provides intuition on how the scheduling policy should manage the packet delays $z_i[m]$ and the inter-delivery times $I_i[m]$ in order to minimize AoI. Moreover, it shows that by utilizing the simplifying assumption of queues always having fresh packets available for transmission, the scheduling policy disregards $z_i[m]$ and fails to address the term in (4.15). Next, we use (4.12) to obtain a lower bound to the AoI minimization problem and, in upcoming sections, we consider scheduling policies that take into account both $I_i[m]$ and $z_i[m]$.

A lower bound on AoI is obtained from Proposition 4.2 using similar arguments as in

the proof of Theorem 3.2. In particular, we apply Jensen's inequality $\bar{\mathbb{M}}[I_i^2] \geq (\bar{\mathbb{M}}[I_i])^2$ to (4.12), manipulate the resulting expression, and then employ a minimization over policies in $\Pi$, which yields

Lower Bound

$$L_B = \min_{\pi \in \Pi} \left\{ \frac{1}{2N} \sum_{i=1}^{N} w_i \left( \frac{1}{\hat{q}_i^{\pi}} + 1 \right) \right\} \tag{4.16a}$$

$$\text{s.t. } \sum_{i=1}^{N} \hat{q}_i^{\pi} / p_i \leq 1 \, ; \tag{4.16b}$$

$$\hat{q}_i^{\pi} \leq \lambda_i, \forall i \, , \tag{4.16c}$$

where (4.16b) and (4.16c) are the necessary conditions for the long-term throughput in (4.9) and (4.8), respectively. Notice that the optimization problem in (4.16a)-(4.16c) depends only on the network's long-term throughput $\{\hat{q}_i^{\pi}\}_{i=1}^{N}$ and that the condition $\hat{q}_i^{\pi} \leq \lambda_i$ limits the throughput to the packet arrival rate of the respective stream. To find the unique solution to (4.16a)-(4.16c), we analyze the associated KKT Conditions.

**Theorem 4.3** (Lower bound)**.** *For any given wireless network with parameters* $(N, p_i, \lambda_i, w_i)$ *and an arbitrary queueing discipline, the optimization problem in* (4.16a)-(4.16c) *provides a lower bound on the AoI minimization problem, namely* $L_B \leq \mathbb{E}[J^*]$. *The unique solution to* (4.16a)-(4.16c) *is given by*

$$\hat{q}_i^{L_B} = \min \left\{ \lambda_i, \sqrt{\frac{w_i p_i}{2N\gamma^*}} \right\}, \forall i \, , \tag{4.17}$$

*where* $\gamma^*$ *yields from Algorithm 3. The lower bound is given by*

$$L_B = \frac{1}{2N} \sum_{i=1}^{N} w_i \left( \frac{1}{\hat{q}_i^{L_B}} + 1 \right) \, . \tag{4.18}$$

To find the unique solution to the KKT Conditions, we use a similar technique as in Theorem 3.7. The complete proof is provided in Appendix 4.B.

**Algorithm 3** Solution to the Lower Bound

1: $\tilde{\gamma} \leftarrow (\sum_{i=1}^{N} \sqrt{w_i/p_i})^2/(2N)$ and $\gamma_i \leftarrow w_i p_i / 2N\lambda_i^2, \forall i$
2: $\gamma \leftarrow \max\{\tilde{\gamma}; \gamma_i\}$
3: $q_i \leftarrow \lambda_i \min\{1; \sqrt{\gamma_i/\gamma}\}, \forall i$
4: $S \leftarrow \sum_{i=1}^{N} q_i/p_i$
5: **while** $S < 1$ **and** $\gamma > 0$ **do**
6: decrease $\gamma$ slightly
7: repeat steps 3 and 4 to update $q_i$ and $S$
8: **end while**
9: **return** $\gamma^* = \gamma$ and $\hat{q}_i^{L_B} = q_i, \forall i$

Next, we develop the Optimal Stationary Randomized policy for different queueing disciplines and derive the closed-form expression for their AoI performance.

## 4.3 Stationary Randomized Policies

Denote by $\Pi_R$ the class of Stationary Randomized policies and let $R \in \Pi_R$ be a policy that makes scheduling decisions randomly, according to fixed probabilities $\{\mu_i\}_{i=1}^{N}$, where $\mu_i = \mathbb{E}[u_i(t)] \in (0,1], \forall i, \forall t$, and $\mu_{idle} = 1 - \sum_{i=1}^{N} \mu_i$.

**Definition 4.4** (Randomized policy)**.** *The Randomized policy selects, in each slot t, stream i with probability $\mu_i$, or selects no stream with probability $\mu_0$.*

If the selected stream $i$ has a non-empty queue, then $u_i(t) = 1$ and the HoL packet is transmitted by the BS to destination $i$. Alternatively, if the selected stream $i$ has an empty queue or policy $R$ selected no stream, then $u_i(t) = 0, \forall i$ and the BS idles. The scheduling probabilities $\mu_i$ are fixed over time and satisfy $\sum_{i=1}^{N} \mu_i = 1 - \mu_0$.

Randomized policies $R \in \Pi_R$ are as simple as possible. Each policy in $\Pi_R$ is fully characterized by the set $\{\mu_i\}_{i=1}^{N}$. They select streams at random, without taking into account $h_i(t)$, $z_i(t)$ or queue backlogs $Q_i(t)$. Notice that policies in $\Pi_R$ are not work-conserving, since they allow the BS to idle during slots in which HoL packets are available for transmission.

Despite their simplicity, we show that by *properly tuning the scheduling probabilities* $\mu_i$ according to the network parameters $(N, p_i, \lambda_i, w_i)$, policies in $\Pi_R$ can achieve performances within a factor of 4 from the AoI-optimal. On the other hand, we also show that naive choices of $\mu_i$ can lead to poor AoI performances. Next, we develop and analyze scheduling policies for different queueing disciplines which are optimal over the class $\Pi_R$. In Secs. 4.3.1, 4.3.2 and 4.3.3 we consider networks employing *Single packet queues*, *No queues* and *FCFS queues*, respectively. Then, in Sec. 4.3.4 we compare their AoI performances.

### 4.3.1 Randomized Policy for Single packet queue

Consider a network employing the *Single packet queue* discipline on $N$ streams with packet arrival rates $\lambda_i$, priorities $w_i$ and channel reliabilities $p_i$. Recall that for the *Single packet queue*, when a new packet arrives, older packets from the same stream are dropped. The BS selects streams according to $R \in \Pi_R$ with scheduling probabilities $\mu_i$.

Following a successful packet transmission from stream $i$, its queue can remain empty or a new packet can arrive. The expected number of (consecutive) slots that queue $i$ remains empty is $1/\lambda_i - 1$. When a new packet arrives to the queue, the BS transmits this packet with probability $\mu_i$. The expected number of slots necessary to successfully deliver this packet is $1/p_i\mu_i$. Under policy $R \in \Pi_R$ and for the case of *Single packet queues*, the sequence of packet deliveries is a renewal process. It follows from the elementary renewal theorem [23] that the long-term throughput is given by

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[d_i^R(t)] = \frac{1}{1/p_i\mu_i + 1/\lambda_i - 1}, \forall i, t \ . \tag{4.19}$$

For the particular case of $\lambda_i = 1$, the AoI process $h_i(t)$ is also stochastically renewed after every packet delivery and the long-term time-average $\mathbb{E}[h_i(t)]$ can be easily obtained using the elementary renewal theorem for renewal-reward processes, as in (2.45). In contrast, for the general case of $\lambda_i \in (0,1]$, the evolution of $h_i(t)$ may be dependent across consecutive inter-delivery intervals due to its relationship with the system time $z_i^S(t)$ given in (4.4). To find an expression for the long-term time-average $\mathbb{E}[h_i(t)]$ we formulate the

problem as a two-dimensional Markov Chain (MC) with countably-infinite state space represented by $(h_i(t), z_i(t))$ and obtain the stationary distribution. Proposition 4.5 follows from the stationary distribution of this two-dimensional MC.

**Proposition 4.5.** *The optimal EWSAoI achieved by a network with Single packet queues over the class $\Pi_R$ is given by (4.20a)-(4.20b), where $R^S$ denotes the Optimal Stationary Randomized policy for the Single packet queue discipline.*

Optimal Randomized policy for Single packet queues

$$\mathbb{E}\left[J^{R^S}\right] = \min_{R\in\Pi_R}\left\{\frac{1}{N}\sum_{i=1}^{N} w_i\left(\frac{1}{\lambda_i} - 1 + \frac{1}{p_i\mu_i}\right)\right\} \tag{4.20a}$$

$$\text{s.t. } \textstyle\sum_{i=1}^{N}\mu_i \leq 1\,; \tag{4.20b}$$

The complete proof is provided in Appendix 4.C. Next, we solve the optimization problem in (4.20a)-(4.20b) and obtain the optimal scheduling probabilities $\{\mu_i^S\}_{i=1}^N$.

**Theorem 4.6.** *Consider a wireless network with parameters $(N, p_i, \lambda_i, w_i)$ operating under the Single packet queues discipline. The optimal scheduling probabilities are given by*

$$\mu_i^S = \frac{\sqrt{w_i/p_i}}{\sum_{j=1}^{N}\sqrt{w_j/p_j}}, \forall i\,, \tag{4.21}$$

*and the performance of the Optimal Stationary Randomized policy $R^S$ is*

$$\mathbb{E}\left[J^{R^S}\right] = \frac{1}{N}\sum_{i=1}^{N} w_i\left(\frac{1}{\lambda_i} - 1\right) + \frac{1}{N}\left(\sum_{i=1}^{N}\sqrt{\frac{w_i}{p_i}}\right)^2. \tag{4.22}$$

*Then, it follows that*

$$\mathbb{E}[J^*] \leq \mathbb{E}\left[J^{R^S}\right] < 4\mathbb{E}[J^*]\,, \tag{4.23}$$

*where $\mathbb{E}[J^*] = \min_{\pi\in\Pi}\mathbb{E}[J^\pi]$ is the minimum AoI over the class of all admissible policies $\Pi$.*

*Proof.* The scheduling probabilities $\{\mu_i^S\}_{i=1}^N$ that minimize (4.20a)-(4.20b) also minimize this equivalent problem

$$\min_{R\in\Pi_R}\left\{\frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{p_i\mu_i}\right\} \text{ s.t. } \sum_{i=1}^{N}\mu_i\leq 1\ . \tag{4.24}$$

Consider the Cauchy–Schwarz inequality

$$\left(\sum_{i=1}^{N}\sqrt{\frac{w_i}{p_i}}\right)^2\leq\left(\sum_{i=1}^{N}\mu_i\right)\left(\sum_{i=1}^{N}\frac{w_i}{p_i\mu_i}\right)\ . \tag{4.25}$$

The LHS is a lower bound on the objective function in (4.24). Notice that Cauchy-Schwarz holds with equality when $\{\mu_i^S\}_{i=1}^N$ is given by (4.21), implying that (4.21) is a solution to both (4.24) and (4.20a)-(4.20b). Substituting the solution $\{\mu_i^S\}_{i=1}^N$ into the objective function in (4.20a) gives (4.22).

Notice that the expression in (4.21) was obtained in Sec. 2.2.3 under the simplifying assumption of all streams always having fresh packets available for transmission. In Theorem 4.6 we show that (4.21) is in fact optimal for streams with stochastic packet arrivals and for any set of arrival rates $\{\lambda_i\}_{i=1}^N$.

For deriving the upper bound in (4.23), consider the Randomized policy $\tilde{R}$ with $\tilde{\mu}_i = \hat{q}_i^{L_B}/p_i, \forall i$. Substitute $\tilde{\mu}_i$ into the RHS of (4.20a) and denote the result as $\mathbb{E}[J^{\tilde{R}}]$. Comparing $L_B$ in (4.18) with $\mathbb{E}[J^{\tilde{R}}]$ and noting from (4.17) that $\hat{q}_i^{L_B}\leq\lambda_i$, gives that

$$\mathbb{E}\left[J^{\tilde{R}}\right]\leq\frac{1}{N}\sum_{i=1}^{N}w_i\left(\frac{2}{p_i\tilde{\mu}_i}-1\right)<4L_B\ . \tag{4.26}$$

By definition, we know that

$$L_B\leq\mathbb{E}[J^*]\leq\mathbb{E}[J^{R^S}]\leq\mathbb{E}[J^{\tilde{R}}]\ . \tag{4.27}$$

Inequality (4.23) follows directly from (4.26) and (4.27). ■

*Intuitively, the optimal probabilities $\{\mu_i\}_{i=1}^N$ should vary with the packet arrival rates $\{\lambda_i\}_{i=1}^N$.* For example, consider a *Single packet queue* with low arrival rate and high

scheduling probability. This queue is often offered service while empty, thus wasting resources. Hence, it seems natural that the optimal $\mu_i$ should vary with $\lambda_i$. In Secs. 4.3.2 and 4.3.3, we show that this is the case for *No queues* and *FCFS queues*. However, Theorem 4.6 shows that *for* Single packet queues *the optimal* $\mu_i^S$ *depends only on* $w_i$ *and* $p_i$. *This result is important for it simplifies the design of networked systems that attempt to minimize AoI*, as discussed in Sec. 4.3.4.

### 4.3.2 Randomized Policy for No queue

Consider a network with parameters $(N, p_i, \lambda_i, w_i)$ employing the *No queue* discipline and a Stationary Randomized policy $R \in \Pi_R$ with scheduling probabilities $\mu_i$. Recall that $R$ is oblivious to packet arrivals and that, under the *No queue* discipline, packets are available for transmission only during the slot in which they arrive to the system. Hence, if $R$ selects stream $i$ during slot $t$, a successful packet delivery occurs only if a packet from stream $i$ arrived at the beginning of slot $t$, i.e. $a_i(t) = 1$, and the channel is ON, i.e. $c_i(t) = 1$. Therefore, for the *No queue* discipline, we have that $d_i(t) = a_i(t)c_i(t)u_i(t), \forall i,t$. This is equivalent to a network with a *virtual channel* that is ON with probability $p_i\lambda_i$ and OFF with probability $1 - p_i\lambda_i$. We use this equivalence to derive the results that follow.

**Proposition 4.7.** *The optimal EWSAoI achieved by a network with No queues over the class* $\Pi_R$ *is given by* (4.28a)-(4.28b)*, where* $R^N$ *denotes the Optimal Stationary Randomized policy for the No queues discipline.*

Optimal Randomized policy for No queues

$$\mathbb{E}\left[J^{R^N}\right] = \min_{R\in\Pi_R}\left\{\frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{p_i\mu_i\lambda_i}\right\} \tag{4.28a}$$

$$\text{s.t. } \textstyle\sum_{i=1}^{N}\mu_i \leq 1\,; \tag{4.28b}$$

*Proof.* Under the *No queues* discipline, all packets are delivered with system time $z_i^N(t) = 0$ and the AoI process $h_i(t)$ is renewed after every packet delivery. Hence, it follows from the

elementary renewal theorem for renewal-reward processes that

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[h_i(t)]=\frac{1}{p_i\mu_i\lambda_i}\,. \tag{4.29}$$

Substituting (4.29) into (4.6) gives (4.28a). ■

**Theorem 4.8.** *Consider a wireless network with parameters* $(N,p_i,\lambda_i,w_i)$ *operating under the No queues discipline. The optimal scheduling probabilities are given by*

$$\mu_i^N=\frac{\sqrt{w_i/p_i\lambda_i}}{\sum_{j=1}^{N}\sqrt{w_j/p_j\lambda_j}},\forall i\,, \tag{4.30}$$

*and the performance of the Optimal Stationary Randomized policy* $R^N$ *is*

$$\mathbb{E}\left[J^{R^N}\right]=\frac{1}{N}\left(\sum_{i=1}^{N}\sqrt{\frac{w_i}{p_i\lambda_i}}\right)^2. \tag{4.31}$$

*Proof.* The proof follows similar steps as in Theorem 4.6. ■

As expected, the similarities between the Optimal Stationary Randomized policies for the *No queue* and *Single packet queue* disciplines increase as the packet arrival rates $\{\lambda_i\}_{i=1}^N$ increase. In particular, notice from (4.21) and (4.30) that $\mu_i^N=\mu_i^S,\forall i$, when $\lambda_i=1,\forall i$, and, as a result, their AoI performance is also identical, namely $\mathbb{E}\left[J^{R^N}\right]=\mathbb{E}\left[J^{R^S}\right]$ when $\lambda_i=1,\forall i$. Recall that $\mu_i^S$ does not change with $\lambda_i$.

### 4.3.3 Randomized Policy for FCFS queue

Consider a network with parameters $(N,p_i,\lambda_i,w_i)$ employing *FCFS queues* and a Stationary Randomized policy $R\in\Pi_R$ with scheduling probabilities $\mu_i$. In this setting, each *FCFS queue* behaves as a *discrete-time Ber/Ber/1 queue* with arrival rate $\lambda_i$ and service rate $p_i\mu_i$. From [30, Sec. 8.10], we know that the *FCFS queue* is *stable* when $p_i\mu_i>\lambda_i$ and that its

steady-state expected backlog is given by

$$\lim_{T\to\infty} \mathbb{E}\left[Q_i(T)\right] = \frac{\lambda_i(1-p_i\mu_i)}{p_i\mu_i-\lambda_i} \,. \tag{4.32}$$

From [104, Theorem 5], we know that the AoI associated with a *stable FCFS queue* is given by

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[h_i(t)] = \frac{1}{p_i\mu_i} + \frac{1}{\lambda_i} + \left[\frac{\lambda_i}{p_i\mu_i}\right]^2 \frac{1-p_i\mu_i}{p_i\mu_i-\lambda_i} \,. \tag{4.33}$$

Notice the similarities between (4.33), the expected backlog in (4.32) and the AoI associated with a *Single packet queue* in (4.20a). Under light load, i.e. when $\lambda_i << p_i\mu_i$, the third term on the RHS of (4.33) is small when compared to the other terms. Hence, the AoI of the *FCFS queue* in (4.33) is similar to the AoI of the *Single packet queue* in (4.20a). On the other hand, under heavy load, as $\lambda_i \to p_i\mu_i$, the third term on the RHS of (4.33) dominates. Both the backlog and the AoI of the *FCFS queue*, in (4.32) and (4.33), respectively, increase sharply. Recall that when the backlog is large, packets have to wait for a long time in the queue before being served, what makes their information stale and, as a result, the AoI large. The *Single packet queue* discipline avoids this issue by keeping only the freshest packet in the queue.

Denote by $R^F$ the Optimal Stationary Randomized policy for the case of *FCFS queues* and let $\{\mu_i^F\}_{i=1}^N$ be the associated scheduling probabilities. Substituting (4.33) into the expression for the EWSAoI in (4.6) gives

**Optimal Randomized policy for FCFS queues**

$$\mathbb{E}\left[J^{R^F}\right] = \min_{R\in\Pi_R} \left\{ \sum_{i=1}^{N} \frac{w_i}{N} \left[ \frac{1}{p_i\mu_i} + \frac{1}{\lambda_i} + \left[\frac{\lambda_i}{p_i\mu_i}\right]^2 \frac{1-p_i\mu_i}{p_i\mu_i-\lambda_i} \right] \right\} \tag{4.34a}$$

$$\text{s.t. } \textstyle\sum_{i=1}^{N} \mu_i \leq 1 \,; \tag{4.34b}$$

$$p_i\mu_i > \lambda_i, \forall i \,. \tag{4.34c}$$

where (4.34b) is the constraint on scheduling decisions and (4.34c) is the condition for network stability.

**Remark 4.9.** *A sufficient condition for* $\{\lambda_i\}_{i=1}^N$ *to be within the stability region of the*

*network is given by* $\sum_{i=1}^{N} \lambda_i/p_i < 1$.

**Theorem 4.10.** *The optimal scheduling probabilities for the case of FCFS queues* $\mu_i^F$ *are given by Algorithm 4 when* $\delta \to 0$.

*Proof.* The auxiliary parameter $\delta > 0$ is used to enforce a closed feasible set to the optimization problem in (4.34a)-(4.34c). We exchange (4.34c) by $p_i\mu_i \geq \lambda_i + \delta, \forall i$, to ensure that Algorithm 4 always finds a unique solution to the KKT Conditions associated with (4.34a)-(4.34c) for any fixed (and arbitrarily small) value of $\delta$. Recall that when $p_i\mu_i \approx \lambda_i$ the AoI performance is poor. Hence, in most cases, the optimal scheduling probabilities $\{\mu_i^F\}_{i=1}^N$ are such that $p_i\mu_i^F$ and $\lambda_i$ are not close, meaning that small changes in $\delta$ should not affect the solution. Algorithm 4 finds the unique solution to the KKT Conditions and is developed using a similar method as in Theorem 3.7. ■

As part of Algorithm 4, we use the partial derivative of (4.33) with respect to $\mu_i$ multiplied by $w_i/N$, which is denoted as

$$g_i(x) = \frac{w_i}{N}\left\{\frac{\lambda_i}{p_i\mu_i^2}\left[\frac{2}{p_i\mu_i} - 1\right] - \frac{p_i(1-\lambda_i)}{(p_i\mu_i - \lambda_i)^2}\right\}_{x=\mu_i} \tag{4.35}$$

**Algorithm 4** Randomized policy for FCFS queue

1: $\gamma_i \leftarrow (\lambda_i + \delta)/p_i \,, \forall i \in \{1, 2, \cdots, N\}$
2: $\gamma \leftarrow \max_i\{-g_i(\gamma_i)\}$ ▷ where $g_i(.)$ is given in (4.35)
3: $\mu_i \leftarrow \max\{\ \gamma_i\ ;\ g_i^{-1}(-\gamma)\ \}$
4: $S \leftarrow \mu_1 + \mu_2 + \cdots + \mu_N$
5: **while** $S < 1$ **do**
6: decrease $\gamma$ slightly
7: repeat steps 3 and 4 to update $\mu_i$ and $S$
8: **end while**
9: **return** $\mu_i^F = \mu_i, \forall i$

### 4.3.4 Comparison of Queueing Disciplines

Next, we compare the performance of four different Stationary Randomized policies: 1) Optimal policy for *Single packet queues*, $R^S$; 2) Optimal policy for *No queues*, $R^N$; 3) Optimal policy for *FCFS queues*, $R^F$; and 4) Naive policy for *FCFS queues*. The EWSAoI of the first three policies is computed using (4.22), (4.31) and the solution to (4.34a)-(4.34c), respectively. The Naive policy shares resources evenly between streams by assigning $\mu_i = 1/N, \forall i$. The EWSAoI of the Naive policy is computed using the expression inside the minimization in (4.34a).

We consider a network with two streams, $w_1 = w_2 = 1$, $p_1 = 1/3$, $p_2 = 1$, $\lambda_1 = \lambda$, $\lambda_2 = \lambda/3$ and varying arrival rates $\lambda \in \{0.01, 0.02, \cdots, 1\}$. In Fig. 4-3, we show the EWSAoI of Randomized policies under different queueing disciplines and display the Lower Bound $L_B$ for comparison. The policy with *Single packet queues* outperforms the policies with other queueing disciplines for every arrival rate $\lambda$, as expected.

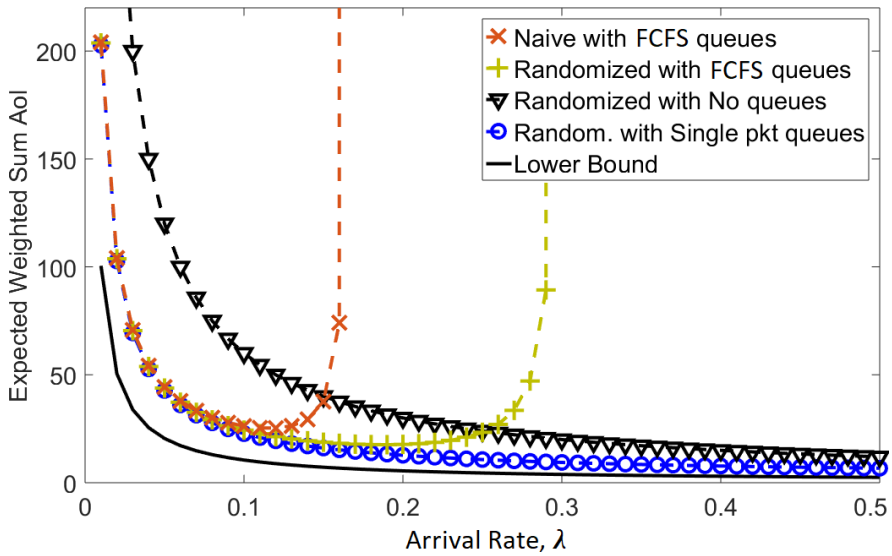



Figure 4-3: Comparison of Stationary Randomized policies in a network with $N = 2$ streams, $w_1 = w_2 = 1$, $p_1 = 1/3$, $p_2 = 1$, $\lambda_1 = \lambda$, $\lambda_2 = \lambda/3$ and increasing $\lambda$.

From Remark 4.9, we know that the network in Fig. 4-3 can be stabilized for $\lambda < 0.3$. The Optimal policy for *FCFS queues* leverages its knowledge of $p_i$ and $\lambda_i$ to stabilize the network, and simultaneously minimize AoI, whenever $\{\lambda_i\}_{i=1}^N$ is within the stability region.

By comparing the performance of the Optimal policy and the Naive policy, it becomes evident that stability is critical for *FCFS queues*.

Both the *Single packet queue* and the *No queue* disciplines present a natural relationship between the rate at which fresh information is generated at the source $\lambda_i$ and the resulting AoI at the destination, namely a higher arrival rate (always) leads to a lower AoI. Furthermore, Theorem 4.6 shows that the optimal scheduling probabilities $\mu_i^S$ for *Single packet queues* are independent of $\lambda_i$. *This result allows us to isolate the design of the arrival rate $\lambda_i$ from the design of the scheduling probability $\mu_i$.* In particular, to minimize the EWSAoI in the network, the arrival rates $\{\lambda_i\}_{i=1}^N$ should be set as high as possible, while the scheduling probabilities $\{\mu_i^S\}_{i=1}^N$ should be proportional to $\sqrt{w_i/p_i}$ according to (4.21). *Since arrival rates and scheduling policies are often defined by different layers of the network stack, this isolation simplifies the design of networked systems. It is important to emphasize that this isolation only holds for networks employing* Single packet queues. *For* FCFS queues *and* No queues *the optimal value of $\mu_i$ changes for different values of $\lambda_i$.* Next, we develop Max-Weight policies that use the knowledge of $h_i(t)$ and $z_i(t)$ for making scheduling decisions in an adaptive manner.

## 4.4 Max-Weight Policies

In this section, we use Lyapunov Optimization [85] to develop Max-Weight policies for each of the queueing disciplines. The Max-Weight policy is designed to reduce the expected drift of the Lyapunov Function at every slot $t$. In doing so, the Max-Weight policy attempts to minimize the AoI of the network.

We use the following linear Lyapunov Function

$$L\left(\{h_i(t)\}_{i=1}^N\right) = L(t) = \frac{1}{N}\sum_{i=1}^{N} \tilde{\alpha}_i h_i(t)\,, \tag{4.36}$$

where $\tilde{\alpha}_i$ is a positive hyperparameter that can be used to tune the Max-Weight policy to different network configurations and queueing disciplines. The Lyapunov Drift is defined

as

$$\Delta(\mathbb{S}(t)) := \mathbb{E}\left[L(t+1) - L(t) | \mathbb{S}(t)\right] , \tag{4.37}$$

where $\mathbb{S}(t) = (\{h_i(t)\}_{i=1}^N, \{z_i(t)\}_{i=1}^N)$ is the network state at the beginning of time slot $t$. The Lyapunov Function $L(t)$ increases with the AoI of the network and the Lyapunov Drift $\Delta(\mathbb{S}(t))$ represents the expected increase of $L(t)$ in one slot. Hence, by minimizing the drift in (4.37) at every slot $t$, the Max-Weight policy is attempting to keep both $L(t)$ and the network's AoI small.

To develop the Max-Weight policy, we analyze the expression for the drift in (4.37). Substituting the evolution of $h_i(t+1)$ from (4.4) into (4.37) and then manipulating the resulting expression, we obtain

$$\Delta(\mathbb{S}(t)) = \frac{1}{N}\sum_{i=1}^{N} \tilde{\alpha}_i - \frac{1}{N}\sum_{i=1}^{N} \tilde{\alpha}_i p_i \left(h_i(t) - z_i(t)\right) \mathbb{E}\left[u_i(t) | \mathbb{S}(t)\right] . \tag{4.38}$$

The scheduling decision in slot $t$ affects only the second term on the RHS of (4.38). The Max-Weight policy minimizes $\Delta(\mathbb{S}(t))$ at every slot $t$.

**Definition 4.11** (Max-Weight policy)**.** *The Max-Weight policy selects, in each slot $t$, the stream $i$ with a HoL packet and the highest value of $\tilde{\alpha}_i p_i \left(h_i(t) - z_i(t)\right)$, with ties being broken arbitrarily.*

Observe that the Max-Weight policy is work-conserving since it idles only when all queues are empty. Substituting $z_i^S(t)$, $z_i^N(t)$ and $z_i^F(t)$ into $\tilde{\alpha}_i p_i \left(h_i(t) - z_i(t)\right)$ gives the Max-Weight policy associated with the *Single packet queue*, $MW^S$, the *No queue*, $MW^N$, and the *FCFS queue*, $MW^F$, respectively. Notice that the difference $h_i(t) - z_i(t)$ represents the AoI reduction accrued from a successful packet delivery to destination $i$. Hence, it makes sense that the Max-Weight policy prioritizes queues with high potential reward $h_i(t) - z_i(t)$.

**Theorem 4.12** (Performance Bounds for $MW^S$)**.** *Consider a wireless network with parameters $(N, p_i, \lambda_i, w_i)$ operating under the Single packet queues discipline. The performance of the Max-Weight policy with $\tilde{\alpha}_i = w_i/p_i\mu_i^S, \forall i$, is such that*

$$\mathbb{E}\left[J^{MW^S}\right] \leq \mathbb{E}\left[J^{R^S}\right], \tag{4.39}$$

*where $\mu_i^S$ and $\mathbb{E}[J^{R^S}]$ are the optimal scheduling probability for the case of Single packet queues and the associated EWSAoI attained by $R^S$, respectively.*

**Theorem 4.13** (Performance Bounds for $MW^N$)**.** *Consider a wireless network with parameters $(N, p_i, \lambda_i, w_i)$ operating under the No queues discipline. The performance of the Max-Weight Policy with $\tilde{\alpha}_i = w_i/p_i\mu_i^N, \forall i$, is such that*

$$\mathbb{E}\left[J^{MW^N}\right] \leq \mathbb{E}\left[J^{R^N}\right], \tag{4.40}$$

*where $\mu_i^N$ and $\mathbb{E}[J^{R^N}]$ are the optimal scheduling probability for the case of No queues and the associated EWSAoI attained by $R^N$, respectively.*

The proofs of Theorems 4.12 and 4.13 are provided in Appendices 4.D and 4.E, respectively. Both proofs rely on the construction of equivalent systems that facilitate the analysis of the expression of the drift in (4.38). The performance of $MW^F$ is evaluated in the next section using simulations.

Stationary Randomized policies select streams randomly, according to a fixed set of scheduling probabilities $\{\mu_i\}_{i=1}^N$. In contrast, Max-Weight policies leverage the knowledge of $h_i(t)$ and $z_i(t)$ to select which stream to serve. Therefore, it is not surprising that Max-Weight policies outperform Randomized policies. However, establishing a performance guarantee as in (4.39) and (4.40) is challenging for it depends on finding a tight upper bound for the performance of Max-Weight policies, which often do not have properties

such as *renewal intervals* that simplify the analysis. Next, we provide numerical results that further validate the superior performance of the Max-Weight policies.

## 4.5 Simulation Results

In this section, we evaluate the performance of scheduling policies in terms of the EWSAoI. We compare: 1) the Optimal Stationary Randomized Policy for the case of *Single packet queues* $R^S$, *No queues* $R^N$ and *FCFS queues* $R^F$; 2) the Max-Weight Policy[5] for the case of *Single packet queues* $MW^S$, *No queues* $MW^N$ and *FCFS queues* $MW^F$; and 3) the Whittle's Index Policy under the *No queues* discipline. The first two policies were developed in Secs. 4.3 and 4.4, respectively, and the last policy was proposed in [37]. The Lower Bound $L_B$ derived in Sec. 4.2 is displayed for comparison.

In Figs. 4-4 and 4-5 we simulate networks with increasing arrival rates, in Figs. 4-6 and 4-7 we simulate networks with increasing channel reliability, and in Figs. 4-8 and 4-9 we simulate networks with increasing number of streams. The performance of the Randomized policies is computed using the closed-form expressions derived in Sec. 4.3 while the performance of the Max-Weight and Whittle's Index policies are averages over 10 simulation runs. The results in Figs. 4-4, 4-5, 4-6 and 4-7 are for networks with $N = 4$ traffic streams and time-horizon of $T = 2 \times 10^6$ slots. The results in Figs. 4-8 and 4-9 are for networks with increasing value of $N$ and time-horizon of $T = N \times 5 \times 10^5$ slots.

In Figs. 4-4 and 4-5, the four streams have weights $w_1 = w_2 = 4$ and $w_3 = w_4 = 1$, channel reliabilities $p_i = i/N, \forall i$, and arrival rates $\lambda_i = (N - i + 1)/N \times \lambda, \forall i$, for an increasing value of $\lambda \in \{0.01, 0.02, \cdots, 0.35\}$. The simulation results are separated into Figs. 4-4 and 4-5 for clarity. The results in Figs. 4-4 and 4-5 suggest that the Max-Weight policy outperforms the corresponding Randomized and Whittle's Index policies with the same queueing discipline for every value of $\lambda$. The results also show that under the same class of scheduling policies, *Single packet queues* outperforms other queueing disciplines for every value of $\lambda$, as expected. It is evident from Fig. 4-4 that network instability, which occurs when

[5]For the Max-Weight Policies $MW^S$, $MW^N$ and $MW^F$, we employ $\beta_i = w_i/p_i\mu_i^X, \forall i$, where $\mu_i^X$ is the optimal scheduling probability for the associated queueing discipline.

$\lambda > 12/77$, is a major disadvantage of employing *FCFS queues*.

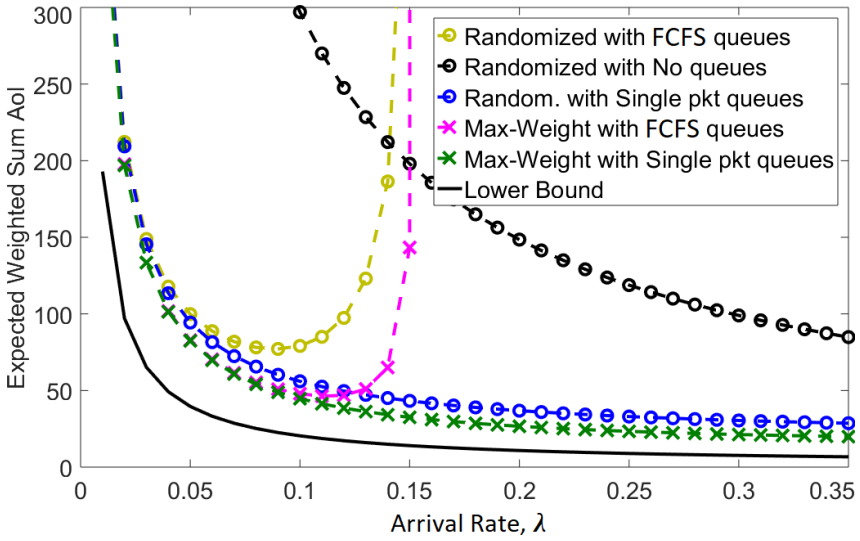



Figure 4-4: Networks with $N = 4$ streams, weights $w_1 = w_2 = 4$ and $w_3 = w_4 = 1$, time-horizon $T = 2 \times 10^6$ slots, channel reliabilities $p_i = i/N$, and $\lambda_i = (N - i + 1)/N \times \lambda, \forall i$, for an increasing arrival rate $\lambda$.

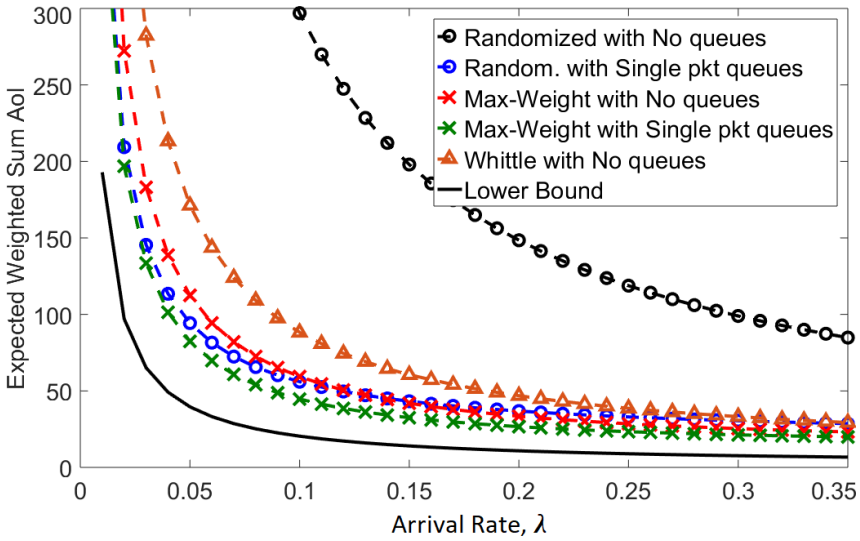



Figure 4-5: Networks with $N = 4$ streams, weights $w_1 = w_2 = 4$ and $w_3 = w_4 = 1$, time-horizon $T = 2 \times 10^6$ slots, channel reliabilities $p_i = i/N$, and $\lambda_i = (N - i + 1)/N \times \lambda, \forall i$, for an increasing arrival rate $\lambda$.

In Figs. 4-6 and 4-7, the four streams have weights $w_1 = w_2 = w_3 = 1$ and $w_4 = 4$, arrival rates $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/10$, and channel reliabilities $p_1 = 4/5$, $p_2 = 3/5$,

$p_3 = 2/5$ with increasing $p_4 \in \{0.05, 0.10, \cdots, 1.00\}$. This variation in $p_4$ can represent a scenario in which the destination is moving. The results in Figs. 4-6 and 4-7 suggest that the performance of *FCFS queues* are the most sensitive to network changes, while *Single packet queues* are the least sensitive. Intuitively, this effect is explained by the accumulation of stale packets in the *FCFS queues* when the channel reliability $p_4$ decreases.

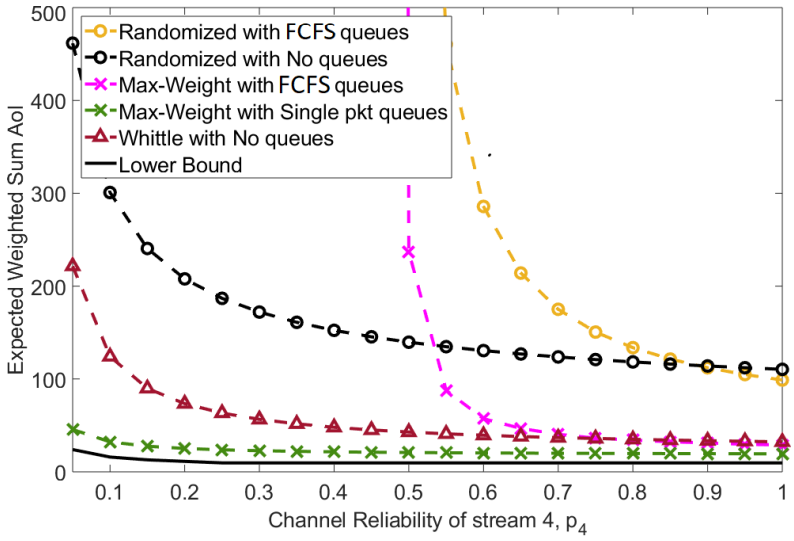



Figure 4-6: Networks with $N = 4$ streams, weights $w_1 = w_2 = w_3 = 1$ and $w_4 = 4$, time-horizon $T = 2 \times 10^6$ slots, arrival rates $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/10$, channel reliabilities $p_1 = 4/5$, $p_2 = 3/5$, $p_3 = 2/5$, and increasing $p_4 \in \{0.05, 0.10, \cdots, 1.00\}$.

In Figs. 4-8 and 4-9 we simulate networks with an increasing number of streams $N \in \{5, 8, 10, 13, 15, 18, \cdots, 25, 28, 30\}$. Streams have identical priorities $w_i = 1, \forall i \in \{1, \cdots, N\}$, arrival rates $\lambda_i = 0.05, \forall i$, and channel reliabilities $p_i = 0.8, \forall i$. The results in Figs. 4-8 and 4-9 show that the AoI performance of scheduling policies under *FCFS queues* degrades sharply as the number of source-destination pairs $N$ increase. In contrast, the performance of the Max-Weight policy for *Single packet queues* degrades gracefully as the network grows. *This comparison is important, especially when we consider that* FCFS queues *are the standard queueing discipline in most communication systems while* LCFS queues *(which are equivalent to* Single packet queues *from the perspective of AoI) are commonly not implemented.*

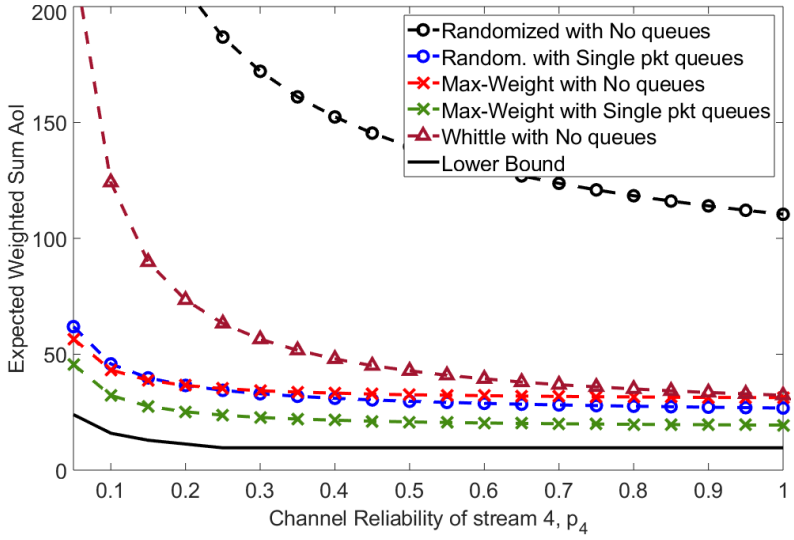



Figure 4-7: Networks with $N = 4$ streams, weights $w_1 = w_2 = w_3 = 1$ and $w_4 = 4$, time-horizon $T = 2 \times 10^6$ slots, arrival rates $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/10$, channel reliabilities $p_1 = 4/5$, $p_2 = 3/5$, $p_3 = 2/5$, and increasing $p_4 \in \{0.05, 0.10, \cdots, 1.00\}$.

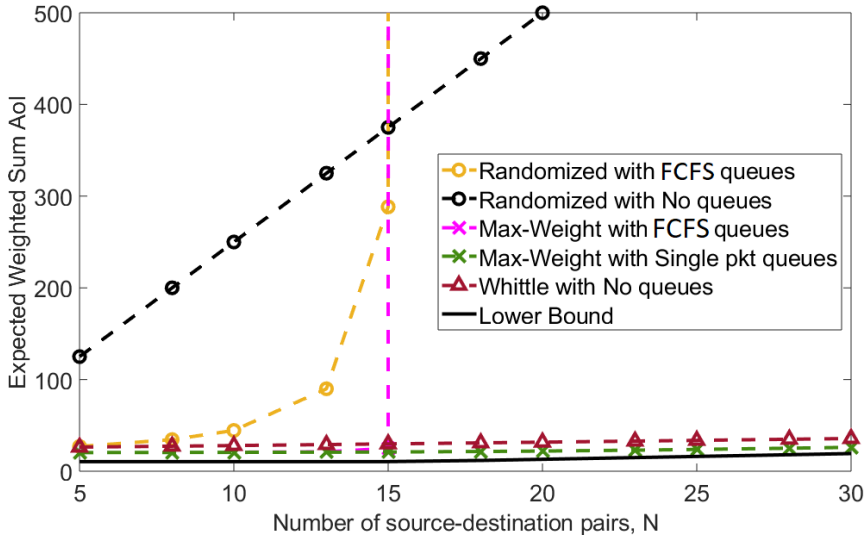



Figure 4-8: Networks with an increasing number of streams $N \in \{5, 8, 10, 13, 15, 18, \cdots, 25, 28, 30\}$. Streams have identical priorities $w_i = 1, \forall i \in \{1, 2, \cdots, N\}$, arrival rates $\lambda_i = 0.05, \forall i$, channel reliabilities $p_i = 0.8, \forall i$, and time-horizon of $T = N \times 5 \times 10^5$ slots.

## 4.6 Summary

In this chapter, we considered a broadcast single-hop wireless network with sources that generate packets according to a stochastic process, enqueue them in separate (per source)

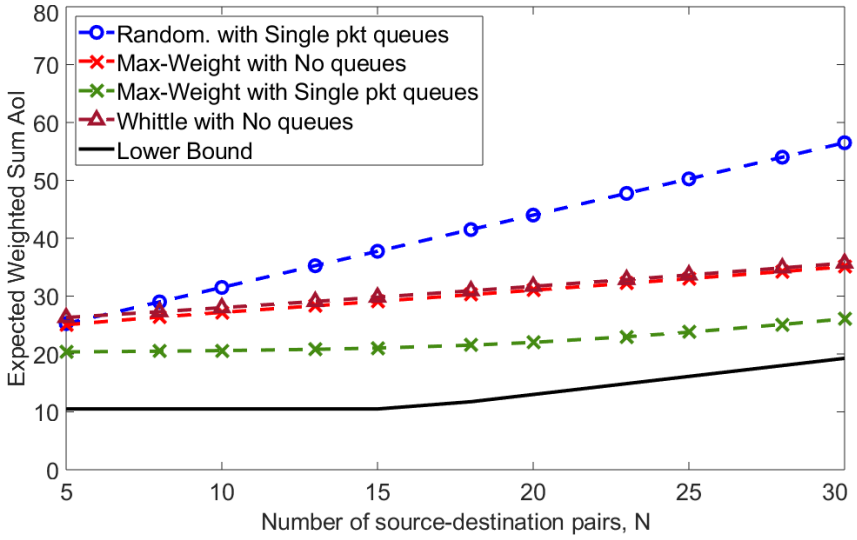



Figure 4-9: Networks with an increasing number of streams $N \in \{5,8,10,13,15,18,\cdots,25,28,30\}$. Streams have identical priorities $w_i = 1, \forall i \in \{1,2,\cdots,N\}$, arrival rates $\lambda_i = 0.05, \forall i$, channel reliabilities $p_i = 0.8, \forall i$, and time-horizon of $T = N \times 5 \times 10^5$ slots.

queues, and transmit them via unreliable communication links. We addressed the problem of minimizing the Expected Weighted Sum AoI in the network.

First, we derived a lower bound on the AoI performance achievable by any given network, operating under any queueing discipline. Then, we considered three common queueing disciplines and developed both a Stationary Randomized policy and a Max-Weight policy under each discipline. A summary of the main results follows:

- Stationary Randomized policy for Single packet queues with optimal scheduling probability $\mu_i^S \propto \sqrt{w_i/p_i}$ is 4-optimal for any network configuration $(N, p_i, \lambda_i, w_i)$. Notice that, contrary to intuition, the optimal scheduling probability $\mu_i^S$ is independent of the packet arrival rate $\lambda_i$.

- Stationary Randomized policies for No queues and FCFS queues have optimal scheduling probabilities $\mu_i^N$ and $\mu_i^F$, respectively, that are sensitive to the packet arrival rate $\lambda_i$, as shown in Theorems 4.8 and 4.10.

- Max-Weight policies for Single packet queues and No queues are shown in Theorems 4.12 and 4.13 to outperform the corresponding Stationary Randomized Policies with the same queueing discipline.

We evaluated the AoI performance both analytically and using simulations. Our approach allowed us to evaluate the combined impact of the stochastic arrivals, queueing discipline and scheduling policy on AoI. Numerical results show that the Max-Weight policy with LCFS queues achieves near optimal performance in various network settings.

# Appendices

## 4.A Proof of Proposition 4.2

**Proposition 4.2.** The infinite-horizon AoI objective function $J^{\pi}$ can be expressed as follows

$$J^{\pi} = \lim_{T\to\infty} \sum_{i=1}^{N} \frac{w_i}{2N} \left[ \frac{\bar{\mathbb{M}}[I_i^2]}{\bar{\mathbb{M}}[I_i]} + \frac{2\bar{\mathbb{M}}[z_i I_i]}{\bar{\mathbb{M}}[I_i]} + 1 \right] \text{ w.p.1 ,}$$

where $I_i[m]$ is the inter-delivery time, $z_i[m]$ is the packet delay and

$$\bar{\mathbb{M}}[z_i I_i] = \frac{1}{D_i(T)} \sum_{m=1}^{D_i(T)} z_i[m-1] I_i[m] \, .$$

*Proof.* Consider a network employing policy $\pi \in \Pi$ during the finite time-horizon $T$. Let $\Omega$ be the sample space associated with this network and let $\omega \in \Omega$ be a sample path. For a given sample path $\omega$, let $D_i(T)$ be the total number of packets delivered to destination $i$, $z_i[m]$ be the packet delay associated with the $m$th packet delivery, $I_i[m]$ be the number of slots between the $(m-1)$th and $m$th packet deliveries and $R_i$ be the number of slots remaining after the last packet delivery. Then, the time-horizon can be written as follows

$$T = \sum_{m=1}^{D_i(T)} I_i[m] + R_i, \forall i \in \{1, 2, \cdots, N\} \, . \tag{4.41}$$

The evolution of $h_i(t)$ is well-defined in each of the time intervals $I_i[m]$ and $R_i$. According to (4.4), during the interval $I_i[m]$, the parameter $h_i(t)$ evolves as $\{z_i[m-1]+1, z_i[m-1]+2, \cdots, z_i[m-1]+I_i[m]\}$. This pattern is repeated throughout the entire time-horizon, for $m \in \{1, 2, \cdots, D_i(T)\}$, and also during the last $R_i$ slots. As a result, the time-average

AoI associated with destination $i$ can be expressed as

$$\frac{1}{T}\sum_{t=1}^{T} h_i(t) = \frac{1}{T}\left[\sum_{m=1}^{D_i(T)} z_i[m-1]I_i[m] + \sum_{m=1}^{D_i(T)} \frac{(I_i[m]+1)I_i[m]}{2} + z_i[D_i(T)]R_i + \frac{(R_i+1)R_i}{2}\right]$$
$$= \frac{1}{2}\left[\frac{D_i(T)}{T}\frac{1}{D_i(T)}\sum_{m=1}^{D_i(T)}\left(I_i^2[m] + 2z_i[m-1]I_i[m]\right) + \frac{R_i^2}{T} + 2\frac{z_i[D_i(T)]R_i}{T} + 1\right], \forall i\,, \tag{4.42}$$

where the second equality uses (4.41) to replace the two linear terms by $T$.

Combining (4.41) with the sample mean $\bar{\mathbb{M}}[I_i]$, yields

$$\frac{T}{D_i(T)} = \frac{\sum_{j=1}^{D_i(T)} I_i[j] + R_i}{D_i(T)} = \bar{\mathbb{M}}[I_i] + \frac{R_i}{D_i(T)}\,. \tag{4.43}$$

Substituting (4.43) into (4.42) and then employing the sample mean operator $\bar{\mathbb{M}}$ on $I_i^2[m]$ and $z_i[m-1]I_i[m]$, gives

$$\frac{1}{T}\sum_{t=1}^{T} h_i(t) = \frac{1}{2}\left[\left(\bar{\mathbb{M}}[I_i] + \frac{R_i}{D_i(T)}\right)^{-1}\left(\bar{\mathbb{M}}[I_i^2] + 2\bar{\mathbb{M}}[z_i I_i]\right) + \frac{R_i^2}{T} + 2\frac{z_i[D_i(T)]R_i}{T} + 1\right], \forall i\,, \tag{4.44}$$

The next step is to take the limit of (4.44) as $T \to \infty$. Prior to taking the limit, we assume in the remaining part of this proof that the system time of the HoL packet in queue $i$ is finite, $z_i(t) < \infty$, as $t \to \infty$, with probability one. Recall from the discussion in Sec. 4.1.2 that if $z_i(t) \to \infty$ with a positive probability, then the objective function diverges, $\mathbb{E}[J^\pi] \to \infty$. Hence, there is no loss of optimality in assuming that $z_i(t) < \infty$ with probability one. From this assumption, it follows that packet delays are finite with probability one, $z_i[m] < \infty$, and that packets are continuously delivered to destination $i$, what makes the number of slots after the last packet delivery $R_i$, finite with probability one. Hence, in the limit as $T \to \infty$, we have continuous packet deliveries, $D_i(T) \to \infty$, and finite $z_i[m]$ and $R_i$ implying that $R_i^2/T \to 0$, $R_i/D_i(T) \to 0$ and $z_i[D_i(T)]R_i/T \to 0$. Employing those limits into (4.44) gives

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} h_i(t) = \lim_{T\to\infty}\left[\frac{\bar{\mathbb{M}}[I_i^2]}{2\bar{\mathbb{M}}[I_i]} + \frac{\bar{\mathbb{M}}[z_i I_i]}{\bar{\mathbb{M}}[I_i]} + \frac{1}{2}\right], \forall i\,. \tag{4.45}$$

To obtain the final expression in (4.12) we employ (4.45) into (4.5), without the expectation.

■

## 4.B Proof of Theorem 4.3

**Theorem 4.3** (Lower Bound). For any given wireless network with parameters $(N, p_i, \lambda_i, w_i)$ and an arbitrary queueing discipline, the optimization problem in (4.16a)-(4.16c) provides a lower bound on the AoI minimization problem, namely $L_B \leq \mathbb{E}[J^*]$. The unique solution to (4.16a)-(4.16c) is given by

$$\hat{q}_i^{L_B} = \min\left\{\lambda_i, \sqrt{\frac{w_i p_i}{2N\gamma^*}}\right\}, \forall i\,,$$

where $\gamma^*$ yields from Algorithm 3. The lower bound is given by

$$L_B = \frac{1}{2N}\sum_{i=1}^{N} w_i\left(\frac{1}{\hat{q}_i^{L_B}} + 1\right)\,.$$

**Algorithm 3** Solution to the Lower Bound

1: $\tilde{\gamma} \leftarrow (\sum_{i=1}^{N}\sqrt{w_i/p_i})^2/(2N)$ and $\gamma_i \leftarrow w_i p_i/2N\lambda_i^2, \forall i$
2: $\gamma \leftarrow \max\{\tilde{\gamma}; \gamma_i\}$
3: $q_i \leftarrow \lambda_i \min\{1; \sqrt{\gamma_i/\gamma}\}, \forall i$
4: $S \leftarrow \sum_{i=1}^{N} q_i/p_i$
5: **while** $S < 1$ **and** $\gamma > 0$ **do**
6: decrease $\gamma$ slightly
7: repeat steps 3 and 4 to update $q_i$ and $S$
8: **end while**
9: **return** $\gamma^* = \gamma$ and $\hat{q}_i^{L_B} = q_i, \forall i$

*Proof.* Consider a network with parameters $(N, p_i, \lambda_i, w_i)$ and an arbitrary queueing discipline. First, we show that (4.16a)-(4.16c) provides a lower bound $L_B$ on the AoI minimization problem $\mathbb{E}[J^*] = \min_{\pi\in\Pi}\mathbb{E}[J^\pi]$, then we find the unique solution to (4.16a)-(4.16c) by analyzing its KKT Conditions. The optimization problem in (4.16a)-(4.16c) is rewritten below for convenience.

**Lower Bound**

$$L_B = \min_{\pi\in\Pi} \left\{ \frac{1}{2N} \sum_{i=1}^{N} w_i \left( \frac{1}{\hat{q}_i^{\pi}} + 1 \right) \right\}$$
$$\text{s.t. } \textstyle\sum_{i=1}^{N} \hat{q}_i^{\pi}/p_i \leq 1 \, ;$$
$$\hat{q}_i^{\pi} \leq \lambda_i, \forall i \, ,$$

Consider the expression for the time-average AoI associated with destination $i$ in (4.42), which is valid for any admissible policy $\pi \in \Pi$ and time-horizon $T$. Substituting the non-negative terms $z_i[m-1]I_i[m]$ and $z_i[D_i(T)]R_i$ by zero, employing the sample mean operator $\bar{\mathbb{M}}$ to $I_i^2[m]$ and then applying Jensen's inequality $\bar{\mathbb{M}}[I_i^2] \geq (\bar{\mathbb{M}}[I_i])^2$, we obtain

$$\frac{1}{T}\sum_{t=1}^{T} h_i(t) \geq \frac{1}{2}\left( \frac{D_i(T)}{T} \left(\bar{\mathbb{M}}[I_i]\right)^2 + \frac{R_i^2}{T} + 1 \right) . \tag{4.47}$$

Substituting (4.43) into (4.47), gives

$$\frac{1}{T}\sum_{t=1}^{T} h_i(t) \geq \frac{1}{2}\left( \frac{1}{T}\frac{(T-R_i)^2}{D_i(T)} + \frac{R_i^2}{T} + 1 \right) . \tag{4.48}$$

By minimizing the LHS of (4.48) analytically with respect to the variable $R_i$, we have

$$\frac{1}{T}\sum_{t=1}^{T} h_i(t) \geq \frac{1}{2}\left( \frac{T}{D_i(T)+1} + 1 \right) . \tag{4.49}$$

Taking the expectation of (4.49) and applying Jensen's inequality, yields

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[h_i(t)\right] \geq \frac{1}{2}\left( \frac{1}{\mathbb{E}\left[\frac{D_i(T)}{T}\right] + \frac{1}{T}} + 1 \right) . \tag{4.50}$$

Applying the limit $T \to \infty$ to (4.50) and using the definition of throughput in (4.7), gives

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[h_i(t)\right] \geq \frac{1}{2}\left( \frac{1}{\hat{q}_i^{\pi}} + 1 \right) . \tag{4.51}$$

Substituting (4.51) into the objective function in (4.5), yields

$$\mathbb{E}\left[J^{\pi}\right] = \lim_{T\to\infty} \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{T}\sum_{t=1}^{T} \mathbb{E}\left[h_i(t)\right] \geq \frac{1}{2N}\sum_{i=1}^{N} w_i\left(\frac{1}{\hat{q}_i^{\pi}}+1\right). \tag{4.52}$$

Inequality (4.52) is valid for any admissible policy $\pi \in \Pi$. Notice that the RHS of (4.52) depends only on the network's long-term throughput $\{\hat{q}_i^{\pi}\}_{i=1}^{N}$. Adding to (4.52) the two necessary conditions for the long-term throughput in (4.8) and (4.9), and then minimizing the resulting problem over all policies in $\Pi$, yields $\mathbb{E}[J^*] = \min_{\pi\in\Pi}\mathbb{E}\left[J^{\pi}\right] \geq L_B$ where $L_B$ is given by (4.16a)-(4.16c).

After showing that (4.16a)-(4.16c) provides a lower bound on the AoI minimization problem, we find the unique set of network's long-term throughput $\{\hat{q}_i^{L_B}\}_{i=1}^{N}$ that solves (4.16a)-(4.16c) by analyzing its KKT Conditions. Let $\gamma$ be the KKT multiplier associated with the relaxation of $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i \leq 1$ and $\{\zeta_i\}_{i=1}^{N}$ be the KKT multipliers associated with the relaxation of $\hat{q}_i^{\pi} \leq \lambda_i, \forall i$. Then, for $\gamma \geq 0$ , $\zeta_i \geq 0$ and $\hat{q}_i^{\pi} \in (0,1], \forall i$, we define

$$\mathscr{L}(\hat{q}_i^{\pi},\zeta_i,\gamma) = \frac{1}{2N}\sum_{i=1}^{N} w_i\left(\frac{1}{\hat{q}_i^{\pi}}+1\right) + \sum_{i=1}^{N}\zeta_i\left(\hat{q}_i^{\pi}-\lambda_i\right) + \gamma\left(\sum_{i=1}^{N}\frac{\hat{q}_i^{\pi}}{p_i}-1\right), \tag{4.53}$$

and, otherwise, we define $\mathscr{L}(\hat{q}_i^{\pi},\zeta_i,\gamma) = +\infty$. Then, the KKT Conditions are

(i) Stationarity: $\nabla_{\hat{q}_i^{\pi}}\mathscr{L}(\hat{q}_i^{\pi},\zeta_i,\gamma) = 0$;

(ii) Complementary Slackness: $\gamma(\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i - 1) = 0$;

(iii) Complementary Slackness: $\zeta_i(\hat{q}_i^{\pi}-\lambda_i) = 0, \forall i$;

(iv) Primal Feasibility: $\hat{q}_i^{\pi} \leq \lambda_i, \forall i$, and $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i \leq 1$;

(v) Dual Feasibility: $\zeta_i \geq 0, \forall i$, and $\gamma \geq 0$.

Since $\mathscr{L}(\hat{q}_i^{\pi},\zeta_i,\gamma)$ is a *convex function*, if there exists a vector $(\{\hat{q}_i^{L_B}\}_{i=1}^{N}, \{\zeta_i^*\}_{i=1}^{N}, \gamma^*)$ that satisfies all KKT Conditions, then this vector is unique. Next, we find the vector $(\{\hat{q}_i^{L_B}\}_{i=1}^{N}, \{\zeta_i^*\}_{i=1}^{N}, \gamma^*)$.

To assess stationarity, $\nabla_{\hat{q}_i^\pi}\mathscr{L}(\hat{q}_i^\pi,\zeta_i,\gamma)=0$, we calculate the partial derivative of $\mathscr{L}(\hat{q}_i^\pi,\zeta_i,\gamma)$ with respect to $\hat{q}_i^\pi$, which gives

$$-\frac{w_i p_i}{2N(\hat{q}_i^\pi)^2}+\zeta_i p_i+\gamma=0\,,\forall i\,. \tag{4.54}$$

From complementary slackness, $\gamma(\sum_{i=1}^N \hat{q}_i^\pi/p_i-1)=0$, we know that either $\gamma=0$ or $\sum_{i=1}^N \hat{q}_i^\pi/p_i=1$. First, we consider the case $\sum_{i=1}^N \hat{q}_i^\pi/p_i=1$. Based on dual feasibility, $\zeta_i\geq 0$, we can separate streams $i\in\{1,\cdots,N\}$ into two categories: streams with $\zeta_i>0$ and streams with $\zeta_i=0$.

*Category 1)* streams $i$ with $\zeta_i>0$. It follows from complementary slackness, $\zeta_i(\hat{q}_i^\pi-\lambda_i)=0$, that $\hat{q}_i^\pi=\lambda_i$. Plugging this value of $\hat{q}_i^\pi$ into (4.54) gives the inequality $\zeta_i p_i=\gamma_i-\gamma>0$, where we define the constant

$$\gamma_i:=\frac{w_i p_i}{2N\lambda_i^2}\,. \tag{4.55}$$

*Category 2)* streams $i$ with $\zeta_i=0$. It follows from (4.54) that

$$\gamma=\gamma_i\left(\frac{\lambda_i}{\hat{q}_i^\pi}\right)^2\Rightarrow\ \hat{q}_i^\pi=\lambda_i\sqrt{\frac{\gamma_i}{\gamma}}\,,\text{for }\gamma_i-\gamma\leq 0\,. \tag{4.56}$$

Hence, for any fixed value of $\gamma\geq 0$, if $\gamma\geq\gamma_i$ then stream $i$ is in Category 2, otherwise, stream $i$ is in Category 1. Moreover, the values of $\zeta_i$ and $\hat{q}_i^\pi$ associated with stream $i$, in either Category, can be expressed as

$$\zeta_i=\max\left\{0;\frac{\gamma_i-\gamma}{p_i}\right\},\forall i\,. \tag{4.57}$$

$$\hat{q}_i^\pi=\lambda_i\min\left\{1;\sqrt{\frac{\gamma_i}{\gamma}}\right\},\forall i\,. \tag{4.58}$$

Notice that when $\gamma>\max\{\gamma_i\}$, then all streams are in Category 2 and $\hat{q}_i^\pi<\lambda_i,\forall i$. By decreasing the value of $\gamma$ gradually, the throughput $\hat{q}_i^\pi$ of each stream $i$ in (4.58) either increases or remain fixed at $\lambda_i$. Our goal is to find the value of $\gamma^*$ which yields $\{\hat{q}_i^\pi\}_{i=1}^N$ satisfying the condition $\sum_{i=1}^N \hat{q}_i^\pi/p_i=1$. Suppose this condition is satisfied when $\gamma>\max\{\gamma_i\}$,

with all streams in Category 2, then it follows that

$$\sum_{i=1}^{N}\frac{\hat{q}_i^{\pi}}{p_i}=\sum_{i=1}^{N}\frac{\lambda_i}{p_i}\sqrt{\frac{\gamma_i}{\gamma}}=\frac{1}{\sqrt{2N\gamma}}\sum_{i=1}^{N}\sqrt{\frac{w_i}{p_i}}=1\;\Rightarrow\;\gamma^*=\tilde{\gamma}:=\frac{1}{2N}\left(\sum_{i=1}^{N}\sqrt{\frac{w_i}{p_i}}\right)^2, \tag{4.59}$$

where $\tilde{\gamma}$ is a fixed constant and the solution is unique $\gamma^*=\tilde{\gamma}$.

Alternatively, suppose that $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i=1$ is satisfied when $\min\{\gamma_i\}\leq\gamma\leq\max\{\gamma_i\}$, with some streams in Category 1 and others in Category 2. To find $\gamma^*$, we start with $\gamma=\max\{\gamma_i\}$ and gradually decrease $\gamma$, adjusting $\{\hat{q}_i^{\pi}\}_{i=1}^{N}$ according to (4.58) until we reach $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i=1$. The uniqueness of $\gamma^*$ follows from the monotonicity of $\hat{q}_i^{\pi}$ with respect to $\gamma$ in (4.58).

Another possibility is for $\gamma$ to reach a value lower than $\min\{\gamma_i\}$ and still result in $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i<1$. Notice from (4.58) that when $\gamma<\min\{\gamma_i\}$, then all streams are in Category 1 and have maximum throughputs, namely $\hat{q}_i^{\pi}=\lambda_i,\forall i$. It follows that $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i=\sum_{i=1}^{N}\lambda_i/p_i<1$, in which case the condition $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i=1$ cannot be satisfied for any value of $\gamma\geq 0$. Hence, from complementary slackness, $\gamma(\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i-1)=0$, we have the unique solution $\gamma^*=0$.

*Proposed algorithm to find* $\gamma^*$ that solves the KKT Conditions: start with $\gamma=\max\{\gamma_i;\tilde{\gamma}\}$. Then, compute $\{\hat{q}_i^{\pi}\}_{i=1}^{N}$ using (4.58) and verify if the condition $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i=1$ is satisfied. If $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i<1$, then gradually decrease $\gamma$ and repeat the procedure. Stop when $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i=1$ or when $\gamma<\min\{\gamma_i\}$. If $\sum_{i=1}^{N}\hat{q}_i^{\pi}/p_i=1$ holds, then assign $\gamma^*\leftarrow\gamma$. Otherwise, if $\gamma<\min\{\gamma_i\}$ holds, then assign $\gamma^*\leftarrow 0$. The solution to the KKT Conditions is given by $\gamma^*$ and the associated $\zeta_i^*$ and $\hat{q}_i^{L_B}$ obtained by substituting $\gamma^*$ into (4.57) and (4.58), respectively.

It is evident from the proposed algorithm that for any given network with parameters $(N,p_i,\lambda_i,w_i)$ and an arbitrary queueing discipline, the solution to the KKT Conditions, $(\{\hat{q}_i^{L_B}\}_{i=1}^{N},\{\zeta_i^*\}_{i=1}^{N},\gamma^*)$, exists and is unique. The proposed algorithm is described using pseudocode in Algorithm 3.

■

## 4.C Proof of Proposition 4.5

**Proposition 4.5**. The optimal EWSAoI achieved by a network with Single packet queues over the class $\Pi_R$ is given by (4.20a)-(4.20b), where $R^S$ denotes the Optimal Stationary Randomized Policy for the Single packet queue discipline.

**Optimal Randomized policy for Single packet queues**

$$\mathbb{E}\left[J^{R^S}\right] = \min_{R\in\Pi_R}\left\{\frac{1}{N}\sum_{i=1}^{N} w_i\left(\frac{1}{\lambda_i} - 1 + \frac{1}{p_i\mu_i}\right)\right\}$$
$$\text{s.t. } \textstyle\sum_{i=1}^{N}\mu_i \leq 1\ ;$$

*Proof.* Consider the evolution of $h_i(t)$ and $z_i^S(t)$ given in (4.4) and (4.1), respectively. Under policy $R \in \Pi_R$, the tuple $(h_i(t), z_i^S(t))$ represents the state of stream $i$ and evolves according to a two-dimensional Markov Chain with countably-infinite state space. The basic structure of this Markov Chain is illustrated in Fig. 4-10.

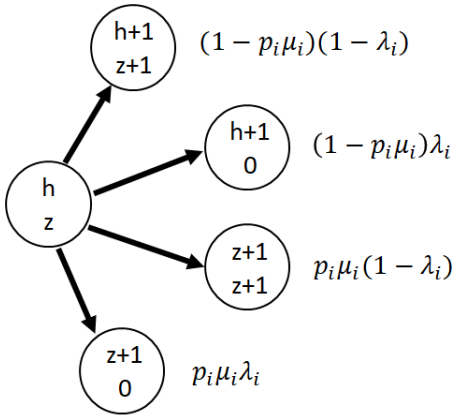



Figure 4-10: Illustration of the state evolution associated with stream $i$ of a network employing policy $R \in \Pi_R$ and operating under the Single packet queue discipline. In partciular, we show the outgoing transition arcs from any given state $(h_i(t), z_i(t)) = (h,z), \forall z \in \{0,1,2,\cdots\}, h \geq z$ with the associated transition probabilities.

From the basic structure in Fig. 4-10, we derive the stationary distribution of stream $i$'s Markov Chain. To that end, we separate state transitions into three categories and obtain the associated probability distributions.

- Transition to an empty[6] queue $(h,h), \forall h \in \{1,2,\cdots\}$:

$$\mathbb{P}(h,h) = \mathbb{P}(1,0)\frac{(1-\lambda_i)^h}{\lambda_i}\left\{\frac{1-(1-p_i\mu_i)^h}{p_i\mu_i}\right\}; \tag{4.60}$$

- Transition to a slot with a new arrival $(h,0), \forall h \in \{1,2,\cdots\}$:

$$\mathbb{P}(h,0) = \mathbb{P}(1,0)\left\{\sum_{n=0}^{h-1}(1-\lambda_i)^{h-1-n}(1-p_i\mu_i)^n\right\}; \tag{4.61}$$

- Uneventful transition to $(h,z), \forall z \in \{1,2,\cdots\}, h > z$:

$$\begin{aligned}\mathbb{P}(h,z) &= \mathbb{P}(h-z,0)(1-\lambda_i)^z(1-p_i\mu_i)^z \\ &= \mathbb{P}(1,0)(1-\lambda_i)^z(1-p_i\mu_i)^z\left\{\sum_{n=0}^{h-z-1}(1-\lambda_i)^{h-z-1-n}(1-p_i\mu_i)^n\right\}. \end{aligned} \tag{4.62}$$

Then, with the stationary distribution, we obtain an expression for the probability of $h_i(t) = h$

$$\mathbb{P}(h) = \sum_{z=0}^{h}\mathbb{P}(h,z) = \frac{\mathbb{P}(1,0)}{\lambda_i}\left[\sum_{n=0}^{h-1}(1-\lambda_i)^{h-1-n}(1-p_i\mu_i)^n\right], \tag{4.63}$$

for $h \geq 1$ and, since $\sum_h \mathbb{P}(h) = 1$, we have that $\mathbb{P}(1,0) = \lambda_i^2 p_i\mu_i$.

Since the countable-state Markov Chain is irreducible and has a stationary distribution, this distribution is unique. Moreover, the chain is positive recurrent and

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[h_i(t)] = \mathbb{E}[h] = \sum_{h=1}^{\infty}h\mathbb{P}(h) = \frac{1}{p_i\mu_i} + \frac{1}{\lambda_i} - 1\,. \tag{4.64}$$

Proposition 4.5 follows from substituting (4.64) into the objective function in (4.6). ■

[6]When the queue is empty, the system time $z$ is not part of the network state. However, to facilitate the analysis, and without loss of generality, we assume in this appendix that $z$ is always part of the state and evolves according to (4.1).

## 4.D Proof of Theorem 4.12

**Theorem 4.12** (Performance Bounds for $MW^S$). Consider a wireless network with parameters $(N, p_i, \lambda_i, w_i)$ operating under the Single packet queues discipline. The performance of the Max-Weight policy with $\tilde{\alpha}_i = w_i/p_i\mu_i^S, \forall i$, is such that

$$\mathbb{E}\left[J^{MW^S}\right] \leq \mathbb{E}\left[J^{R^S}\right] ,$$

where $\mu_i^S$ and $\mathbb{E}[J^{R^S}]$ are the optimal scheduling probability for the case of Single packet queues and the associated EWSAoI attained by $R^S$, respectively.

*Proof.* Consider stream $i$ from a network operating under the *Single packet queue* discipline. In each slot $t$, a packet is transmitted, i.e. $u_i(t) = 1$, if the stream is selected and its queue is non-empty. Hence, packet transmissions $u_i(t)$ depend on the queue backlog. To decouple packet transmissions from the queue backlog, we create *dummy packets* that can be transmitted without affecting the AoI. In particular, suppose that at time $t$ queue $i$ is selected and successfully transmits a packet with $z_i^S(t) = z$. Then, at the beginning of slot $t+1$, with probability $1-\lambda_i$ we place a *dummy packet* with $z_i^S(t+1) = z+1$ at the HoL of the queue, otherwise we place a real packet with $z_i^S(t) = 0$. From that moment on, the behavior of dummy packets is indistinguishable from real packets. Notice that due to the choice of $z_i^S(t+1) = z+1$, when a dummy packet is delivered to the destination, it does not change the associated AoI. Moreover, the system time $z_i^S(t)$ is now defined at every slot $t$ following (4.1). Next, we analyze the equivalent system with dummy packets.

The Max-Weight policy minimizes the drift in (4.38). Hence, any other policy $\pi \in \Pi$ yields a higher (or equal) value of $\Delta(\mathbb{S}(t))$. Consider the Stationary Randomized policy for Single packet queues defined in Sec. 4.3.1 with scheduling probability $\mu_i^S$ and let

$$\mathbb{E}\left[u_i(t)|\mathbb{S}(t)\right] = \mathbb{E}\left[u_i\right] = \mu_i^S . \tag{4.65}$$

Substituting $\mu_i^S$ into the Lyapunov Drift gives the upper bound

$$\Delta(\mathbb{S}(t)) \leq \frac{1}{N}\sum_{i=1}^{N}\tilde{\alpha}_i - \frac{1}{N}\sum_{i=1}^{N}\tilde{\alpha}_i p_i \left(h_i(t) - z_i^S(t)\right)\mu_i^S \,. \tag{4.66}$$

Now, taking the expectation with respect to $\mathbb{S}(t)$ and then the time-average on the interval $t \in \{1,2,\cdots,T\}$ yields

$$\frac{\mathbb{E}\left[L(T+1)\right]}{T} - \frac{\mathbb{E}\left[L(1)\right]}{T} \leq \frac{1}{N}\sum_{i=1}^{N}\tilde{\alpha}_i - \frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}\tilde{\alpha}_i p_i \mathbb{E}\left[h_i(t) - z_i^S(t)\right]\mu_i^S \,. \tag{4.67}$$

Manipulating this expression, assigning $\tilde{\alpha}_i = w_i/p_i\mu_i^S$ and taking the limit as $T \to \infty$, gives

$$\mathbb{E}\left[J^{MW^S}\right] \leq \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{p_i\mu_i^S} + \lim_{T\to\infty}\frac{1}{TN}\sum_{i=1}^{N}\sum_{t=1}^{T} w_i \mathbb{E}\left[z_i^S(t)\right] \,. \tag{4.68}$$

From the evolution of $z_i^S(t)$ in (4.1), we know that

$$\lim_{T\to\infty}\frac{1}{TN}\sum_{i=1}^{N}\sum_{t=1}^{T} w_i \mathbb{E}\left[z_i^S(t)\right] = \frac{1}{N}\sum_{i=1}^{N} w_i\left(\frac{1}{\lambda_i} - 1\right) \,. \tag{4.69}$$

Substituting (4.69) into (4.68) and then comparing the result with (4.20a) yields

$$\mathbb{E}\left[J^{MW^S}\right] \leq \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{p_i\mu_i^S} + \frac{1}{N}\sum_{i=1}^{N} w_i\left(\frac{1}{\lambda_i} - 1\right) = \mathbb{E}\left[J^{R^S}\right] \,. \tag{4.70}$$

■

## 4.E Proof of Theorem 4.13

**Theorem 4.13** (Performance Bounds for $MW^N$). Consider a wireless network with parameters $(N, p_i, \lambda_i, w_i)$ operating under the No queues discipline. The performance of the Max-Weight Policy with $\tilde{\alpha}_i = w_i/p_i\mu_i^N, \forall i$, is such that

$$\mathbb{E}\left[J^{MW^N}\right] \leq \mathbb{E}\left[J^{R^N}\right] ,$$

where $\mu_i^N$ and $\mathbb{E}[J^{R^N}]$ are the optimal scheduling probability for the case of No queues and the associated EWSAoI attained by $R^N$, respectively.

*Proof.* Consider stream $i$ from a network operating under the *No queue* discipline. In each slot $t$, a packet is successfully transmitted, i.e. $d_i(t) = 1$, if a packet arrives, the stream is selected and the channel is ON. Notice that all delivered packets have $z_i^N(t) = 0$. This is equivalent to a network with packets that are always fresh, i.e. $z_i^N(t) = 0, \forall i, t$, and with a virtual channel that is ON with probability $p_i\lambda_i$ and OFF with probability $1 - p_i\lambda_i$. The Lyapunov Drift for this equivalent system with fresh packets and virtual channels is given by:

$$\Delta(\mathbb{S}(t)) = \frac{1}{N}\sum_{i=1}^{N}\hat{\alpha}_i - \frac{1}{N}\sum_{i=1}^{N}\hat{\alpha}_i\lambda_i p_i h_i(t)\mathbb{E}\left[u_i(t)|\mathbb{S}(t)\right] . \tag{4.71}$$

For minimizing $\Delta(\mathbb{S}(t))$, the *Max-Weight policy selects, in each slot t, the stream i with a HoL packet and the highest value of* $\hat{\alpha}_i\lambda_i p_i h_i(t)$, with ties being broken arbitrarily. By comparing the drift of the equivalent system (4.71) and the original system (4.38), it is easy to see that $\tilde{\alpha}_i = \hat{\alpha}_i\lambda_i$.

The Max-Weight policy minimizes the drift in (4.71). Hence, any other policy $\pi \in \Pi$ yields a higher (or equal) value of $\Delta(\mathbb{S}(t))$. Consider the Stationary Randomized policy for No queues defined in Sec. 4.3.2 with scheduling probability $\mu_i^N$ and let

$$\mathbb{E}\left[u_i(t)|\mathbb{S}(t)\right] = \mathbb{E}\left[u_i\right] = \mu_i^N . \tag{4.72}$$

Substituting $\mu_i^N$ into the Lyapunov Drift gives the upper bound

$$\Delta(\mathbb{S}(t)) \leq \frac{1}{N}\sum_{i=1}^{N}\hat{\alpha}_i - \frac{1}{N}\sum_{i=1}^{N}\hat{\alpha}_i\lambda_i p_i h_i(t)\mu_i^N \,. \tag{4.73}$$

Now, taking the expectation with respect to $\mathbb{S}(t)$ and then the time-average on the interval $t \in \{1,2,\cdots,T\}$ yields

$$\frac{\mathbb{E}\left[L(T+1)\right]}{T} - \frac{\mathbb{E}\left[L(1)\right]}{T} \leq \frac{1}{N}\sum_{i=1}^{N}\hat{\alpha}_i - \frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}\hat{\alpha}_i\lambda_i p_i \mathbb{E}\left[h_i(t)\right]\mu_i^N \,. \tag{4.74}$$

Manipulating this expression, assigning $\hat{\alpha}_i = w_i/\lambda_i p_i \mu_i^N$ and taking the limit as $T \to \infty$, gives

$$\mathbb{E}\left[J^{MW^N}\right] \leq \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{\lambda_i p_i \mu_i^N} \,. \tag{4.75}$$

For deriving the upper bound in (4.40), consider the Optimal Stationary Randomized policy $R^N$. Substituting $\mu_i^N$ into (4.28a) and then comparing with (4.75) gives

$$\mathbb{E}\left[J^{MW^N}\right] \leq \mathbb{E}\left[J^{R^N}\right] \,. \tag{4.76}$$

■

Chapter 5

# WiFresh: AoI from Theory to Implementation

In this chapter, we study AoI in practical wireless networks. We show that as the congestion in the network increases, the AoI degrades sharply, leading to outdated information at the destination. To address this challenge, we propose WiFresh: an unconventional architecture that achieves near optimal information freshness for wireless networks of any size, even when the network is overloaded.

To illustrate the impact of information freshness on time-sensitive applications, consider a *monitoring system* composed of a remote monitor, a wireless base station (BS), and $N$ mobile nodes. Each node $i \in \{1,2,\cdots,N\}$ moves with an average velocity of $v_i$ meters per second, generates status information from time to time, and sends this information to the remote monitor via the wireless base station. Status information can include the node's current position, inertial measurements, and pictures of the environment. The remote monitor keeps track of the information, and is particularly interested in the position of the nodes. Assume that at time $t$, the latest packet received by the remote monitor from node $i$ had information about its position at time $\tau_i(t)$. The AoI $h_i(t) = t - \tau_i(t)$ captures how fresh the position information is at time $t$. In particular, an AoI of $h_i(t) = 2$ seconds represents that at time $t$ the remote monitor knows the location of node $i$ two seconds ago. Hence, the uncertainty about node *i's position at time t* is captured by the quantity $v_i h_i(t)$,

as illustrated in Fig. 5-1, and a *large AoI corresponds to a large position uncertainty*.

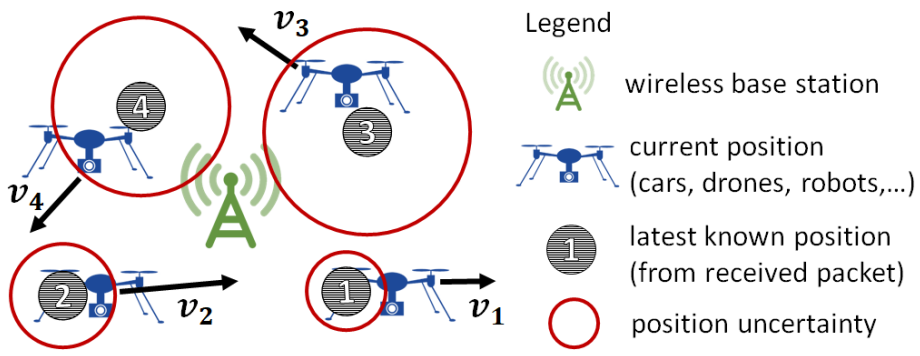



Figure 5-1: Illustration of a monitoring system. The wireless base station receives information from the $N = 4$ mobile nodes and forwards this information to the remote monitor. The position uncertainty of node $i$ from the perspective of the remote monitor is represented by the red circle with radius $v_i h_i(t)$ centered at the last known position of node $i$. The remote monitor does not know the current position of node $i$, which is illustrated by the blue drone.

In Table 5.1, we consider a sequence of monitoring systems with increasing size $N$ and display the average AoI $h_i(t)$ in seconds, where the average is taken over time $t$ and over all the $N$ nodes. Each *node* is a Raspberry Pi generating position information using the Stratus GPYes 2.0 u-blox 8 GPS receiver, and also generating inertial measurements using the Pololu MinIMU-9 v5 sensor. The *base station* is a Raspberry Pi receiving data from the $N$ nodes via: 1) WiFi UDP, which is the standard communication method; 2) WiFi Age Control Protocol (ACP), which uses the Transport layer protocol developed in [92] to *control the packet generation rates* at the source nodes in order to optimize information freshness; or 3) WiFresh, which is the network architecture proposed in this chapter. Details about the experiment and the complete set of measurements are provided in Sec. 5.4.3.

**WiFi UDP.** The measurements in the *first row* of Table 5.1 show that as the number of nodes $N$ in the WiFi UDP network increases, the *network becomes overloaded and the average AoI* $\mathbf{h_i(t)}$ *degrades sharply*. The average AoI for $N = 12$ nodes is 0.32 seconds, while for $N = 20$ nodes is 22.43 seconds, which means that the information at the remote monitor is (on average) 22 seconds old. This staleness directly affects the capability of the monitoring system of tracking the current position of the nodes.

**WiFi ACP.** The measurements in the *second row* of Table 5.1 show that the average

Table 5.1: Average AoI $h_i(t)$ (in seconds) of a monitoring system employing standard WiFi UDP (first row), WiFi with controlled packet generation rate (second row), or WiFresh (third row).

| $N$ | 2 | 8 | 12 | 20 | 24 |
|---|---|---|---|---|---|
| WiFi UDP | 0.34 | 0.32 | 0.32 | 22.43 | 34.09 |
| WiFi ACP | 0.99 | 0.97 | 0.88 | 5.76 | 6.91 |
| WiFresh | 0.29 | 0.35 | 0.39 | 0.49 | 0.54 |

AoI $h_i(t)$ for WiFi ACP with $N = 20$ nodes is 5.76 seconds. By controlling the packet generation rates at the source nodes, ACP *improves the average AoI by a factor of four* when compared with WiFi UDP. Notice that a high packet generation rate may overload the network and lead to a high average AoI, while a low packet generation rate may result in infrequent information updates at the destination, which may also lead to a high average AoI. The ACP dynamically adapts the packet generation rates at the nodes in order to drive the network to the point of optimal information freshness. This point of minimum AoI is illustrated in Fig. 5-3. Details about ACP can be found in [92].

**WiFresh.** The measurements in the *third row* of Table 5.1 show that the average AoI $h_i(t)$ for WiFresh with $N = 20$ nodes is 0.54 seconds. WiFresh *improves the average AoI by a factor of forty* when compared with WiFi UDP. Experimental results in Sec. 5.4 show that this improvement increases for larger $N$. The superior performance of WiFresh is due to the combination of three elements: Last-Come First-Served (LCFS) queues, Polling Multiple Access mechanism, and Max-Weight scheduling policy. The choice of each of these elements is underpinned by theoretical research. The LCFS queue was shown to be the optimal queueing discipline in terms of information freshness in different settings [10,21,60]. The Polling Multiple Access with Max-Weight scheduling policy was analyzed in terms of average AoI in chapter 4 and in [42,44,51].

**Scalability problem.** Neither WiFi UDP nor WiFresh attempt to control the packet generation rate at the source nodes. Hence, when the number of nodes $N$ increases to the point that the cumulative packet generation rate exceeds the capacity of the network, the network becomes overloaded and the number of backlogged packets grows rapidly. For

WiFi UDP, a large backlog in the First-Come First-Served (FCFS) queues leads to high packet delay and, thus, to high average AoI. In contrast, as observed in Table 5.1, *WiFresh scales gracefully even when the network is overloaded*, with average AoI increasing linearly[1] with $N$. This is due to:

- the **Polling mechanism** that *prevents packet collisions*, allowing for efficient resource allocation among nodes, which is critical in networks with large $N$;
- the **Max-Weigh policy** that determines the sequence of nodes to poll in order to *optimize information freshness*, keeping the AoI of each node as low as possible; and
- the **LCFS queues** that prioritize the *packet with lowest delay*, leading nodes to always transmit the freshest packets to the destination.

**Applications of WiFresh.** WiFresh is designed to support time-sensitive applications that rely on the knowledge of the current state of the system. For example: monitoring mobile ground-robots in automated fulfillment warehouses at Amazon [107, 111] and Alibaba [89]; collision prevention applications [61] for vehicles on the road [3,16,27,63]; path planning, localization and motion control for multi-robot formations using drones [2,4] and using ground-robots [106]; multi-drone system for tracking a mobile spectrum cheater [88]; multi-drone system for automated aerial cinematography [83]; multi-drone system for exploration of subterranean environments [79]; multi-robot simultaneous localization and mapping (SLAM) using drones [69, 77] and using ground-robots [75]; real-time surveillance system using a fleet of ground-robots [80]; and data collection from sensors, drones and cameras for agriculture using the Azure FarmBeats IoT platform [56, 108].

*The various time-sensitive applications in* [2–4, 16, 27, 56, 63, 69, 75, 77, 79, 80, 83, 88, 89, 106–108, 111] *are all implemented using the IEEE 802.11 standard (WiFi).* WiFi is an attractive choice for it is low-cost, well-established, and immediately available in drones [83], computing platforms running the Robot Operating System (ROS) [80], sensors that measure soil temperature, pH, and moisture [108], and in the Raspberry Pis used in this chapter. Moreover, as showcased by these various implementations and by the results in the first row of Table 5.1, *small-scale underloaded* WiFi networks are able to support time-

[1]Notice that the universal lower bound in Theorem 2.1 shows that the optimal average AoI cannot scale better than linearly on $N$.

sensitive applications. Two main shortcomings of WiFi, or any other wireless technology employing FCFS queues and Random Access mechanisms, are scalability and congestion.

**Our contribution**. Leveraging the theoretical results in previous chapters, we propose WiFresh: a network architecture that scales gracefully, achieving near optimal information freshness for wireless networks of any size, even when the network is overloaded. We propose and realize two strategies for implementing WiFresh: WiFresh Real-Time, in which LCFS queues, Polling mechanism and Max-Weight scheduling policy are implemented at the *MAC layer* in a network of eleven FPGA-based Software Defined Radios (Fig. 5-7) using hardware-level programming and operating at the *microsecond* time-scale; and WiFresh App which is a customization of WiFresh implemented at the *Application layer*, without modifications to lower layers of the communication system, in a network of twenty five Raspberry Pis (Fig. 5-9) using Python 3. WiFresh App runs over standard WiFi UDP, *manipulating WiFi UDP into behaving as WiFresh*, making it easy to integrate into applications that already run over WiFi such as [2–4, 16, 27, 56, 63, 69, 75, 77, 79, 80, 83, 88, 89, 106–108, 111]. Our experimental results in Sec. 5.4 show that WiFresh can improve information freshness by two orders of magnitude when compared to an equivalent standard WiFi UDP network. *To the best of our knowledge, this is the first work to propose and experimentally evaluate a practical wireless network architecture that scales gracefully in terms of information freshness.*

The remainder of this chapter is organized as follows. In Sec. 5.1, we describe related work on AoI. In Sec. 5.2, we discuss the impact of the multiple access mechanism, transmission scheduling policy, and queueing discipline on information freshness. In Sec. 5.3, we describe the design and implementation of WiFresh Real-Time and WiFresh App. In Sec. 5.4, we evaluate the performance of WiFresh in a network with increasing load and in a network with increasing size. The chapter is concluded in Sec. 5.5.

## 5.1 Related Work

Most papers on AoI focus on theory and a few consider system implementation. A literature review of theoretical works is provided in Sec. 1.2. System implementation is considered in

[57,92,95]. In [95], the authors consider a source-destination pair transmitting packets over the Internet and measure the AoI for different packet generation rates. In [57], the authors consider a vehicular network and develop an Application layer algorithm that adapts the packet generation rates at the sources to improve information freshness. This algorithm is validated using the ORBIT testbed with wireless nodes employing WiFi, in particular the IEEE 802.11a standard. In [92], the authors consider an Internet-of-Things network and develop a Transport layer protocol named Age Control Protocol (ACP) that adapts the packet generation rates at the sources in order to optimize information freshness. This protocol is validated using ten sources connected via WiFi to the Internet and sending packets to a destination in another continent. In Sec. 5.4, we implement ACP and evaluate its performance against WiFresh.

## 5.2 Background on Age of Information

The AoI $h_i(t)$ measures the time elapsed since the generation of the freshest packet received by the destination from source $i$. The evolution of the AoI process is illustrated in Fig. 1-2 which is displayed below for convenience. The time-average expected AoI associated with source $i$ is given by $\int_{t=0}^{T} \mathbb{E}[h_i(t)]dt/T$. From Fig. 5-2, we can see that to keep the information at the destination as fresh as possible, i.e. minimize the time-average expected AoI, it is necessary to simultaneously provide: i) low packet delay; ii) high data throughput; and iii) service regularity[2]. To minimize AoI, we consider the network as a whole and optimize the system across the queueing discipline, the multiple access mechanism and the transmission scheduling policy. Next, we discuss each of them in detail.

### 5.2.1 Queueing Discipline

The queueing discipline employed at the source nodes is central for minimizing AoI. In this section, we compare FCFS and LCFS queues and evaluate their performance in terms of AoI. FCFS queues are widely deployed in communication systems and they are the basis

[2]It is important to emphasize the difference between delivering packets regularly and providing a minimum throughput. In general, a given minimum throughput can be achieved even if long periods with no delivery occur, as long as those are balanced by short periods of consecutive packet deliveries.

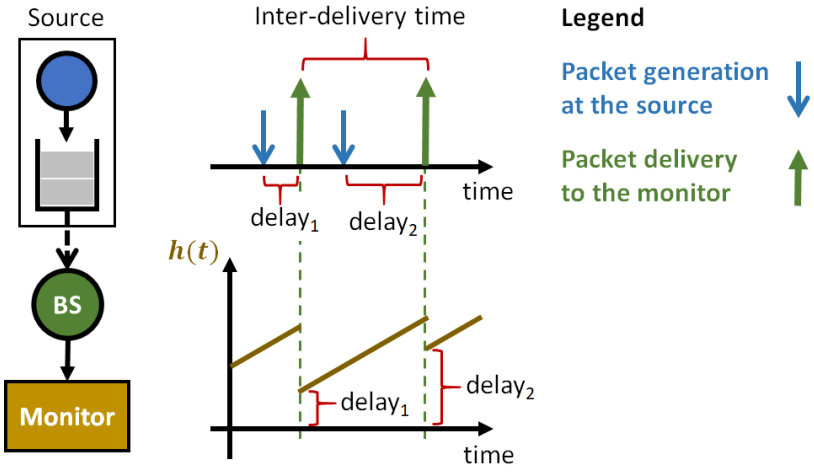



Figure 5-2: Illustration of the AoI evolution in a network with a single source sending packets to a single destination through a wireless base station (BS). Packets generated at the source wait in the queue before being served.

for other disciplines such as Priority Queueing and Fair Queueing. FCFS queues transmit packets in order of arrival, meaning that the *freshest packet is always placed at the tail of the queue*. Under heavy loads, the FCFS queue is often backlogged and the freshest packet has to wait for a long queueing delay before being delivered to the destination. The high queueing delay leads to stale information at the destination and to high AoI. Naturally, this effect is more prominent for large FCFS queues, as discussed in Sec. 5.4.1.

LCFS queues are often considered in the AoI literature [10, 11, 21, 60, 87], but they are not commonly deployed in communication systems. LCFS queues place the *most recently generated packet at the Head-of-Line (HoL)*, leading to source nodes that transmit the freshest packet first, which makes LCFS queues ideal for applications that rely on the knowledge of the current state of the system, i.e. applications that need fresh information at the destination. Under heavy loads, the LCFS queue is frequently replacing its HoL packet with fresher packets. We expect that the higher the packet generation rate at the source nodes, the lower the average AoI at the destination. LCFS queues are not commonly found in communication systems. LCFS is not one of the queueing discipline (qdisc) options in Linux nor in the Software Defined Radios (SDRs) we utilized for implementing WiFresh. In both cases, as expected, the standard queueing discipline is FCFS.

**Comparing FCFS and LCFS.** Consider an M/M/1 queueing system with infinite queue

size, fixed packet service rate of $\mu = 1$ packet per second and variable packet generation rate $\lambda$, employing either FCFS or LCFS discipline. In Fig. 5-3, we display the time-average expected AoI for FCFS and LCFS, the expected packet delay and the expected inter-delivery time. The analytical expressions for the AoI associated with FCFS and LCFS queues were obtained in [59] and [21], respectively, and the expressions for packet delay and inter-delivery time can be found in [30].

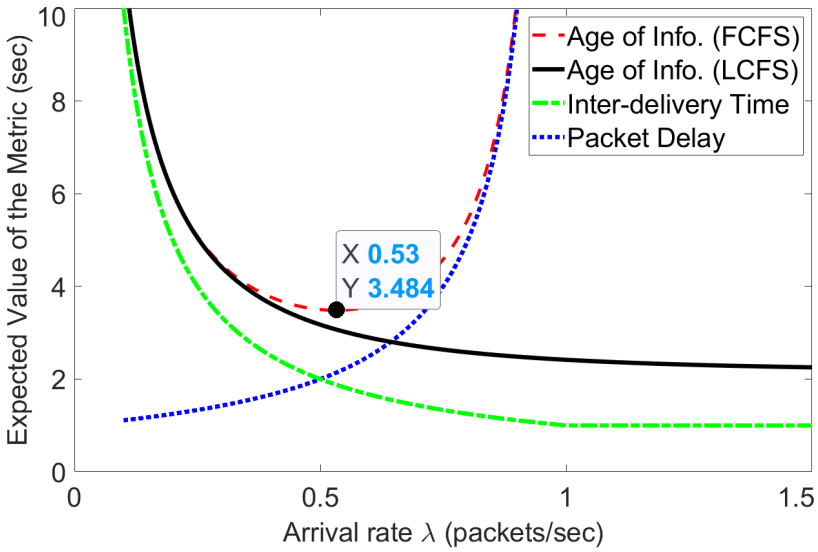



Figure 5-3: Expected delay, expected inter-delivery time and expected average AoI of the queueing system with service rate of $\mu = 1$ and variable packet generation rate $\lambda$.

**Choice of LCFS for WiFresh.** From Fig. 5-3, we can see that the minimum time-average AoI for FCFS queues is achieved at moderate loads, in particular $\lambda/\mu \approx 0.53$, while for LCFS queues the higher the packet generation rate $\lambda$, the lower the AoI. In addition, LCFS outperforms FCFS for every packet generation rate $\lambda$. In fact, LCFS was shown to be the optimal queueing discipline in different settings including single queue systems [21, 60, 87], single-hop wireless networks [11] and multi-hop wireless networks [10]. Thus, we propose to use the LCFS discipline in WiFresh.

**Effect of dropping packets.** Nodes with LCFS queues transmit the freshest packet first. Notice that when a packet with *older information* is delivered to the destination after a packet with *fresher information*, the freshness of the information is not affected and, thus, the value of $h_i(t)$ *remains unchanged*. Hence, if packets with older information were

dropped at the source as soon as a fresher packet arrived to the LCFS queue, the information freshness at the destination would not be affected. It follows that, *from the perspective of AoI, a LCFS queue is equivalent to a head-drop FCFS queue of size* 1 *packet, in which only the freshest packet is kept*. The advantage of dropping older packets at the source is saving communication resources. One possible disadvantage is that dropped packets might contain useful information. For example, in a position tracking system, older packets can be used to predict future movement.

### 5.2.2 Multiple Access Mechanism

Consider the network in Fig. 5-4 with $N$ source nodes sending time-sensitive information to the remote monitor via the wireless BS. Packets are generated at the sources and enqueued in separate queues. The multiple access mechanism controls the method utilized by each of the $N$ sources for sharing the common wireless channel. In this section, we compare two types of multiple access mechanism, Random Access and Polling, in terms of information freshness.

To capture the freshness of the information *in the network*, we define the Expected Weighted Sum AoI (EWSAoI) as

$$EWSAoI = \lim_{T\to\infty} \frac{1}{TN} \int_{t=0}^{T} \sum_{i=1}^{N} w_i \mathbb{E}\left[h_i(t)\right] dt \ , \tag{5.1}$$

where $T$ is a positive real number that represents the time-horizon and $w_i$ is a positive real number that depicts the weight (or priority) of source $i$. To minimize the EWSAoI, the multiple access mechanism should attempt to: i) provide high communication efficiency by avoiding packet collisions and reducing the control overhead; and ii) prioritize transmissions from sources with high current AoI $h_i(t)$, favorable channel conditions and high weight $w_i$.

**Random Access** is a widely deployed class of multiple access mechanisms, e.g. WiFi, ZigBee, Wireless Body Area networks [68], and traditional cellular systems such as GSM. The fundamental idea is that, when a source has a packet to transmit, it uses a randomized algorithm to contend for channel access. Randomization is employed to reduce the proba-

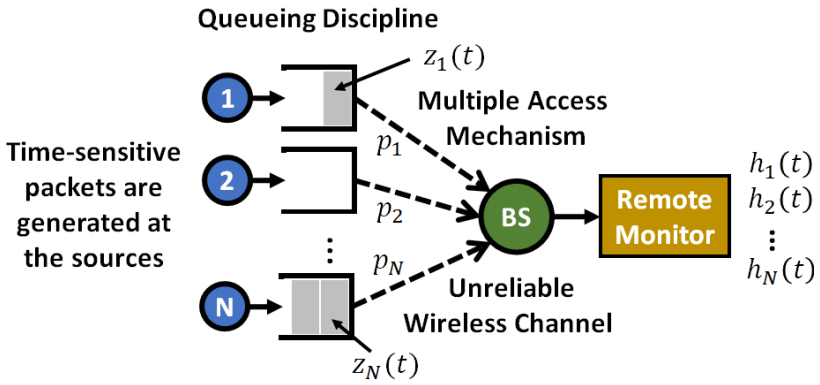



Figure 5-4: Illustration of the network with $N$ source nodes sending time-sensitive information to the remote monitor via the wireless BS.

bility of two or more sources transmitting packets simultaneously, which would result in a packet collision. Random Access mechanisms are discussed in detail in chapter 6. Some advantages of Random Access are simplicity, decentralization and low control signaling overhead. Two disadvantages are the probability of packet collision that increases with the number of sources $N$ and the distributed operation that makes it challenging to implement a dynamic transmission prioritization based on parameters such as AoI $h_i(t)$ and/or current channel conditions.

**Polling mechanism** is a well-known alternative to Random Access. The BS coordinates the communication in the network by sending *poll packets* to the sources selected for transmission, as illustrated in Fig. 5-5. The BS selects the next source to poll based on the *scheduling policy*, which may be a function of dynamic parameters such as AoI $h_i(t)$ and/or current channel conditions. The polling mechanism attempts to eliminate packet collisions and enables dynamic prioritization, making it suitable for large-scale time-sensitive applications.

Two important challenges associated with polling mechanisms are the control overhead and the choice of scheduling policy. *Control overhead:* the BS transmits a poll packet before receiving each data packet. In contrast, Random Access may require that the BS transmit an acknowledgment packet following the reception of each data packet. Hence, the control overhead of both mechanisms is comparable. *Scheduling policy:* the BS dynamically chooses the next source to poll. Evidently, a naive policy can degrade the perfor-

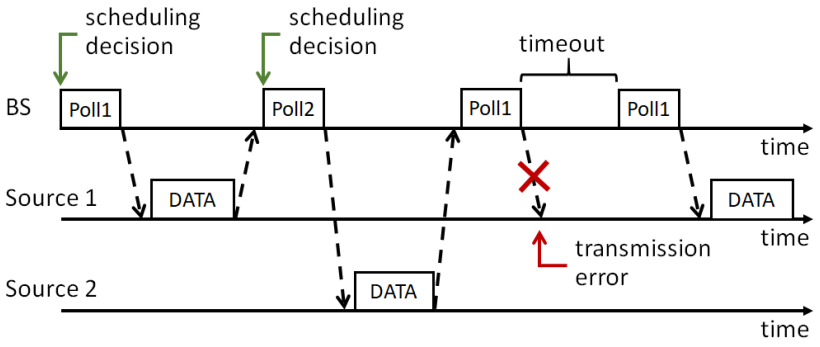



Figure 5-5: Polling mechanism with the BS controlling the channel access for N=2 sources.

mance. Next, we discuss scheduling policies that are designed to optimize the information freshness in the network.

### 5.2.3 Scheduling Policy

The problem of obtaining an optimal scheduling policy for single-hop wireless networks in terms of information freshness was shown in [32] to be NP-hard. Numerous heuristic policies based on Approximate Dynamic Programming [38], Restless Multi-Armed Bandits [51,52] and Lyapunov Optimization [42,44,50,51] have been proposed in the literature. The MW policy is chosen for WiFresh because it is intuitive, low-complexity and has superior performance guarantees, as seen in chapter 4.

**Max-Weight (MW) policy.** Consider the network in Fig. 5-4 employing LCFS queues and a Polling mechanism. Assume that $t$ is the current decision time of the next poll packet. Let $p_i \in (0,1]$ be the channel reliability associated with source $i$, namely the probability of a successful reception of a data packet following the transmission of a poll packet to source $i$. Let $\tau^{HoL}(t)$ be the time-stamp of the current Head-of-Line packet from source $i$ at time $t$ and let $z_i(t) = t - \tau^{HoL}(t)$ be the *current system time of this HoL packet*. Notice that if this HoL packet were delivered to the BS at time $t$, then $z_i(t)$ would be the associated packet delay and the AoI would be reduced from $h_i(t)$ to $z_i(t)$, as illustrated in Fig. 5-2. Hence, the difference $h_i(t) - z_i(t)$ represents the *potential AoI reduction* of polling source $i$ at time $t$. Assume that the scheduling policy knows the values of $z_i(t)$ and $p_i$, and denote $\mathscr{I}(i,t) := w_i p_i (h_i(t) - z_i(t))^2$ as the index of source $i$ at time $t$. Then, the MW policy

selects, at every decision time $t$, the source $i^*(t)$ with highest value of $\mathscr{I}(i,t)$, with ties being broken arbitrarily. Intuitively, the MW is polling the source with highest weighted potential AoI reduction. This particular expression for $\mathscr{I}(i,t)$ was derived in Sec. 4.4. The assumptions of known $z_i(t)$, and static and known $p_i$ were utilized in chapter 4 to derive performance guarantees in terms of information freshness. In this chapter, we do not assume knowledge of $z_i(t)$ and $p_i$. In WiFresh, we estimate both $z_i(t)$ and $p_i$ over time, as described in Sec. 5.3.2. For building intuition into the MW policy, we provide a simple example.

**Example.** Consider an ideal symmetric network in which all sources have the same weight $w_i = w > 0$, the same channel reliability $p_i = p \in (0,1]$, and all sources generate fresh packets continuously such that their LCFS queues always have fresh HoL packets, meaning that $z_i(t) = 0, \forall i,t$. Under these conditions, the MW policy selects, at each decision time $t$, the source $i^*(t) = \arg\max\{\sqrt{w_i p_i}(h_i(t) - z_i(t))\} \equiv \arg\max\{h_i(t)\}$. This intuitive policy, which always polls the source with most outdated information, is called Maximum Age First (MAF), and it was shown in Theorem 2.7 to achieve optimal information freshness (i.e. minimum EWSAoI) for the case of ideal symmetric networks. Both the MW policy and the MAF policy are implemented in real networks and evaluated in Sec. 5.4.

## 5.3 Design and Implementation

In this section, we discuss the design and implementation of WiFresh in its two forms: WiFresh Real-Time and WiFresh App. The design and implementation of WiFresh Real-Time is discussed in Sec. 5.3.2 and Sec. 5.3.3, respectively, and the design and implementation of WiFresh App is discussed in Sec. 5.3.4 and Sec. 5.3.5, respectively. Their performance is evaluated in Sec. 5.4. Prior to delving into the details, we describe the main challenges.

### 5.3.1 Challenges

**Complexity of implementation.** To achieve high performance, WiFresh Real-Time was implemented at the *MAC layer* using a network of FPGA-based Software Defined Radios. The Polling mechanism and the MW scheduling policy were fully implemented in FPGAs with 10 MHz clocks, enabling WiFresh Real-Time to make the scheduling decision and trigger the transmission of the next poll packet in approximately 20 microseconds. Keeping this time-interval short and limiting the length of the poll packet are important factors in reducing the control overhead and achieving high performance. The results in Sec. 5.4.2 show that WiFresh Real-Time can improve information freshness by a factor of 200 when compared to an equivalent WiFi network. The main challenge of implementing WiFresh Real-Time at the MAC layer is the high complexity associated with implementing numerous real-time functions (such as Polling, MW policy, queueing discipline and estimation algorithms) using hardware-level programming.

**Barrier to adoption.** Targeting an alternative implementation of WiFresh that could be easily integrated into applications that already run over WiFi such as [2–4, 16, 27, 56, 63, 69, 75, 77, 79, 80, 83, 88, 89, 106–108, 111], we propose WiFresh App which is implemented in Python 3 and runs at the *Application layer*, without modifications to lower layers of the communication system. The main challenge is in the design of a Python application that is capable of driving a standard WiFi UDP network (with FCFS queues and Random Access) to behave as a WiFresh network (with LCFS queues and Polling mechanism with MW policy). This design is discussed in Sec. 5.3.4. The experimental results in Sec. 5.4.3 show that WiFresh App can improve information freshness by a factor of 65 when compared to an equivalent WiFi UDP network.

**Bridging theory and practice.** Theoretical works on Age of Information often assume that: 1) nodes in the network are synchronized; 2) nodes generate packets on-demand or according to simple stochastic processes; 3) each source is associated with a single type of information such as position, inertial measurements or images; 4) each data packet contains a complete information update; 5) channel reliabilities $\{p_i\}_{i=1}^{N}$ are fixed and known; and/or 6) system times of HoL packets $\{z_i(t)\}_{i=1}^{N}$ are known. To leverage the theory and im-

plement (for the first time) an AoI-based network architecture composed of LCFS queues and Polling mechanism with MW scheduling policy, we augment WiFresh with algorithms that synchronize clocks, dynamically learn $\{p_i\}_{i=1}^N$ and $\{z_i(t)\}_{i=1}^N$, manage sources with multiple information types, and manage packet fragmentation.

**Fragmentation of information updates.** The AoI is reduced when fresh information is received at the destination. The evolution of $h_i(t)$ assumes that each data packet contains a complete information update. To accommodate large information updates, such as images, WiFresh has to manage packet fragmentation. Two issues are discussed below.

The first issue is when to reduce the AoI $h_i(t)$. In general, the AoI $h_i(t)$ can be reduced (or partially reduced) upon reception of a subset of fragments. In this work, fragmentation is used for transmitting images and, in this case, it makes sense to reduce AoI only when all fragments are received. The second issue is whether the LCFS queue should replace the HoL packet as soon as a new information update arrives, or if the LCFS queue should wait until all fragments from the previous information update are delivered before replacing the HoL packet. Notice that if information updates are generated with a high rate, then replacing the HoL packet as soon as a new information update arrives may hinder the *complete transmission* of information updates. For this reason, in this work, we choose to transmit all fragments before replacing the information update at the LCFS queue.

WiFresh Real-Time runs at the MAC layer. Hence, it is blind to the concept of *information* and can only see individual data packets. This makes the adjustments discussed above challenging to implement. To overcome this problem, WiFresh Real-Time could gather information regarding fragmentation from other layers of the communication system. In this work, we implement fragmentation only in WiFresh App, which runs at the Application layer and is aware of information updates. Recall that information updates are generated and received by the Application layer.

### 5.3.2 Design of WiFresh Real-Time

In this section, we describe WiFresh Real-Time (WiFresh RT) in detail. The layers of the communication system are illustrated in Fig. 5-6. Next, we describe the three types of packets used in WiFresh RT:

- **Poll packet** is sent from the BS to a selected source to indicate that the source can transmit one data packet. The duration of the data packet transmission is limited to $TXOP$ milliseconds;

- **Data packet** is sent from the source to the BS following a poll packet. The data packet contains a time-stamp indicating when the packet reached the MAC layer.

- **Empty packet** is sent from the source to the BS following a poll packet when there is no data packet to be transmitted, i.e. when the queue at the polled source is empty. The Empty packet is used for the BS to differentiate between not receiving data due to a transmission error or due to an empty queue at the source. This differentiation is useful for estimating $p_i$ and $z_i(t)$.

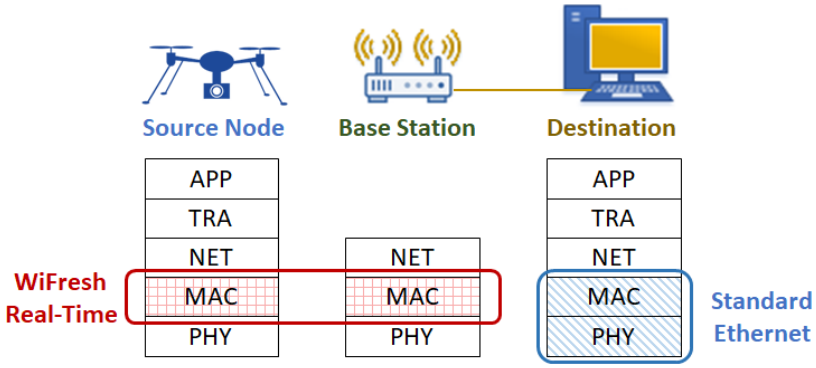



Figure 5-6: Components of the WiFresh RT system. The MAC layers at the source and BS are emphasized for they are central to the implementation of WiFresh RT.

**Basic source behavior.** The source generates information updates in the Application layer and forwards them to lower layers of the communication system. When a data packet arrives at the MAC layer, WiFresh RT appends a time-stamp to the packet and then stores it in a head-drop FCFS queue of size 1 packet. Recall that this queue keeps only the freshest packet and discards older packets. The source can be in one of two states: 1) waiting for a poll packet from the BS; or 2) transmitting the freshest data packet to the BS. While waiting for a poll packet, the source manages its queue. Upon receiving a poll packet, the source fetches a data packet from its queue and transmits this packet to the BS through the wireless channel. If the queue is empty, the source transmits an empty packet to the

BS. After transmitting either the data packet or the empty packet, the source goes back to waiting for the next poll packet.

**Basic BS behavior.** The BS does not generate data packets. Its main responsibility is to coordinate the communication in the network. The BS can be in one of two states: 1) waiting for a data packet; or 2) transmitting a poll packet. While waiting for a data packet, the BS keeps track of the waiting period. If the waiting period exceeds $BS\_Timeout = 100$ microseconds or a data packet is received, the BS updates its estimate of the network state $(\hat{h}_i(t), \hat{z}_i(t), \hat{p}_i(t))_{i=1}^N$, where $\hat{h}_i(t)$ is the estimate of $h_i(t)$, $\hat{z}_i(t)$ is the estimate of $z_i(t)$ and $\hat{p}_i(t)$ is the estimate of $p_i$ at time $t$. These estimates are used by the MW policy to determine the source $i^*(t)$ with highest index $\mathscr{I}(i,t) = w_i \hat{p}_i(t)(\hat{h}_i(t) - \hat{z}_i(t))^2$. After transmitting a poll packet to source $i^*(t)$, the BS goes back to waiting for the next data packet, as illustrated in Fig. 5-5. The algorithms used to estimate $h_i(t)$, $p_i$ and $z_i(t)$ are discussed next.

**Clock synchronization** is needed to accurately compute $h_i(t) := t - \tau_i(t)$, where $t$ is the current time measured by the BS and $\tau_i(t)$ is a time-stamp created by source $i$. If clocks are not synchronized, the values of $h_i(t)$ for different sources may have different biases, which may lead to wrong scheduling decisions by the MW policy. To estimate the time-stamp offset between each source and the BS, and obtain the estimates $\{\hat{h}_i(t)\}_{i=1}^N$, some possible approaches are: adding GPS antennas to every node in the system and then using GPS time; synchronizing the Operating System (OS) of every node using the Network Time Protocol [82] via the Internet and then using the OS time; or implementing a synchronization algorithm as part of WiFresh. In WiFresh RT we use the OS time. In WiFresh App we implement a built-in synchronization algorithm described in Sec. 5.3.4.

**Learning channel reliability.** To estimate the value of $p_i \in (0,1]$ associated with each source $i$, we implement a simple estimator. Let $\mathscr{P}_i(t)$ be the number of poll packets transmitted to source $i$ in the last $\mathscr{W}$ seconds and let $\mathscr{D}_i(t)$ be the number of data packets and empty packets successfully received from source $i$ in the same period. Then, the estimate of $p_i$ at time $t$ is given by

$$\hat{p}_i(t) = \frac{\mathscr{D}_i(t)+1}{\mathscr{P}_i(t)+1}\,. \tag{5.2}$$

We choose a time window of $\mathscr{W} = 0.5$ seconds. Notice that when the number of poll

packets $\mathscr{P}_i(t)$ is low, the estimate $\hat{p}_i(t)$ tends to be optimistic, i.e. higher. In particular, when $\mathscr{P}_i(t) = \mathscr{D}_i(t) = 0$, we have $\hat{p}_i(t) = 1$. This high value of $\hat{p}_i(t)$ when the number of poll packets is low creates an incentive for the MW policy to select sources that have not been polled recently.

To determine the changes in $\mathscr{D}_i(t)$ and $\mathscr{P}_i(t)$ for each source $i$ over time, we log the transmission and reception events within the window $\mathscr{W}$ using large arrays. This log is created at the on-board processor of the SDR (as opposed to the FPGA) in order to spare the limited FPGA resources. This design choice imposes the following sequence of events: 1) the FPGA processes the transmission/reception events and communicates them to the on-board processor; 2) the processor logs the events, calculates $\hat{p}_i(t)$ and communicates the updated estimate to the FPGA; 3) the FPGA uses the latest value of $\hat{p}_i(t)$ as input to the MW policy. The disadvantage of keeping the log at the processor is the added round-trip communication delay between on-board processor and FPGA which is of approximately 500 microseconds. Since $\hat{p}_i(t)$ represents the average channel reliability in the last $\mathscr{W} =$ 0.5 seconds, it follows that this relatively small round-trip communication delay has a negligible impact on the performance of the MW policy. The estimate of $p_i$ is the only portion of WiFresh RT which is not fully implemented at the FPGA.

**Learning the system times.** Recall that the difference $h_i(t) - z_i(t)$ represents the *potential AoI reduction* of polling source $i$ at time $t$ and that the MW policy wishes to use this difference for selecting the appropriate source to poll. The problem is that the MW policy does not know the system times of the HoL packets $\{z_i(t)\}_{i=1}^N$, which are only known by the respective sources, as illustrated in Fig. 5-4. One approach for estimating $z_i(t)$ could be to develop an algorithm that generates estimates $\hat{z}_i(t)$ based on the entire history of transmission and reception events, especially the sequence of previously received time-stamps. The main drawback of this approach is its complexity. A less accurate but much simpler approach is to estimate $z_i(t)$ based on the latest received packet only. In particular, we know that when the freshest data packet from source $i$ is received at time $t$, the potential AoI reduction of polling source $i$ again at time $t$ is (most likely[3]) zero, which is represented

[3]The potential AoI reduction of polling source $i$ again at time $t$ might be greater than zero if source $i$ generates a new data packet while the previous data packet was being transmitted. We assume that this is an unlikely event and neglect its effect.

by $z_i(t) = h_i(t)$. Similarly, when an empty packet is received at time $t$, the potential AoI reduction of polling source $i$ again at time $t$ is (most likely) zero. Hence, we can estimate $z_i(t)$ using the following mechanism:

- $\hat{z}_i(t) \leftarrow \hat{h}_i(t)$ following the successful reception of a data packet or an empty packet from source $i$ at time $t$; and
- $\hat{z}_i(t)$ remains constant over time while no packet is received.

*This simple mechanism prevents the MW policy to repeatedly schedule the same source $i$ with a high AoI $h_i(t)$ and an empty queue.* Notice that estimation errors in $\hat{h}_i(t)$, $\hat{p}_i(t)$ or $\hat{z}_i(t)$ affect the performance of the MW policy only when they lead to wrong scheduling decisions. In the next section, we discuss the implementation of WiFresh RT.

### 5.3.3 Implementation of WiFresh Real-Time

We implement WiFresh RT in a Software Defined Radio testbed composed of a desktop computer operating as the remote monitor, one NI USRP 2974 operating as the wireless base station, and ten nodes operating as sources: seven NI USRP 2974 and three NI USRP 2953R, as shown in Fig. 5-7. The code is developed using the LabVIEW Communications 802.11 Application Framework v3.0 (802.11 AFW). The 802.11 AFW [40] is a modifiable reference design of WiFi with Transport layer based on UDP, MAC layer based on the Distributed Coordination Function (DCF), PHY layer based on Orthogonal Frequency-Division Multiplexing (OFDM), and no Network Layer. We use this WiFi reference design as a starting point to implement WiFresh RT.

The 802.11 AFW is composed of two main codes: the Host code (running at the on-board Intel i7 6822EQ 2 GHz Quad Core processor) and the FPGA code (running at the Xilinx Kintex-7 XC7K410T FPGA). The Host code is responsible for the generation/reception of data packets, radio configuration, and displaying measurements and plots. The FPGA code is responsible for processing data packets, generating control packets (e.g. Clear-to-Send, Request-to-Send and Acknowledgments), accessing the wireless channel using DCF, time management (e.g. Interframe spaces and timeouts), etc. The FPGA code allows us

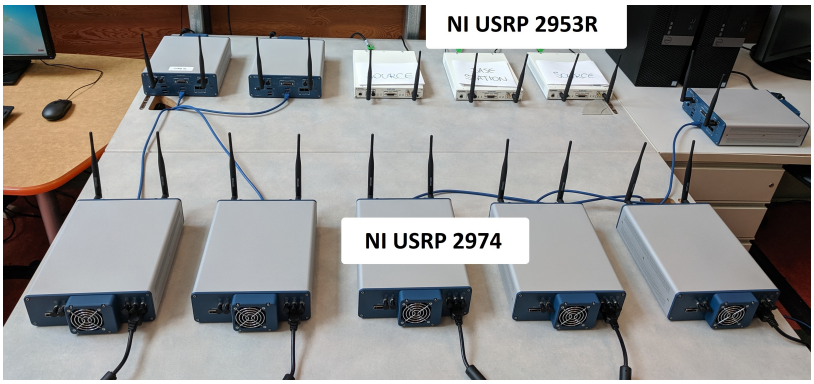



Figure 5-7: WiFresh RT testbed.

to implement real-time functions at the hardware level. The FPGA clock is of 10 MHz, meaning that these functions run at the microsecond time-scale.

*For implementing WiFresh RT, we created numerous real-time functions at the FPGA*, including: 1) Polling Multiple Access mechanism; 2) Max-Weight scheduling policy; 3) head-drop FCFS queue with size 1 packet; 4) time-stamp processing; 5) learning algorithms; and 6) measurement logs. The PHY layer of the WiFi reference design was kept unchanged. The PHY layer is based on the IEEE 802.11n standard and it has center frequency 2.437 GHz, bandwidth of 20 MHz and a fixed MCS index of 5.

### 5.3.4 Design of WiFresh App

WiFresh App is an implementation of WiFresh that aims to be easily integrated into time-sensitive applications that already run over WiFi such as [2–4, 16, 27, 56, 63, 69, 75, 77, 79, 80, 83, 88, 89, 106–108, 111]. WiFresh App is implemented in Python 3 and runs at the Application layer, without modifications to lower layers of the communication system, as illustrated in Fig. 5-8. This Python application is designed to drive a standard WiFi UDP network (with FCFS queues and Random Access) to behave as a WiFresh network (with LCFS queues and Polling mechanism with MW policy). WiFresh App contains all elements of WiFresh RT and some additional features, namely fragmentation of large information updates, a built-in synchronization algorithm, and support for sources that generate multiple types of information. Next, we describe WiFresh App.

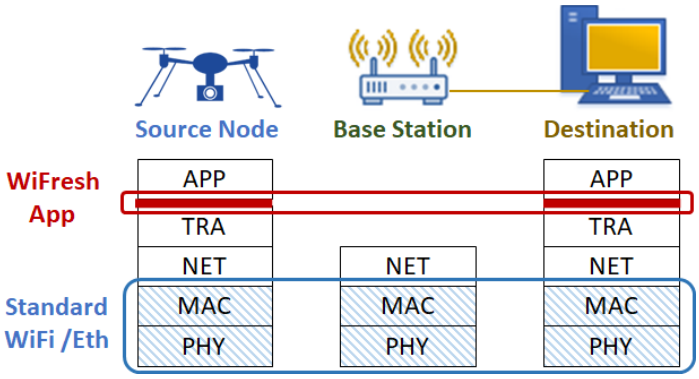



Figure 5-8: Components of the WiFresh App system. The Application layer at the source and destination is emphasized for it is central to the implementation of WiFresh App.

**Basic source behavior.** The source generates information updates at the Application layer. WiFresh App time-stamps the information updates and stores them in a LCFS queue, which is implemented using a *Python LIFO stack*. The LCFS queue releases a single information update only when the source receives a poll packet from the destination. If the released information update fits into a single data packet, this information update is encapsulated into a data packet and forwarded to lower layers of the communication system. Otherwise, the information update is fragmented, stored and the first data packet is forwarded. Fragments are stored in a FCFS queue which is separate from the LCFS queue containing information updates. Upon receiving the next poll packet acknowledging the first fragment, the source forwards the second fragment, and so on, until all fragments are successfully delivered to the destination. When the poll packet acknowledging the final fragment is received, the LCFS queue releases the next information update, which is then fragmented, stored and transmitted following the same procedure.

When a fragment reaches the source's MAC layer, WiFi stores it in a FCFS queue and transmits it to the destination using Random Access. Ideally, since the destination only generates a new poll packet after the previous fragment is received, there should be *at most one source* attempting transmission using Random Access at any given time. This means that, even when all sources are generating information updates with a high rate, the underlying WiFi network is handling one data packet at a time, resulting in low packet delay and low probability of transmission errors, which still may occur due to the unreliable nature

of the wireless channel. When transmission errors occur, WiFi may attempt to retransmit the data packet. Notice that by implementing LCFS queues and Polling mechanisms with MW policy at the Application layer, *WiFresh App is driving the underlying WiFi network to operate as a WiFresh network.*

**Basic destination behavior.** Similarly to WiFresh RT, the destination in WiFresh App generates poll packets, implements a timeout of *Destination_Timeout* = 300 milliseconds, updates its estimate of the network state $(\hat{h}_i(t), \hat{z}_i(t), \hat{p}_i(t))_{i=1}^N$, and uses the MW policy to decide which source to poll next. The main differences are that scheduling decisions are made at the Application layer at the millisecond time-scale, as opposed to the MAC layer at the microsecond time-scale, and that the destination manages the fragmentation procedure described above, uses a built-in clock synchronization algorithm to estimate $\hat{h}_i(t)$, and supports sources that generate multiple types of information.

**Clock synchronization.** Let $t^D$ be the current time measured by the destination and let $t_i^S$ be the current time measured by source $i$. The time-stamp offset is given by $\Delta_i = t^D - t_i^S$. To estimate the value of $\Delta_i$ we implement a two-way handshake called the *on-wire protocol* that is part of the Network Time Protocol. Details about the handshake are provided in Appendix 5.A and in [82, Sec. 8]. To synchronize the entire network, the destination performs a two-way handshake with each source $i \in \{1, 2, \cdots, N\}$. The synchronization procedure runs periodically with period *sync_period* = 300 seconds. The sequence of past and present time-stamp offset measurements for each source $i$ is filtered and estimates of $\{\hat{\Delta}_i(t)\}_{i=1}^N$ are obtained. The AoI with time-stamp offset correction is given by $\hat{h}_i(t) = t - \tau_i(t) - \hat{\Delta}_i(t)$. As soon as the synchronization procedure ends, the destination resumes using the MW policy with $\hat{h}_i(t)$ to schedule data packet transmissions.

**Multiple information types per source.** The AoI is associated with a single type of information such as position, inertial measurements or images. In a network with sources that generate multiple types of information, we create for each tuple (source, information type) a separate instance of WiFresh App with independent LCFS queue, time-stamp offset, AoI $\hat{h}_i(t)$, channel reliability $\hat{p}_i$, system time $\hat{z}_i(t)$, and priority $w_i$. For example, in a network with two sources, each generating three types of information updates, there are seven instances of WiFresh App: three per source and one at the destination. The destina-

tion treats each instance of WiFresh App at the sources as an independent entity, and sends individual poll packets to each of them. For simplicity, in the description of WiFresh App that follows, we assume that each source has a single instance of WiFresh App, and use the terms *source* and *WiFresh App instance* interchangeably.

### 5.3.5 Implementation of WiFresh App

We implement WiFresh App in a Raspberry Pi (Raspi) testbed composed of one desktop computer operating as the remote monitor, one Raspberry Pi 3B+ with a WiFi USB adapter operating as the wireless BS, and twenty four nodes operating as sources: ten Raspberry Pi 3B+ fetching data from sensors and fourteen Raspberry Pi Zero W generating synthetic data that emulates the sensors. For the measurements in Sec. 5.4, nodes are static and placed indoors. The distance between sources and destination is between 2 and 3 meters. In Fig. 5-9, we display some of the sources[4] and the three sensors described below:

- cameras (Arducam 5 Megapixels 1080p) generating jpg images with resolution 256x144 pixels and size of approximately 19 kbytes at a rate of 2 Hz.
- Inertial Measurement Units (Pololu MinIMU-9 v5 Gyro, Accelerometer, and Compass) generating information updates of size 20 bytes at a rate of 100 Hz; and
- GPS units (Stratux GPYes 2.0 u-blox 8) generating information updates of size 50 bytes at a rate of 1 Hz;

To create synthetic GPS data for indoor environments, we use a NMEA sentence generator [6] that emulates the GPS unit.

The Raspberry Pis run the Raspbian Stretch OS and communicate via WiFi, in particular the IEEE 802.11g standard at 2.4 GHz. WiFresh App is implemented using Python 3. The main functionalities we developed are: 1) Polling mechanism; 2) Max-Weight scheduling policy; 3) LCFS queue; 4) fragmentation management; 5) time-stamp processing; 6) learning algorithms; 7) interface with sensors; 8) synthetic generation of data packets emulating each type of sensor; 9) graphical user interfaces; and 10) measurement logs. The

[4]The remote control cars and battery packs are used for running tests outdoors. The measurements in this chapter were performed indoors.

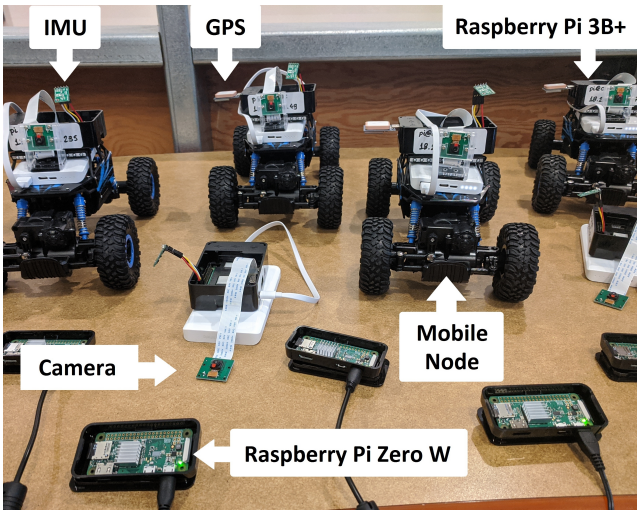



Figure 5-9: WiFresh App sources and their sensors.

Transport, Network, MAC and PHY layers were kept unchanged, as illustrated in Fig. 5-8. *WiFresh App is built over standard UDP, IP and WiFi protocols.*

## 5.4 Experimental Results

In this section, we evaluate the performance of WiFresh in a dynamic indoor University office space with *multiple external sources of interference* such as mobile phones, laptops and campus WiFi base stations. We evaluate WiFresh RT and WiFresh App, and compare them with other communication systems. In particular, using the SDR testbed described in Sec. 5.3.3, we compare:

- **WiFresh RT:** as described in Sec. 5.3.2;
- **WiFresh RT FCFS:** identical to WiFresh RT but with sources employing FCFS queues;
- **WiFi UDP FCFS**: UDP over standard WiFi; and
- **WiFi UDP LCFS**: UDP over standard WiFi but with sources employing LCFS queues instead of FCFS queues.

In addition, using the Raspi testbed described in Sec. 5.3.5, we compare:

- **WiFresh App:** as described in Sec. 5.3.4;
- **WiFresh MAF**: identical to WiFresh App but with a scheduling policy that, at every decision time $t$, selects the source $i^*(t)$ with highest value of AoI $h_i(t)$;
- **WiFi UDP FCFS**: UDP over standard WiFi;
- **WiFi TCP FCFS**: TCP over standard WiFi; and
- **WiFi ACP FCFS**: Age Control Protocol (ACP) over standard WiFi. ACP is a Transport layer protocol recently proposed in [92] that adapts the packet generation rate of each source $i$ in order to minimize the AoI in the network. Recall that in our testbed, the packet generation rate is fixed and determined by the associated sensor. Hence, in our implementation of ACP, we approximate the target packet generation rate by regularly discarding some of the packets before they reach the FCFS queue.

Next, we present the results of experimental evaluations of WiFresh.

### 5.4.1 Single Source with High Load

In this section, we consider a network with a destination, a wireless BS and a single source generating packets of 150 bytes with rate $\lambda \in \{5,6,7\}$ kHz. These short packets of 150 bytes represent Machine to Machine (M2M) status updates, and different values of $\lambda$ represent different levels of congestion. In Tables 5.2 and 5.3, we measure the time-average Age of Information (in seconds) and the effective throughput (in Mbps). The effective throughput is measured at the Application layer of the destination and, thus, it refers to the number of *useful bits* received per second. In Table 5.2, we consider WiFresh RT and WiFi UDP FCFS in the SDR testbed, and in Table 5.3, we consider WiFresh App and WiFi UDP FCFS in the Raspi testbed. Each experiment runs for 10 minutes.

**External interference.** The results in Tables 5.2 and 5.3 show that when the packet generation rate increases from 5 kHz to 7 kHz, the effective throughput does not change significantly, indicating that sources with $\lambda \geq 5$ kHz are saturated, i.e. always have data to transmit. Table 5.2 shows that the throughput of WiFresh RT is higher than the throughput

Table 5.2: Measurements using the SDR testbed.

| SDR | WiFresh RT | | WiFi UDP FCFS | |
|---|---|---|---|---|
| | AoI (sec) | Thr. (Mbps) | AoI (sec) | Thr. (Mbps) |
| $\lambda = 5$k | 0.003 | 4.866 | 0.306 | 2.406 |
| $\lambda = 6$k | 0.003 | 4.905 | 0.304 | 2.433 |
| $\lambda = 7$k | 0.004 | 4.412 | 0.320 | 2.328 |

Table 5.3: Measurements using the Raspi testbed.

| Raspi | WiFresh App | | WiFi UDP FCFS | |
|---|---|---|---|---|
| | AoI (sec) | Thr. (Mbps) | AoI (sec) | Thr. (Mbps) |
| $\lambda = 5$k | 0.040 | 0.229 | 224.8 | 1.242 |
| $\lambda = 6$k | 0.046 | 0.197 | 248.2 | 1.183 |
| $\lambda = 7$k | 0.042 | 0.208 | 242.3 | 1.281 |

of WiFi UDP FCFS. This is because WiFresh RT does not back off when competing with other wireless networks for channel access, making it less susceptible to external interference. Recall that WiFresh RT is not designed to coexist with other networks. *WiFresh RT is designed to support large-scale time-critical applications* and, to that end, its Polling Multiple Access mechanism attempts to leverage all the available communication resources. In contrast, WiFresh App runs over standard WiFi, making it as susceptible to external interference as WiFi. Table 5.3 shows that the throughput of WiFresh App is lower than the throughput of WiFi UDP FCFS. The main reason for the lower throughput is the *control overhead* associated with running a Polling mechanism over standard WiFi. Notice that acknowledgement packets follow the successful transmission of every poll and data packets, thus increasing the control overhead. Despite the lower throughput, WiFresh App significantly outperforms WiFi UDP FCFS in terms of information freshness, as we see next.

**Queueing discipline.** The results in Tables 5.2 and 5.3 show that WiFresh RT and

WiFresh App can improve AoI by two orders of magnitude when compared to WiFi UDP FCFS. In this single source scenario, the performance gain comes from using a LCFS queue instead of a FCFS queue. In Fig. 5-10, we compare the AoI $h_i(t)$ evolution over time for a system employing FCFS and LCFS. It is clear that *a high packet generation rate improves the AoI $h_i(t)$ performance of LCFS, and degrades the performance of FCFS.*

**Queue size.** In all three WiFi UDP FCFS experiments in Table 5.3, the AoI $h_i(t)$ grows as in Fig. 5-10 throughout the entire experiment, i.e. for 600 seconds, giving a time-average AoI of at least 220 seconds. This result suggests that the FCFS queue of the Raspberry Pi did not overflow, which would have helped stabilizing AoI. In contrast, in all three WiFi UDP FCFS experiments in Table 5.2, the FCFS queue[5] overflows in the first few seconds, limiting the AoI $h_i(t)$ growth and resulting in a time-average AoI of around 0.3 seconds. This suggests that a larger FCFS queue at the SDRs would have further impaired the performance of WiFi UDP FCFS.

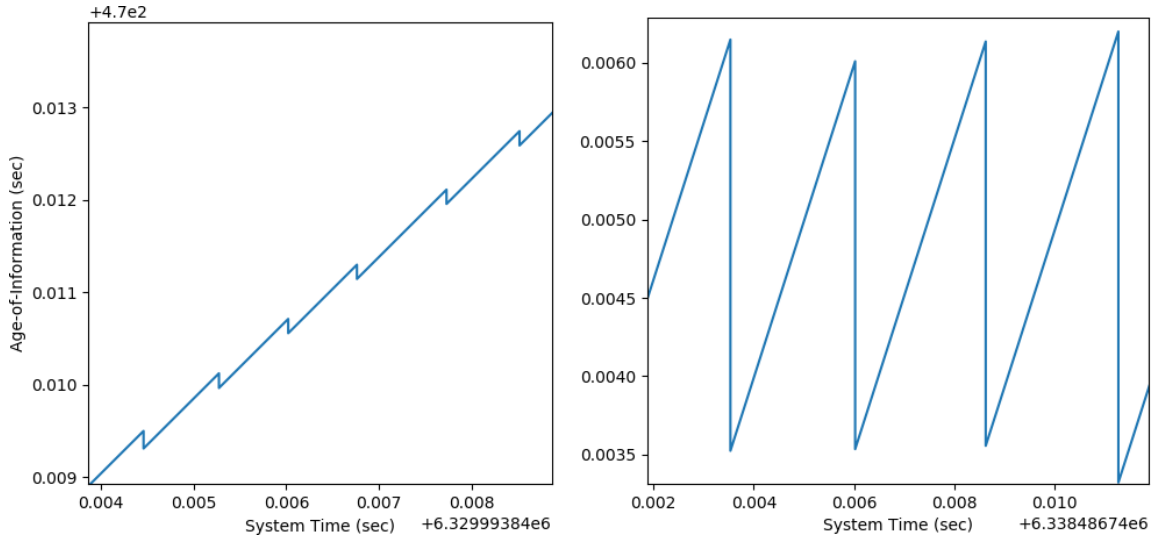



Figure 5-10: AoI $h_i(t)$ evolution over time in the Raspi testbed with $\lambda = 6$ kHz. On the LHS we have WiFi UDP FCFS and on the RHS we have WiFresh App, which uses LCFS.

[5]The transmission queue of the SDR can store one megabyte of data. Notice that for $\lambda = 5$kHz we are generating 0.75 megabyte per second.

### 5.4.2 Network with Increasing Load

In this section, we consider a network with a destination, a wireless BS and *ten sources* generating packets of 150 bytes with rate $\lambda$. In Fig. 5-11, we display the Expected Weighted Sum AoI measurements (in seconds) for the SDR testbed employing the following communication systems: 1) WiFresh RT; 2) WiFi UDP LCFS; 3) WiFresh RT FCFS; and 4) WiFi UDP FCFS. The weights of the ten sources are set to $w_i = 1, \forall i$. Each experiment runs for 10 minutes.

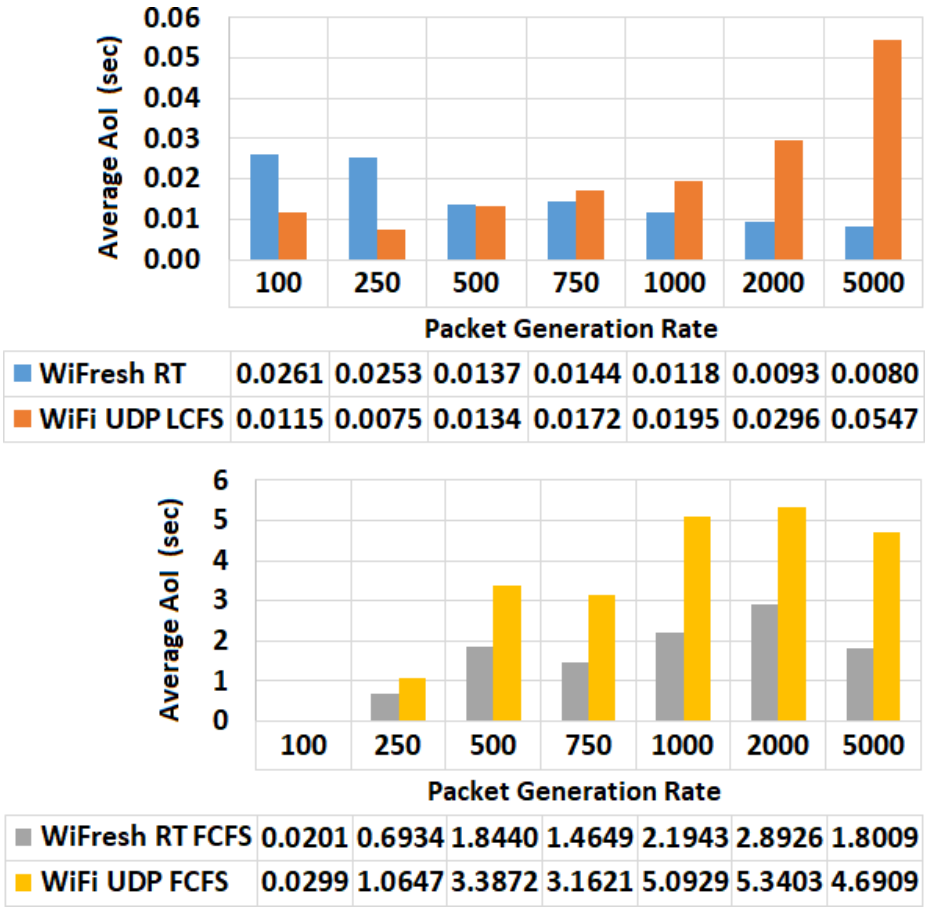



Figure 5-11: Time-average AoI measurements for the SDR testbed with ten sources generating packets of 150 bytes with rate $\lambda \in \{100, 250, 500, 750, 1k, 2k, 5k\}$ Hz.

By comparing the results of WiFresh RT and WiFi UDP FCFS for $\lambda \geq 500$ Hz, we can see that *WiFresh RT improves information freshness by (at least) a factor of* **200** *when compared to an equivalent standard WiFi network*. To understand how much of this im-

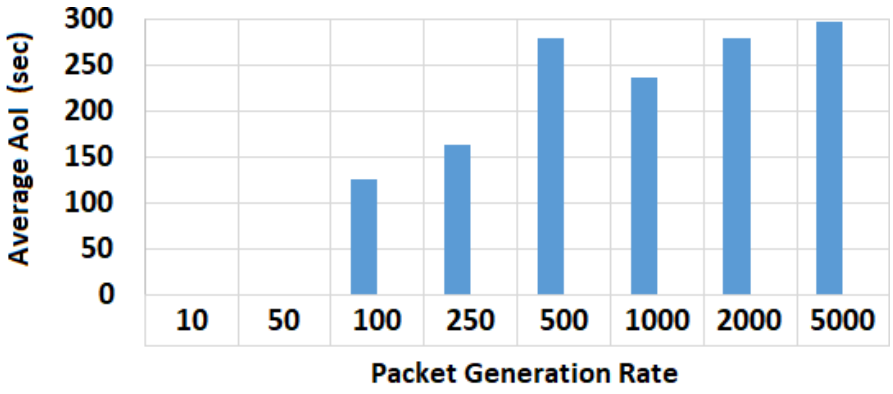
provement is due to the queueing discipline and how much is due to the multiple access mechanism, we draw additional comparisons. By comparing WiFresh RT and WiFi UDP LCFS, both of which use LCFS queues, we can assess the impact of the multiple access mechanism on information freshness. As expected, the improvement of Polling over Random Access increases as the packet generation rate $\lambda$ increases. In particular, for $\lambda = 5$ kHz, WiFresh RT improves AoI by a factor of 7 when compared to WiFi UDP LCFS. To assess the impact of queueing, we compare WiFresh RT and WiFresh RT FCFS, both of which use Polling with MW policy. For $\lambda \geq 500$ Hz, the LCFS queue improves information freshness by (at least) a factor of 100 when compared to the FCFS queue. *Both the queueing discipline and the multiple access mechanism improve AoI significantly*. However, the effect of the queueing discipline is clearly dominant.

In Fig. 5-12, we display the EWSAoI measurements (in seconds) for the Raspi testbed employing the following communication systems: 1) WiFresh App; and 2) WiFi UDP FCFS. The weights of the ten sources are set to $w_i = 1, \forall i$. Each experiment runs for 10 minutes. The results in Fig. 5-12 show that for $\lambda \geq 100$ Hz, *WiFresh App improves information freshness by three orders of magnitude when compared to an equivalent standard WiFi network*. We note that WiFi UDP FCFS performs differently in the Raspi and SDR testbeds due to differences in the platforms, and in particular due to *differences in the FCFS queue sizes*. The large FCFS queues at the Raspberry Pis have a negative effect on WiFi UDP FCFS, which amplifies the performance gain of WiFresh App at high packet generation rates $\lambda$.

### 5.4.3 Network with Increasing Size

In this section, we consider a network with a destination, a wireless BS and $N$ sources, each source generates up to three types of information updates: position information of 50 bytes at 1 Hz, inertial measurements of 20 bytes at 100 Hz, and images of 19 kbytes at 2 Hz. Notice that a network with $N$ physical sources can have up to $3N$ sources of information, each source of information with its own independent instance of WiFresh App, its own queue, and its own AoI $h_i(t)$ evolution.

In Figs. 5-13 and 5-14, we display the EWSAoI measurements (in seconds) for the




| | 10 | 50 | 100 | 250 | 500 | 1000 | 2000 | 5000 |
|---|---|---|---|---|---|---|---|---|
| Wifi UDP FCFS | 0.065 | 0.021 | 126.4 | 163.1 | 279.4 | 236.3 | 278.4 | 296.7 |
| WiFresh App | 0.132 | 0.076 | 0.076 | 0.069 | 0.068 | 0.093 | 0.074 | 0.088 |

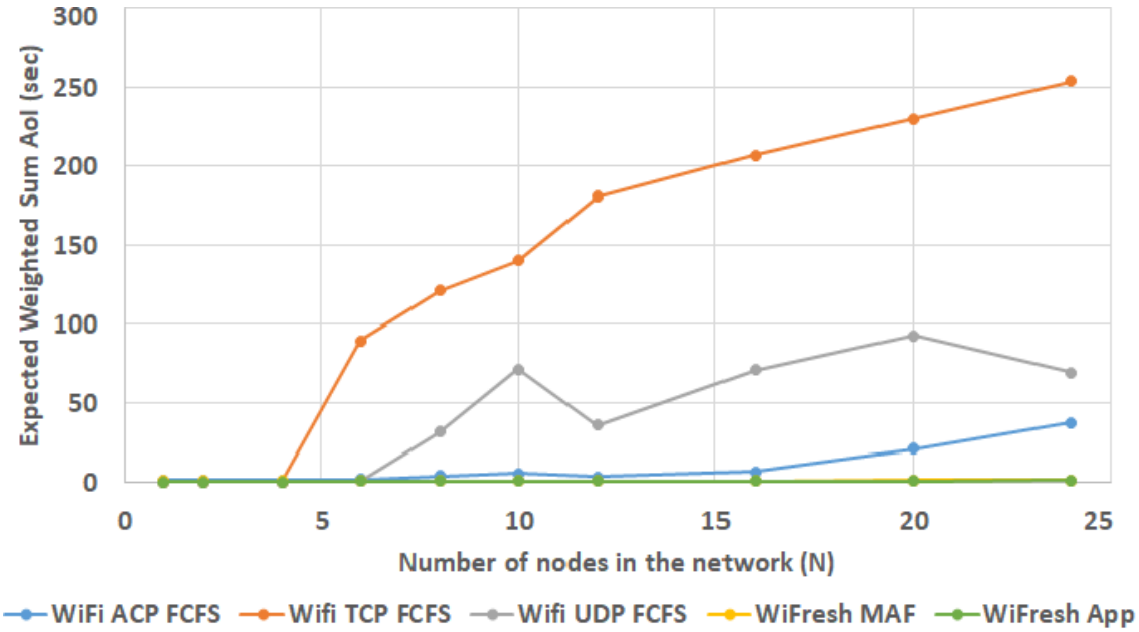
Figure 5-12: Time-average AoI measurements for the Raspi testbed with ten sources generating packets of 150 bytes with rate $\lambda \in \{10, 50, 100, 250, 500, 750, 1k, 2k, 5k\}$ Hz.

Raspi testbed employing the following communication systems: 1) WiFresh App; 2) WiFresh MAF; 3) WiFi UDP FCFS; 4) WiFi TCP FCFS; and 5) WiFi ACP FCFS. In Fig. 5-13, we consider sources generating both position information and images, and in Fig. 5-14, we consider sources generating both position information and inertial measurements. The weights of the sources of information are set to $w_i = 1, \forall i$. Each configuration runs for a total of 10 minutes.

**TCP over standard WiFi.** The results in Figs. 5-13 and 5-14 show that WiFi TCP FCFS has the worst performance in terms of AoI. TCP provides reliable and in-order packet delivery by requesting retransmissions and rearranging out-of-order packets before forwarding them to the Application layer. Both of these features can degrade information freshness, especially when sources are generating packets at high rates.

**Age Control Protocol over standard WiFi.** ACP dynamically adapts the packet generation rates at the sources (by regularly discarding some of the packets) in order to drive the underlying WiFi network to the point of minimum AoI. The results in Figs. 5-13 and 5-14 show that, for $N = 20$, WiFi ACP FCFS improves information freshness by a factor of 4 when compared to WiFi UDP FCFS; in turn WiFresh App improves information freshness by (at least) a factor of 10 when compared to WiFi ACP FCFS.

**Impact of scheduling policy.** The only difference between WiFresh App and WiFresh

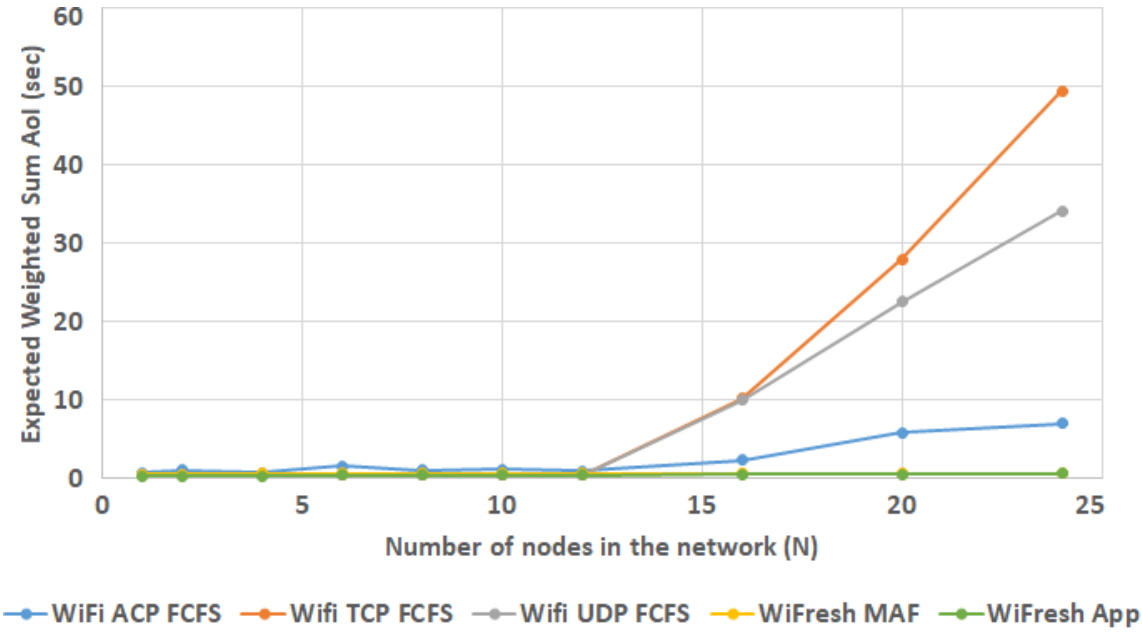



| Nodes (N) | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 16 | 20 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|
| Wifi TCP FCFS | 0.45 | 0.45 | 0.47 | 89.65 | 120.89 | 139.79 | 180.33 | 206.12 | 229.48 | 253.66 |
| Wifi UDP FCFS | 0.45 | 0.45 | 0.45 | 0.77 | 31.71 | 71.43 | 35.90 | 70.92 | 92.91 | 69.43 |
| Wifi ACP FCFS | 1.15 | 1.11 | 1.04 | 1.97 | 4.17 | 5.46 | 3.36 | 6.81 | 20.83 | 37.56 |
| WiFresh MAF | 0.55 | 0.56 | 0.58 | 0.65 | 0.67 | 0.67 | 0.72 | 0.91 | 1.18 | 1.36 |
| WiFresh App | 0.44 | 0.44 | 0.47 | 0.52 | 0.60 | 0.60 | 0.69 | 0.83 | 0.98 | 1.06 |

Figure 5-13: EWSAoI measurements for the Raspberry Pi testbed with $N \in \{1,2,4,6,8,10,12,16,20,24\}$ sources generating position information and images.

MAF is the scheduling policy. MAF schedules the source with highest value of $\hat{h}_i(t)$, and neglects information about channel conditions $\hat{p}_i(t)$ and information about the HoL packet at the source's queue $\hat{z}_i(t)$. In networks with sources that generate data packets at different rates, such as the Raspi testbed, MAF can often poll sources with empty queues, what degrades its AoI performance. This is a main reason for the performance gap between WiFresh MAF and WiFresh App in Figs. 5-13 and 5-14.

**Impact of traffic.** The results in Fig. 5-13 show that for $N \geq 16$, WiFresh App improves information freshness by a factor of 65 when compared to WiFi UDP FCFS, and by a factor of 230 when compared to WiFi TCP FCFS. The results in Fig. 5-14 show that for $N \geq 16$, WiFresh App improves information freshness by a factor of 20 when compared to either WiFi UDP FCFS or WiFi TCP FCFS. The improvement is more evident in Fig. 5-13 since cameras generate more traffic than IMUs. In particular, the camera generates approximately




| Nodes (N) | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 16 | 20 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|
| Wifi TCP FCFS | 0.34 | 0.34 | 0.34 | 0.34 | 0.32 | 0.33 | 0.32 | 10.17 | 27.91 | 49.31 |
| Wifi UDP FCFS | 0.34 | 0.34 | 0.34 | 0.34 | 0.32 | 0.33 | 0.32 | 9.96 | 22.43 | 34.09 |
| Wifi ACP FCFS | 0.66 | 0.99 | 0.74 | 1.51 | 0.97 | 1.12 | 0.88 | 2.23 | 5.76 | 6.91 |
| WiFresh MAF | 0.52 | 0.52 | 0.53 | 0.55 | 0.54 | 0.55 | 0.55 | 0.56 | 0.58 | 0.58 |
| WiFresh App | 0.29 | 0.29 | 0.31 | 0.33 | 0.35 | 0.36 | 0.39 | 0.43 | 0.49 | 0.54 |

Figure 5-14: EWSAoI measurements for the Raspberry Pi testbed with $N \in \{1,2,4,6,8,10,12,16,20,24\}$ sources generating position information and inertial measurements.

304 kbits per second per source while the IMU generates approximately 16 kbits per second per source.

From the measurements in this section, it is clear that *the more congested the network, the more prominent is the superiority of WiFresh when compared with WiFi in terms of information freshness, making WiFresh well-suited for large-scale applications that rely on sharing large amounts of time-sensitive information.* The average AoI in a WiFresh network scales gracefully with the packet generation rate $\lambda$, as seen in Sec. 5.4.2, and with the number of nodes $N$, as seen in Sec. 5.4.3. WiFresh Real-Time achieves the highest performance in terms of throughput and average AoI, while WiFresh App achieves high performance and can be easily integrated into time-sensitive applications that already run over WiFi, as discussed in Sec. 5.3.4.

## 5.5 Summary

In this chapter, we proposed WiFresh: an unconventional architecture that achieves near optimal information freshness for wireless networks of any size. The superior performance of WiFresh is due to the combination of three elements: LCFS queues, Polling Multiple Access mechanism, and Max-Weight scheduling policy. The choice of each of these elements is underpinned by theoretical research. We proposed and realized two strategies for implementing WiFresh: 1) WiFresh Real-Time, in which our architecture is implemented at the MAC layer in a network of eleven FPGA-based Software Defined Radios using hardware-level programming; and 2) WiFresh App which is a customization of WiFresh implemented at the Application layer, without modifications to lower layers of the communication system, in a network of twenty five Raspberry Pis using Python 3. A key advantage of WiFresh App is that it can be easily integrated into time-sensitive applications that already run over WiFi such as [2–4, 16, 27, 56, 63, 69, 75, 77, 79, 80, 83, 88, 89, 106–108, 111]. Our experimental results showed that WiFresh can improve information freshness by two orders of magnitude when compared to an equivalent standard WiFi network.

# Appendices

## 5.A Synchronization for WiFresh App

To synchronize the clocks of the sources with the destination, we implement a two-way handshake called the on-wire protocol that is part of the Network Time Protocol [82, §8]. Let $t^D$ be the current time measured by the destination and let $t_i^S$ be the current time measured by source $i$. The time-stamp offset is given by $\Delta_i = t^D - t_i^S$. To estimate the value of $\Delta_i$ for a given source $i$ we implement a handshake with five basic steps:

1. The destination starts the handshake by measuring the current time $t_1^D$ and then immediately transmitting a *sync request packet* to source $i$ containing the value of $t_1^D$.

2. Source $i$ receives the sync request and records the reception time $t_{i,2}^S$. Notice that $t_{i,2}^S - t_1^D = \delta^{D\to i} - \Delta_i$, where $\delta^{D\to i}$ is the time elapsed since the start of the transmission at the destination until the packet reception at the source.

3. Source $i$ measures the current time $t_{i,3}^S$ and then immediately transmits a *sync response packet* to the destination containing the values of $t_1^D$, $t_{i,2}^S$ and $t_{i,3}^S$.

4. The destination receives the sync response and records the reception time $t_4^D$. Notice that $t_4^D - t_{i,3}^S = \delta^{i\to D} + \Delta_i$, where $\delta^{i\to D}$ is the time elapsed since the start of the transmission at the source until the packet reception at the destination.

5. Assuming that $\delta^{D\to i} \approx \delta^{i\to D}$, an estimate of the offset $\Delta_i$ is given by the expression

$$\tilde{\Delta}_i(t) = \frac{(t_4^D - t_{i,3}^S) - (t_{i,2}^S - t_1^D)}{2}\,. \tag{5.3}$$

To synchronize the network, the destination performs a two-way handshake with each source $i \in \{1,2,\cdots,N\}$. After each successful handshake, we use Eq.5.3 to obtain a new offset measurement $\tilde{\Delta}_i(t)$. The entire synchronization procedure runs periodically with period *sync_period* = 300 seconds. The sequence of past and present time-stamp offset measurements for each source $i$ is filtered and estimates of $\{\hat{\Delta}_i(t)\}_{i=1}^N$ are obtained. The

AoI with time-stamp offset correction is given by $\hat{h}_i(t) = t - \tau_i^S(t) - \hat{\Delta}_i(t)$. As soon as the synchronization procedure ends, the destination resumes using the MW policy with $\hat{h}_i(t)$ to schedule data packet transmissions.

Chapter 6

# AoI in Random Access Networks

Random Access is a multiple access technique that underpins protocols such as *Slotted-ALOHA* and *Carrier-Sense Multiple Access* (CSMA). A main difference between the two protocols is that CSMA utilizes carrier sensing capabilities to avoid packet collisions, while Slotted-ALOHA is a simpler protocol that does not assume that nodes have carrier sensing capabilities.

In chapter 5, we provided numerous examples of time-sensitive applications that are implemented using Random Access [2–4, 16, 27, 56, 63, 69, 75, 77, 79, 80, 83, 88, 89, 106–108, 111], discussed two important shortcomings of networks that employ FCFS queues and Random Access, namely scalability and congestion, and then proposed an alternative network architecture composed of LCFS queues, Polling mechanism and Max-Weight scheduling policy. In contrast, in this chapter, we consider networks that employ LCFS queues and Random Access, in particular Slotted-ALOHA and CSMA, and propose a framework to analyze and optimize information freshness.

The literature on the analysis and optimization of Slotted-ALOHA and CSMA networks is vast, dating almost five decades [1, 64, 91]. For a survey on throughput and delay optimization of CSMA networks, we refer the readers to [117]. The optimization of *centralized* multiple access mechanisms in terms of AoI has been considered in previous chapters and in numerous works including [32, 38, 42, 49, 51, 74, 99, 101, 104, 105, 114]. The optimization of *distributed* mechanisms, such as Slotted-ALOHA and CSMA, in terms of AoI has been

recently considered in [18, 20, 26, 41, 57, 58, 67, 76, 103].

The authors of [18, 58, 103] studied Slotted-ALOHA networks with sources that generate packets on demand. Slotted-ALOHA networks with stochastic packet generation were considered in [20, 67]. In particular, the authors of [20] analyzed Slotted-ALOHA networks in the limit as the number of sources goes to infinity, and proposed a mechanism to dynamically drop packets in order to minimize the average AoI in the network. The authors of [67] used Queueing Theory to analyze AoI in a wireless network with Bernoulli packet arrivals and geometric inter-delivery times, and used simulation results to optimize the AoI performance under three classes of multiple access mechanisms, including Slotted-ALOHA.

CSMA networks were considered in [26, 41, 57, 76]. In particular, the authors of [57] used simulations and experimental results to evaluate the average AoI in a CSMA network. The authors of [26] developed a discrete-time model for a CSMA network with sources that generate packets on demand, derived an expression for the average AoI, and then used Game Theory to analyze the coexistence of WiFi and Dedicated Short-Range Communications (DSRC) in terms of throughput and AoI. The authors of [76] developed a continuous-time model for a collision-free CSMA network with stochastic packet generation, derived an expression for the average AoI, and then used this expression to find the optimal back-off rate. Notice that [76] does not consider the effects of packet collisions which, as we see in this chapter, play an important role in the AoI optimization of Random Access networks.

**Our contributions.** In this chapter, we propose a framework to analyze and optimize the average AoI in Random Access networks with stochastic packet generation. In particular, we develop a discrete-time network model that accounts for the effects of packet collisions and derive an accurate approximation for the average AoI in the network. We then use the analytical model to optimize the Random Access mechanism in terms of AoI. Our approach allows us to evaluate the combined impact of the packet generation rate, transmission probability, and size of the network on the AoI performance. Finally, we implement the optimized CSMA network in the Software Defined Radio testbed in Fig. 6-8 and compare the AoI measurements with analytical results and simulations. *To the best of our knowledge, this is the first work to provide theoretical results on the optimization of a*

*CSMA network with stochastic packet generation and packet collisions, and the first work to implement a CSMA mechanism optimized for AoI.*

The remainder of this chapter is organized as follows. In Sec. 6.1, we present the network model. In Sec. 6.2, we derive expressions for the inter-delivery interval, packet delay and AoI. In Sec. 6.3, we optimize the Random Access network in terms of AoI. In Sec. 6.4, we implement the optimized CSMA network and discuss experimental results. This chapter is concluded in Sec. 6.5.

## 6.1 System Model

Consider a broadcast single-hop wireless network with $N$ sources transmitting time-sensitive information to the Base Station (BS) using Random Access. Let the time be slotted, with *mini-slot* duration $\delta$ seconds and mini-slot index $k \in \{1,2,\cdots,K\}$, where $K\delta$ is the time-horizon of this discrete-time system. At the beginning of every mini-slot $k$, each source $i \in \{1,2,\cdots,N\}$ generates a new packet with probability $\lambda_i \in (0,1]$. Let $a_i(k)$ be the indicator function that is equal to 1 when source $i$ generates a fresh packet in mini-slot $k$, and equal to 0, otherwise. The packet generation process is i.i.d. over mini-slots and independent across different sources, with $\mathbb{P}(a_i(k)=1)=\lambda_i$.

**Queueing discipline.** *Sources keep only the most recently generated packet, i.e. the freshest packet, in their transmission queue.* When source $i$ generates a new packet at the beginning of mini-slot $k$, older packets are discarded from its transmission queue. This queueing discipline is equivalent to LCFS and it is known to optimize the AoI in a variety of contexts [10,21,60]. Notice that delivering the most recently generated packet provides the freshest information to the BS. Moreover, notice that when a packet with *older information* is delivered after a packet with *fresher information*, the information freshness at the BS is not affected. Hence, discarding *older packets* from the transmission queue when a *fresher packet* is generated has no effect on the information freshness.

**Random Access mechanism.** When there is a transmission opportunity in mini-slot $k$ and source $i$ has an undelivered packet, then source $i$ starts transmitting with probability $\mu_i \in (0,1]$, and idles with probability $1-\mu_i$. The mini-slot duration $\delta$ is set to the time

needed for any source to detect a transmission from other sources. Hence, if source $i$ is the only source to start transmitting in mini-slot $k$, then all other sources detect the transmission by the beginning of mini-slot $k+1$ and defer new transmissions until source $i$ stops transmitting. As a result, if source $i$ is the only source to start transmitting in mini-slot $k$, then this transmission is successful. Otherwise, if two or more sources start transmitting in the same mini-slot $k$, then there is a packet collision and the BS is unable to receive these packets. After the collision, sources continue to employ Random Access to retransmit their undelivered packets. The duration of a collision or a successful packet transmission is $L$ mini-slots, as illustrated in Fig. 6-1. We assume that there are no hidden/exposed sources and that the feedback from the BS is instantaneous and without error.

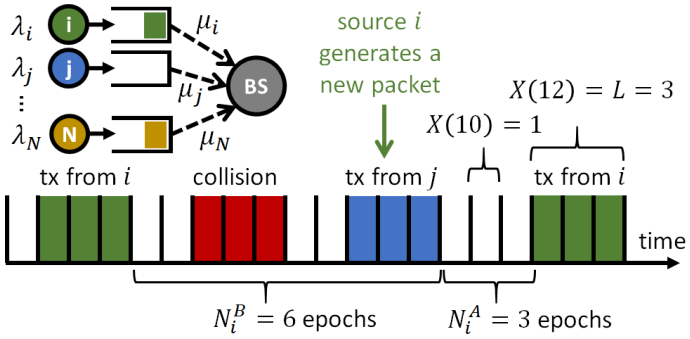



Figure 6-1: Illustration of the Random Access network and associated timeline with packet generation, transmission and collision events.

**Definition of epoch.** Sources continually sense the wireless channel and defer transmissions until the channel is idle. Transmission opportunities occur when the channel is idle at the beginning of a mini-slot. Denote by *epoch* the time interval between two consecutive transmission opportunities, let $t \in \{1,2,\cdots,T\}$ be the *epoch index*, where $T$ is the total number of epochs, and let $X(t)$ be the number of mini-slots contained in epoch $t$. It follows that $X(t) = L$ mini-slots when epoch $t$ is busy, i.e. contains a transmission attempt, and $X(t) = 1$ mini-slot when epoch $t$ is idle. By dividing the time-horizon into epochs, we obtain $K = \sum_{t=1}^{T} X(t)$. In Fig. 6-1, we have $K = 20$ mini-slots and $T = 12$ epochs. Notice that each epoch is associated with a *single* transmission opportunity.

### 6.1.1 Transmission probability

Source $i$ can transmit at the beginning of every *epoch* in which its transmission queue has a packet. Let $q_i(t)$ be the probability of source $i$ transmitting a packet in epoch $t$. Naturally, if the transmission queue is empty, then $q_i(t) = 0$, and if the transmission queue has a packet, then $q_i(t) = \mu_i$. It follows that the probability of epoch $t$ being idle, containing a successful packet transmission from source $i$, and containing a packet collision are given by

$$\mathbb{P}^I(t) = \prod_{i=1}^{N}(1 - q_i(t)) \tag{6.1a}$$

$$\mathbb{P}_i^S(t) = q_i(t) \prod_{j=1, j\neq i}^{N} (1 - q_j(t)) \tag{6.1b}$$

$$\mathbb{P}^C(t) = 1 - \mathbb{P}^I(t) - \sum_{i=1}^{N} \mathbb{P}_i^S(t) \,, \tag{6.1c}$$

respectively. Notice that $\mathbb{P}^I(t)$, $\mathbb{P}_i^S(t)$, and $\mathbb{P}^C(t)$ depend on state of the transmission queues of every source in the network. For simplicity, and since we do not assume global knowledge of the state of the queues, we approximate the transmission probability in epoch $t$, $q_i(t)$, by its expected time-average $q_i = \lim_{T\to\infty} \sum_{t=1}^{T} \mathbb{E}[q_i(t)]/T$. Equivalent approximations are employed in various works that analyze Random Access networks, such as [14, 17, 18, 26]. In Secs. 6.2 and 6.4, we compare the analytical model with simulation and experimental results, and validate this approximation.

To obtain a closed-form expression for the *transmission probability* $q_i$, we consider the time interval between two consecutive packet deliveries from source $i$, and divide this interval into two parts: before and after a new packet generation. Let $N_i^B$ be the number of consecutive *epochs* following the start of the inter-delivery interval (i.e., after the successful packet transmission) and preceding the packet generation, and let $N_i^A$ be the number of consecutive *epochs* following the packet generation and preceding the actual packet delivery, as illustrated in Fig. 6-1. Let $X_i^B(t)$ be the number of mini-slots contained in epoch $t \in \{1, \cdots, N_i^B\}$ within the interval $N_i^B$, and let $X_i^A(t)$ be the number of mini-slots contained in epoch $t \in \{1, \cdots, N_i^A\}$ within the interval $N_i^A$. The sequence of packet deliveries from source $i$ is a renewal process and, thus, we can employ the elementary renewal

theorem [23, Sec. 5.6] to obtain the following expression for the transmission probability

$$q_i = \frac{(\mathbb{E}[N_i^A]+1)\mu_i}{\mathbb{E}[N_i^B]+(\mathbb{E}[N_i^A]+1)}\,,\forall i\,. \tag{6.2}$$

Notice that the denominator in (6.2) represents the expected number of *transmission opportunities* in the inter-delivery interval and the numerator in (6.2) represents the expected number of *transmission opportunities* in which source $i$ has a packet to transmit. The expected number of *mini-slots* in the inter-delivery interval is discussed in Sec. 6.2.1.

We define the probability that all nodes *other than* $i$ are idle during an arbitrary epoch $t$ as

$$Q^{-i} = \prod_{j=1, j\neq i}^{N} (1-q_j)\,. \tag{6.3}$$

**Proposition 6.1.** *The expected time-average transmission probability of source $i$ is given by*

$$q_i = \left(\frac{(1-\lambda_i)^L Q^{-i}}{1-(1-\lambda_i)Q^{-i}-(1-\lambda_i)^L(1-Q^{-i})}+\frac{1}{\mu_i}\right)^{-1}. \tag{6.4}$$

*Proof.* To obtain the transmission probability in (6.4), we start by deriving expressions for $\mathbb{E}[X_i^B]$ and $\mathbb{E}[X_i^A]$. Then, we use these expected values, the law of iterated expectations, and the fact that $X_i^B(t)$ are i.i.d. over time to derive expressions for $\mathbb{E}[N_i^B]$ and $\mathbb{E}[N_i^A]$. Finally, we substitute $\mathbb{E}[N_i^B]$ and $\mathbb{E}[N_i^A]$ into (6.2) to obtain (6.4). The complete proof is provided in Appendix 6.A. ■

Notice from (6.4) that, as expected, $q_i \in (0,\mu_i]$. Moreover, notice that changing the packet generation probability $\lambda_i$ or the conditional transmission probability $\mu_i$ of a particular source $i$, changes the transmission probability $q_j$ of *all sources in the network*. In particular, changing $\lambda_i$ or $\mu_i$, changes $q_i$ according to (6.4). In turn, $q_i$ affects $Q^{-j}, \forall j \neq i$, which affects the transmission probabilities $q_j$ of all sources in the network. The set of functions $\{q_i\}_{i=1}^N$ captures the influence that one source has on other sources in the network. This set of functions is further discussed in Secs. 6.2 and 6.3. Next, we develop a framework for analyzing information freshness.

## 6.2 Analysis of Age of Information

In this section, we derive expressions for the inter-delivery interval, packet delay and AoI, and then, compare the analysis with simulation results. In Sec. 6.3, we use this framework to optimize Slotted-ALOHA and CSMA networks in terms of AoI.

### 6.2.1 Inter-delivery interval

The sequence of packet deliveries from source $i$ is a renewal process. For this reason, henceforth in this section, we focus on a single inter-delivery interval. Consider the inter-delivery interval in Fig. 6-1. Let $I_i$ be the number of *mini-slots* between two consecutive packet deliveries from source $i$. It follows that

$$\mathbb{E}[I_i] = \mathbb{E}\left[\sum_{t=1}^{N_i^B} X_i^B(t) + \sum_{t=1}^{N_i^A} X_i^A(t) + L\right], \tag{6.5}$$

where the first and second sums on the RHS of (6.5) represent the total number of mini-slots in the time intervals $N_i^B$ and $N_i^A$, respectively.

**Proposition 6.2.** *A (tight) lower bound on the expected inter-delivery interval is given by*

$$\mathbb{E}[I_i] \geq \frac{(1-\lambda_i)^L}{\lambda_i} + \left(L\,\frac{1-Q^{-i}}{Q^{-i}} + 1\right)\frac{1}{\mu_i} + L - 1\,, \tag{6.6}$$

*Proof.* To obtain the lower bound in (6.6), we derive expressions for the first and second sums on the RHS of (6.5). The expected value of the second sum $\sum_{t=1}^{N_i^A} X_i^A(t)$ can be obtained by employing Wald's equality [23, Sec. 5.5] and then using $\mathbb{E}\left[N_i^A\right]$ and $\mathbb{E}\left[X_i^A(t)\right]$

derived in Appendix 6.A, as follows

$$\mathbb{E}\left[\sum_{t=1}^{N_i^A} X_i^A(t)\right] = \mathbb{E}\left[N_i^A\right]\mathbb{E}\left[X_i^A(t)\right] = \frac{1-\mu_i Q^{-i}}{\mu_i Q^{-i}}\left[\frac{(1-\mu_i)Q^{-i}}{1-\mu_i Q^{-i}} + L\,\frac{1-Q^{-i}}{1-\mu_i Q^{-i}}\right] . \tag{6.7}$$

Notice that we cannot employ Wald's equality to find an expression for the first sum $\sum_{t=1}^{N_i^B} X_i^B(t)$. This is because the random variables $X_i^B(t)$ and $N_i^B$ are dependent. Recall that packet generation events occur at the beginning of *mini-slots*, as opposed to *epochs*. Hence, if the epoch duration $X_i^B(t)$ increases, the number of epochs until the first packet generation $N_i^B$ decreases. To obtain an approximate expression for the first sum, we use the lower bound $Y_i^B \leq \sum_{t=1}^{N_i^B} X_i^B(t)$, where $Y_i^B$ is the number of mini-slots that precede the first packet generation. The probability distribution of $Y_i^B$ is given by

$$\mathbb{P}(Y_i^B = 0) = 1-(1-\lambda_i)^L ; \tag{6.8a}$$

$$\mathbb{P}(Y_i^B = k) = (1-\lambda_i)^{L+k-1}\lambda_i, \forall k \in \{1,2,\cdots\} , \tag{6.8b}$$

where (6.8a) represents the probability of source $i$ generating a new packet while delivering the previous packet or in the first mini-slot after the delivery, and (6.8b) represent the probability of source $i$ generating a new packet $k+1$ mini-slots after delivering the previous packet. From (6.8a) and (6.8b) we obtain the lower bound

$$\mathbb{E}[Y_i^B] = \frac{(1-\lambda_i)^L}{\lambda_i} \leq \mathbb{E}\left[\sum_{t=1}^{N_i^B} X_i^B(t)\right] . \tag{6.9}$$

Substituting (6.7) and (6.9) into the inter-delivery interval in (6.5) gives (6.6). ■

Notice that if the packet generation occurs during a busy epoch, as illustrated in Fig. 6-1, then $Y_i^B \leq \sum_{t=1}^{N_i^B} X_i^B(t) \leq Y_i^B + L - 1$. Otherwise, if the packet generation occurs during an idle epoch, then $Y_i^B = \sum_{t=1}^{N_i^B} X_i^B(t)$. The approximation $\mathbb{E}[Y_i^B] \approx \mathbb{E}[\sum_{t=1}^{N_i^B} X_i^B(t)]$ simplifies the analysis and is particularly accurate in networks with small value of $L$ and/or low transmission probabilities $q_i$. In Slotted-ALOHA networks, in which $L=1$, we have

$\mathbb{E}[Y_i^B] = \mathbb{E}[\sum_{t=1}^{N_i^B} X_i^B(t)]$ and the lower bound on the inter-delivery interval in (6.6) holds with *equality*. Numerical results in Sec. 6.2.3 show that the lower bound in (6.6) is tight in a wide range of network configurations, including CSMA networks with large $L$, and is particularly accurate near the point of optimal AoI. Next, we derive an analytical expression for the Age of Information.

### 6.2.2 Age of Information

Consider the inter-delivery interval in Fig. 6-2. Let $\tau_i(k)$ be the time-stamp of the *freshest* packet received by the destination from source $i$ at the beginning of mini-slot $k$. Then, the AoI is defined as $h_i(k) := k - \tau_i(k)$. At the beginning of the mini-slot that follows a packet delivery from source $i$, the value of $\tau_i(k)$ is updated to the time-stamp of the new packet, and the AoI is reduced to the *packet delay*, namely $h_i(k) = z_i = k - \tau_i(k)$, where $z_i$ is the delay associated with the freshest packet delivered from source $i$. The evolution of AoI and its relationship with the packet delay are illustrated in Fig. 6-2.

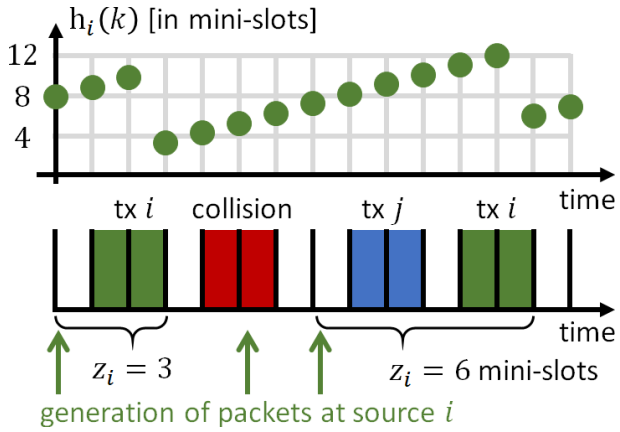



Figure 6-2: Timeline with packet generation, transmission and collision events, and associated packet delay $z_i$ and AoI evolution $h_i(k)$.

To capture the Age of Information in the entire network, we define the infinite-horizon expected network AoI (NAoI) as

$$NAoI := \lim_{K\to\infty} \frac{1}{KN} \sum_{k=1}^{K} \sum_{i=1}^{N} \mathbb{E}[h_i(k)] \ . \tag{6.10}$$

**Theorem 6.3.** *The expected NAoI is (accurately) approximated by*

$$NAoI \approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1-\lambda_i}{\lambda_i}+\left(L\,\frac{1-Q^{-i}}{Q^{-i}}+1\right)\frac{1}{\mu_i}\right)+\frac{3(L-1)}{2}+ \\ +\frac{1}{2N}\sum_{i=1}^{N}\frac{\frac{(1-\lambda_i)^L}{\lambda_i}\left[\frac{2}{\lambda_i}+L-1\right]-(L-1)\left[\frac{1}{\mu_i}-1\right]}{\left(\frac{(1-\lambda_i)^L}{\lambda_i}+\left(L\,\frac{1-Q^{-i}}{Q^{-i}}+1\right)\frac{1}{\mu_i}+L-1\right)}. \qquad (6.11)$$

*Proof.* To obtain an expression for the infinite-horizon expected network AoI in (6.10), we first analyze the evolution of $h_i(k)$ over time, and then we employ tools from Renewal Theory. From Fig. 6-2, we can see that in an inter-delivery interval with duration $I_i$ mini-slots and packet delay $z_i$ mini-slots, the value of $h_i(k)$ evolves according to the sequence $z_i, z_i+1, \cdots, z_i+I_i-1$. The sum of AoI in this inter-delivery interval is $z_iI_i+I_i(I_i-1)/2$.

The sequence of packet deliveries over time is a renewal process. This fact allowed us to use Renewal Theory to obtain the expression for the transmission probability $q_i$ in (6.2). However, since the packet delay is not independent across consecutive inter-delivery intervals - notice from Fig. 6-2 that the value of the packet delay is upper bounded by the previous inter-delivery interval - we cannot use Renewal Theory to obtain an expression for NAoI.

To overcome this challenge, we define the *augmented packet delay* $\tilde{z}_i \in \{L, L+1, \cdots\}$, which is unbounded and independent across inter-delivery intervals. The augmented packet delay is an upper bound on the packet delay, namely $z_i \leq \tilde{z}_i$, with probability distribution $\mathbb{P}(\tilde{z}_i = L+k) = (1-\lambda_i)^k\lambda_i, k \in \{0,1,\cdots\}$. This upper bound is particularly tight when the inter-delivery intervals $I_i$ are large and/or the packet generation probability $\lambda_i$ is high. Using the elementary renewal theorem for renewal-reward processes [23, Sec. 5.4] and the

augmented packet delay $\tilde{z}_i$ into (6.10), we get

$$\lim_{K\to\infty} \frac{1}{KN} \sum_{k=1}^{K} \sum_{i=1}^{N} \mathbb{E}\left[h_i(k)\right] \leq \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbb{E}[\tilde{z}_i I_i + I_i(I_i-1)/2]}{\mathbb{E}[I_i]}$$
$$= \mathbb{E}[\tilde{z}_i] + \frac{\mathbb{E}[I_i^2]}{2\mathbb{E}[I_i]} - \frac{1}{2}\,. \tag{6.12}$$

Substituting the first and second[1] moments of the inter-delivery interval, and the first moment of the augmented packet delay into (6.12), we obtain the approximated expression for NAoI in (6.11). ■

### 6.2.3 Numerical Results

In this section, we consider *symmetric* Random Access networks with $N = 10$ sources, packet generation probabilities $\lambda_i = \lambda, \forall i$, and conditional transmission probabilities $\mu_i = \mu, \forall i$, in four different settings: Slotted-ALOHA networks with $L = 1$ and two values of $\lambda \in \{0.05, 0.5\}$; and CSMA networks with $L = 50$ and two values of $\lambda \in \{0.05, 0.5\}$. We simulate the Random Access networks described in Sec. 6.1, and compare the simulation results with the analytical expressions of the transmission probability $q$ in (6.4) and NAoI in (6.11).

In Figs. 6-3, 6-4, 6-5, and 6-6, we simulate networks with increasing conditional transmission probability $\mu \in (0, 1]$. In Figs. 6-3 and 6-4, we simulate Slotted-ALOHA networks with packet transmission duration of $L = 1$ mini-slot, and in Figs. 6-5 and 6-6, we simulate CSMA networks with $L = 50$. In Figs. 6-3 and 6-5, we plot the network AoI, and in Figs. 6-4 and 6-6, we plot the transmission probability $q$. Simulations have a time-horizon of $K = 20 \times 10^6$ mini-slots, each mini-slot with normalized duration $\delta = 1$.

From Figs. 6-3, 6-4, 6-5, and 6-6, it is evident that the analytical expressions for $q$ and NAoI developed in Secs. 6.1 and 6.2, respectively, closely follow the simulation results in a wide range of network configurations, including low and high values of packet transmission duration $L \in \{1, 50\}$, low and high packet generation probabilities $\lambda \in \{0.05, 0.5\}$ and conditional transmission probabilities $\mu$ in the interval $(0, 1]$.

[1] A lower bound on the second moment of the inter-delivery interval can be obtained using a similar approach as in Proposition 6.2.

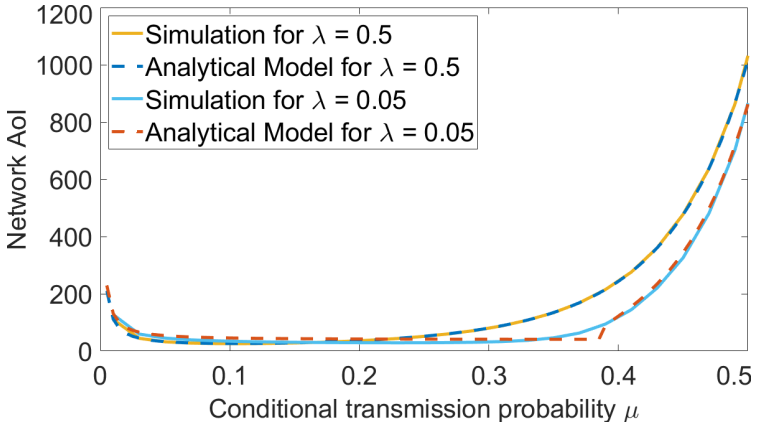



Figure 6-3: Simulation of symmetric Slotted-ALOHA networks with $L = 1$, increasing conditional transmission probability $\mu$ and two different packet generation probabilities $\lambda$.

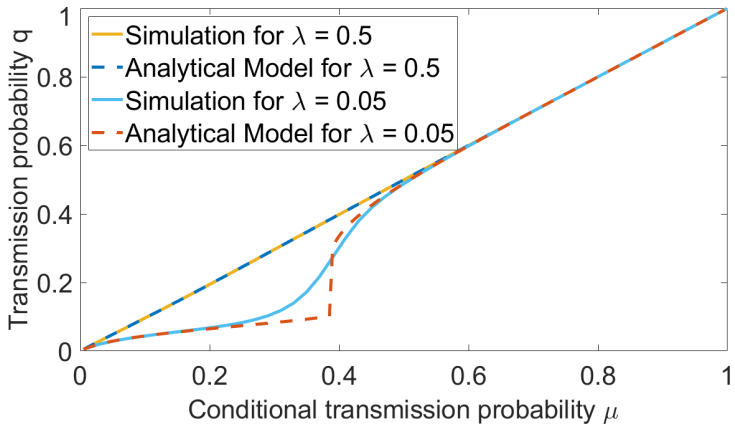



Figure 6-4: Simulation of symmetric Slotted-ALOHA networks with $L = 1$, increasing conditional transmission probability $\mu$ and two different packet generation probabilities $\lambda$.

By comparing Figs. 6-3 and 6-5, we can observe that networks with larger packet duration $L$ are less sensitive to changes in the packet generation probability $\lambda$. Recall that $\lambda$ directly affects the number of epochs in which the transmission queue is empty $N_i^B$ and the packet delay $z_i$. A larger $L$ significantly reduces $N_i^B$ and increases the inter-delivery interval $I_i$, which reduces the impact of $N_i^B$ and $z_i$ on the NAoI performance, thus making the network with larger $L$ less sensitive to variations in $\lambda$.

Figures 6-3 and 6-5 also show that: 1) a sub-optimal operating point $\mu$ can severely degrade the NAoI performance of the network; and 2) the point of minimum NAoI changes significantly in different network settings. Both observations highlight the importance of

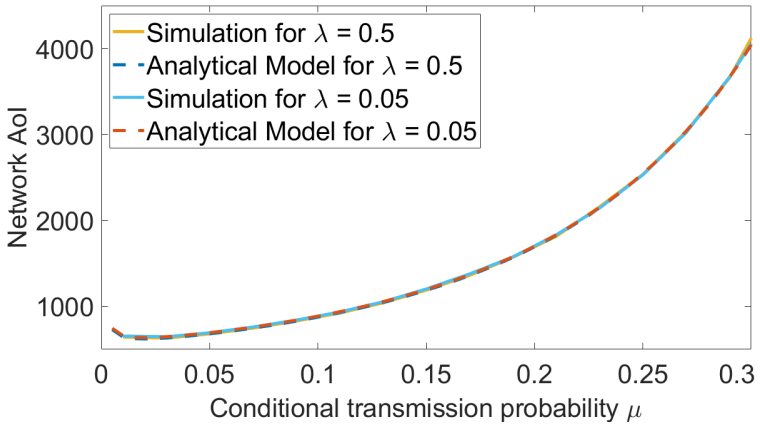



Figure 6-5: Simulation of symmetric CSMA networks with $L = 50$, increasing conditional transmission probability $\mu$ and two different packet generation probabilities $\lambda$.

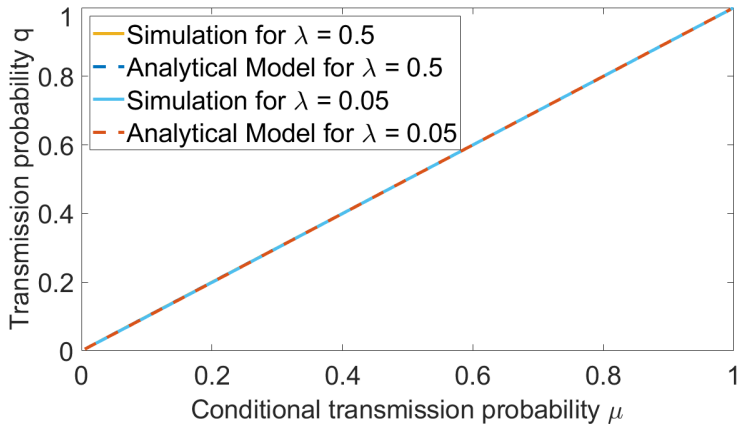



Figure 6-6: Simulation of symmetric CSMA networks with $L = 50$, increasing conditional transmission probability $\mu$ and two different packet generation probabilities $\lambda$.

optimizing NAoI in Random Access networks, which is addressed in the next section.

Figures 6-4 and 6-6 show that for high $L$ and/or high $\lambda$, the transmission probability $q$ is comparable to the conditional transmission probability $\mu$, i.e. $q \approx \mu$. In contrast, when $\lambda$ is low, the relationship between $q$ and $\mu$, which is governed by the *iterated function* $q = g(q,\mu,\lambda)$ in (6.4), is more involved. Notice that the plot in Fig. 6-4 for networks with $L = 1$ shows a discontinuity around $\mu = 0.4$ when $\lambda = 0.05$. In turn, for networks with $L = 50$, this discontinuity appears for $\lambda < 0.05$. Figure 6-7 shows the discontinuity for $\lambda \in \{0.0013, 0.0019\}$. This discontinuity plays an important role in the optimization of NAoI, as we will discuss in Propositions 6.4 and 6.6.

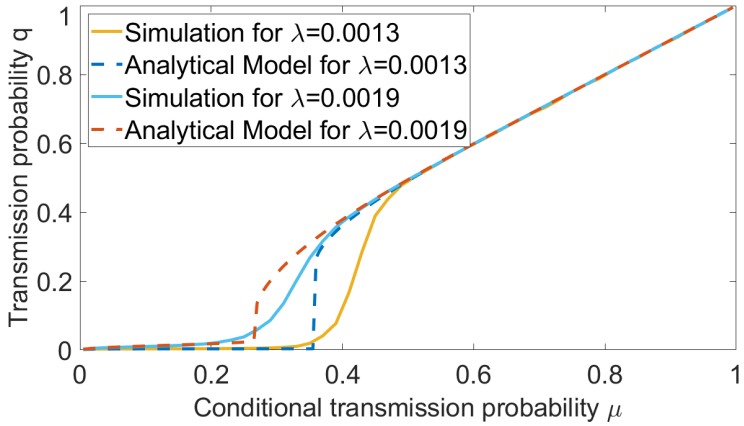



Figure 6-7: Simulation of symmetric CSMA networks with $L = 50$, increasing conditional transmission probability $\mu$ and two different packet generation probabilities $\lambda$.

## 6.3 Network Optimization

In this section, we optimize NAoI in *symmetric* Random Access networks with packet generation probabilities $\lambda_i = \lambda \in (0,1], \forall i$, and conditional transmission probabilities $\mu_i = \mu \in (0,1], \forall i$. In particular, we find the optimal value of $\mu$ in terms of the parameters $(N, L, \lambda)$ for three important cases: 1) Slotted-ALOHA networks, in which $L = 1$; 2) saturated CSMA networks, in which $L > 1$ and packets are generated on demand, i.e. $\lambda = 1$; and 3) general CSMA networks with low packet generation probability $\lambda \ll 1$. Then, we show that the three cases are strongly interconnected. In particular, we show that the results of the third case subsume the results of the first two cases. In Sec. 6.4, we compare the analytical optimization of NAoI with experimental and numerical results.

### 6.3.1 Slotted-ALOHA networks

Consider a symmetric Slotted-ALOHA network. Substituting $L = 1$ into the expression of NAoI in (6.11), we obtain

$$NAoI \approx \frac{1-\lambda}{\lambda} + \frac{1}{\mu Q} + \frac{\frac{1-\lambda}{\lambda^2}}{\frac{1-\lambda}{\lambda} + \frac{1}{\mu Q}}, \tag{6.13}$$

where $Q = (1-q)^{N-1}$ is the probability that all but one of the nodes are idle during an arbitrary epoch $t$ and $q$ is the transmission probability given by

$$q = \frac{1}{\frac{1-\lambda}{\lambda}Q + \frac{1}{\mu}} \,. \tag{6.14}$$

**Proposition 6.4.** *In a symmetric Slotted-ALOHA network, the solution candidates* $\mu \in (0,1)$ *for the NAoI minimization are given by*

$$\mu^{(1)} = \frac{1}{N - \frac{1-\lambda}{\lambda}\left(1-\frac{1}{N}\right)^{N-1}} \,; \tag{6.15a}$$

$$\mu^{(2)} = \frac{q^{(2)}}{1-\sqrt{1-\lambda}} \,; \tag{6.15b}$$

$$\mu^{(3)} = \frac{\left(q^{(3)}\right)^2 (N-1)}{q^{(3)}N - 1} \,, \tag{6.15c}$$

*where* $q^{(2)} \in (0,1)$ *and* $q^{(3)} \in [1/N, 1)$ *are the solutions to the equations below*

$$q^{(2)}\left(1-q^{(2)}\right)^{N-1} = \frac{\lambda}{\sqrt{1-\lambda}} \,; \tag{6.16a}$$

$$\left(q^{(3)}\right)^2\left(1-q^{(3)}\right)^{N-2} = \frac{\lambda}{(1-\lambda)(N-1)} \,. \tag{6.16b}$$

*Proof.* To find the value of $\mu \in (0,1)$ that minimizes NAoI, we analyze (6.13). The challenge is that the expression of NAoI in (6.13) is a function of $\mu$, $\lambda$, and $q$, where $q$ is not directly controllable. The transmission probability $q$ is determined by the *iterated function* $q = g(q,\mu,\lambda)$ in (6.14). To simplify the expression of NAoI, we substitute (6.14) into (6.13), which gives

$$NAoI \approx \frac{1}{q(1-q)^{N-1}} + \frac{1-\lambda}{\lambda^2} q(1-q)^{N-1} \,, \tag{6.17}$$

where $q(1-q)^{N-1}$ is the probability of a successful transmission from any given source $i$.

By analyzing the expression of NAoI in (6.17) and its partial derivative with respect to $q$, and then analyzing the iterated function in (6.14) together with its first and second partial derivatives, we obtain the solution candidates in (6.15a)-(6.15c). Notice that the analysis of the iterated function is non-trivial, since the associated *fixed points* $q$ are not guaranteed to span the entire interval $(0,1)$ due to the discontinuities displayed in Figs. 6-4 and 6-7. The complete proof is provided in Appendix 6.B. ■

**Global minimum NAoI.** For a given packet generation probability $\lambda \in (0,1]$, we calculate the solution candidates $\mu^{(j)}$, for $j \in \{1,2,3\}$, using (6.15a)-(6.16b). Then, we substitute $\lambda$ and $\mu^{(j)}$ into (6.14) to find the associated transmission probability $q$. Finally, by substituting $\lambda$, $\mu^{(j)}$ and $q$ into the NAoI expression in (6.13), we can find and compare the values of NAoI from the different solution candidates. The solution candidate that yields the lowest NAoI is the global minimizer. Notice that this is a low-complexity procedure. Equations (6.16a) and (6.16b) have up to four solutions, meaning that the total number of solution candidates from (6.15a)-(6.15c) is at most five, for any values of $N$ and $\lambda$.

### 6.3.2 Saturated CSMA networks

In this section, we optimize NAoI in symmetric CSMA networks with sources that generate packets on demand, i.e. $\lambda = 1$.

**Proposition 6.5.** *In a symmetric CSMA network with $\lambda = 1$, and large values of $N$ and $L$. The value of $\mu$ that minimizes NAoI is given by*

$$\mu^* \approx \frac{1}{N}\sqrt{\frac{2}{L}}\,. \tag{6.18}$$

*Proof.* Substituting $\lambda = 1$ into the expression of the transmission probability $q$ in (6.4), yields $q = \mu$. Taking the partial derivative of NAoI in (6.11) with respect to $\mu$ gives

$$\frac{2(L-1)}{\mu^2} - \frac{2L(1-N\mu)}{\mu^2(1-\mu)^N} + \frac{L(L-1)N(1-\mu)^{N-1}}{(L-(L-1)(1-\mu)^N)^2}\,. \tag{6.19}$$

Since the first and second terms of the partial derivative in (6.19) are dominant, especially for CSMA networks with large $L$, we neglect the third term, equate the partial derivative to zero, and obtain

$$(L-1)(1-\mu)^N = L(1-N\mu)\,, \tag{6.20}$$

which has a unique solution $\mu^* \in (0, 1/N]$. Then, we approximate $(1-\mu)^N$ in (6.20) by its second degree Taylor Polynomial to obtain the closed-form solution

$$\mu^* \approx \frac{-N+\sqrt{N^2+2(L-1)N(N-1)}}{(L-1)N(N-1)}\,. \tag{6.21}$$

Notice that when $N$ and $L$ are large, equation (6.21) is equivalent to (6.18). ■

### 6.3.3 General CSMA networks

In this section, we optimize NAoI in symmetric CSMA networks with low packet generation probability, $\lambda \ll 1$, and then discuss the relationship between the NAoI optimization for general CSMA networks, saturated CSMA networks, and Slotted-ALOHA networks.

**Proposition 6.6.** *In a symmetric CSMA network with $\lambda \ll 1$. The solution candidates $\mu \in (0,1)$ for the NAoI minimization are given by*

$$\mu^{(j)} = \left(\frac{1}{q^{(j)}} - \frac{(1-\lambda)^L Q^{(j)}}{1-(1-\lambda)Q^{(j)}-(1-\lambda)^L(1-Q^{(j)})}\right)^{-1}, \tag{6.22}$$

*for $j \in \{1,2,3\}$, where $Q^{(j)} = (1-q^{(j)})^{N-1}$ and $q^{(j)} \in (0,1)$ are the solutions to the equations below*

$$(L-1)\left(1-q^{(1)}\right)^N = L\left(1-Nq^{(1)}\right)\,, \tag{6.23a}$$

$$\left(q^{(2)}\right)^2\left(1-q^{(2)}\right)^{N-2} = \frac{\lambda\left[L-(L-1)\left(1-q^{(2)}\right)^{N-1}\right]^2}{(1-\lambda L)L(N-1)}\,, \tag{6.23b}$$

*and*

$$2\lambda^2 L^2\left(1-q^{(3)}\right)^2-4\lambda^2(L-1)L\left(1-q^{(3)}\right)^{N+2}+$$
$$+\left(1-q^{(3)}\right)^{2N}\lambda^2(L-1)\left[L\left(3\left(q^{(3)}\right)^2-4q^{(3)}+2\right)-2\left(1-q^{(3)}\right)^2\right]+$$
$$+\left(1-q^{(3)}\right)^{2N}\left(q^{(3)}\right)^2(\lambda(L+1)-2)=0\,. \tag{6.23c}$$

*Proof.* The proof follows similar steps as Proposition 6.4. The main difference is that the expressions for NAoI in (6.11) and the iterated function in (6.4) are more challenging to analyze for CSMA networks than for Slotted-ALOHA networks. To simplify the analysis of (6.4) and (6.11) for networks with $\lambda \ll 1$, we use the binomial approximation $(1-\lambda)^L \approx 1-\lambda L$. The complete proof is provided in Appendix 6.C. ■

**Global minimum NAoI.** To find the solution candidate $\mu^{(j)}$ that yields the lowest NAoI, we follow the low-complexity procedure described at the end of Sec. 6.3.1 using the results in Proposition 6.6, the expression for the transmission probability $q$ in (6.4), and the expression for NAoI in (6.11). After running this procedure for various network configurations, we observe that *in practice* the candidate $\mu^{(2)}$ associated with (6.23b) is the minimizer when $\lambda$ is low, the candidate $\mu^{(1)}$ associated with (6.23a) is the minimizer when $\lambda$ is (relatively) high, and the candidate $\mu^{(3)}$ associated with (6.23c) is never the minimizer. This observation is illustrated in Sec. 6.4.2.

Propositions 6.4, 6.5 and 6.6 determine the optimal value of $\mu$ for Slotted-ALOHA networks, saturated CSMA networks with $\lambda = 1$ and general CSMA networks with $\lambda \ll 1$, respectively. Despite these differences, we show that the three propositions are strongly interconnected. In particular, we show that the results in Propositions 6.4 and 6.5 are special cases of Proposition 6.6. Consider (6.23a)-(6.23c) from Proposition 6.6. Notice that when $\lambda = 1$, the equation in (6.23a) is equivalent to (6.20) from Proposition 6.5, and when $L = 1$, the equation in (6.23a) yields (6.15a) from Proposition 6.4. Similarly, it is easy to see that when $L = 1$, the equations in (6.23b) and (6.23c) are equivalent to (6.16b) and (6.16a) from Proposition 6.4, respectively. Hence, the results in Proposition 6.6 apply not only to CSMA

networks with $\lambda \ll 1$, but also to Slotted-ALOHA networks with $\lambda \in (0,1]$, and to CSMA networks with $\lambda = 1$. Thus, we *conjecture that the solution candidates* $\mu$ *for the NAoI minimization given in Proposition 6.6 are a good approximation to the optimal solution for general Random Access networks with arbitrary parameters* $(N, L, \lambda)$. Next, we validate this conjecture by comparing the analytical optimization of NAoI in Proposition 6.6 with experimental and numerical results.

## 6.4 Experimental and Numerical Results

In this section, we describe the experimental setup and then compare the analytical expressions for the NAoI performance (Theorem 6.3) and NAoI optimization (Proposition 6.6) with numerical and experimental results. Prior to delving into the details of the experimental setup, we describe the key characteristics of the *optimized CSMA network* that we implemented:

- sources use queues that keep only the freshest packet, as described in Sec. 6.1;
- sources have a conditional transmission probability $\mu$ that can be tuned to the optimal value. To adjust the conditional transmission probability to a given $\mu' \in (0,1]$, we set[2] the contention window of the Distributed Coordination Function (DCF) to $W = 2/\mu' - 1$, as proposed in [14, Eq.(8)]; and
- the BS has a time-stamp manager that logs the evolution of $h_i(k)$ over time for every source in the network. Notice that keeping track of $h_i(k) = k - \tau_i(k)$ requires that all nodes in the network are synchronized.

### 6.4.1 Experimental Setup

We implement the *optimized CSMA network* in the FPGA-based Software Defined Radio (SDR) testbed in Fig. 6-8 composed of one NI USRP 2974 operating as the Base Station,

[2] In this chapter, we manually set $W = 2/\mu' - 1$. Distributed algorithms that can dynamically tune $\mu$ aiming to maximize throughput in CSMA networks were developed in various works including [15,28,33,43]. Similar algorithms can be used to tune $\mu$ for minimizing NAoI. The implementation of such algorithms is out of the scope of this thesis.

and ten sources: seven NI USRP 2974 and three NI USRP 2953R. The code is developed using a modifiable WiFi reference design [40] with Transport layer based on UDP, MAC layer based on DCF, PHY layer based on the IEEE 802.11n standard with center frequency 2.437 GHz, bandwidth of 20 MHz, mini-slot duration of $\delta = 9\mu$secs, and a fixed MCS index of 5. *We use this WiFi reference design as a starting point, and implement the queueing discipline, the mechanism for adjusting $\mu$, and the time-stamp manager at the FPGA of the SDRs using hardware-level programming.*

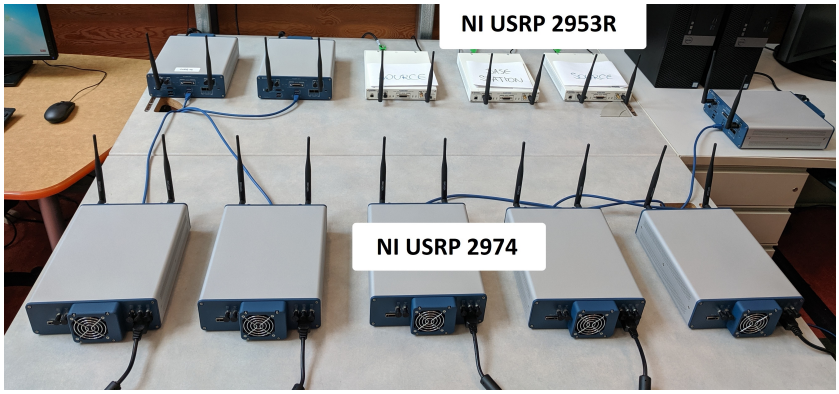



Figure 6-8: Software Defined Radio testbed.

### 6.4.2 Results and Discussion

In this section, we evaluate the NAoI performance and optimization using experimental, numerical, and analytical results. We consider a network with $N = 10$ sources, $L = 50$ mini-slots, mini-slot duration of $\delta = 9\mu$secs, and different values of $\lambda$ and $\mu$, and we compare:

- **Experimental results**, in which we run the SDR testbed for ten minutes and measure the time-average NAoI as in (6.10). Each SDR generates packets of 280 bytes with a period of $\delta/\lambda$ seconds and transmits these packets at a rate of approximately $5M$bps, which gives a packet transmission duration of approximately $L = 50$ mini-slots. For each fixed value of $\lambda \in \{2.25, 4.5, 9, 45\} \times 10^{-3}$, we find the optimal $\mu^*$ by comparing the values of NAoI for different contention windows $W \in \{8, 16, 32, 64, 128, 256\}$. Recall that $\mu = 2/(W+1)$;

- **Simulation results**, in which we obtain NAoI by simulating the Random Access network described in Sec. 6.1 for a time-horizon of $K = 20 \times 10^6$ mini-slots. For each fixed $\lambda \in (0,1)$, we find the optimal $\mu^*$ by comparing the values of NAoI for different values of $\mu \in (0,1)$; and
- **Analytical Model**, in which we compute NAoI using Theorem 6.3 and compute the optimal $\mu^*$ associated with a given $\lambda \in (0,1)$ using Proposition 6.6.

Table 6.1: NAoI performance (in milliseconds) from the experiments with the SDR testbed, from the simulation results, and from the analytical expression in Theorem 6.3.

| Experimental results | | | | | | |
|---|---|---|---|---|---|---|
| $W$ | 8 | 16 | 32 | 64 | 128 | 256 |
| $\lambda = 2.25 \times 10^{-3}$ | 12.82 | 6.79 | 6.77 | 7.11 | 8.39 | 9.10 |
| $\lambda = 4.5 \times 10^{-3}$ | 25.24 | 6.89 | 6.50 | 7.09 | 7.82 | 8.56 |
| $\lambda = 9 \times 10^{-3}$ | 24.21 | 8.46 | 6.89 | 6.41 | 7.50 | 8.79 |
| $\lambda = 45 \times 10^{-3}$ | 20.87 | 9.13 | 6.38 | 5.98 | 6.43 | 7.56 |
| Simulation results | | | | | | |
| $W$ | 8 | 16 | 32 | 64 | 128 | 256 |
| $\lambda = 2.25 \times 10^{-3}$ | 10.97 | 6.40 | 6.18 | 6.43 | 6.95 | 8.73 |
| $\lambda = 4.5 \times 10^{-3}$ | 20.02 | 8.45 | 6.40 | 5.78 | 5.93 | 7.19 |
| $\lambda = 9 \times 10^{-3}$ | 21.58 | 8.91 | 6.62 | 5.80 | 5.85 | 6.96 |
| $\lambda = 45 \times 10^{-3}$ | 22.93 | 9.56 | 6.84 | 5.81 | 5.74 | 6.72 |
| Analytical Model | | | | | | |
| $W$ | 8 | 16 | 32 | 64 | 128 | 256 |
| $\lambda = 2.25 \times 10^{-3}$ | 11.17 | 9.54 | 9.59 | 9.74 | 10.03 | 11.18 |
| $\lambda = 4.5 \times 10^{-3}$ | 18.35 | 8.37 | 6.99 | 6.70 | 6.93 | 8.22 |
| $\lambda = 9 \times 10^{-3}$ | 21.29 | 8.99 | 6.75 | 5.99 | 6.04 | 7.17 |
| $\lambda = 45 \times 10^{-3}$ | 22.91 | 9.50 | 6.84 | 5.83 | 5.77 | 6.72 |

In Table 6.1, we display the NAoI performance from experimental, simulation, and analytical results, for $W \in \{8,16,32,64,128,256\}$ and $\lambda \in \{2.25,4.5,9,45\} \times 10^{-3}$. The results in Table 6.1 show that the analytical model closely follows both the simulation and experimental results, and is particularly accurate when $\lambda$ is large. The lower accuracy of the analytical model for small $\lambda$ is a result of two approximations: 1) $q(t) \approx q$, introduced in

Sec. 6.1.1; and 2) $z \approx \tilde{z}$, introduced in the proof of Theorem 6.3. Notice that when packet generation is infrequent, i.e. $\lambda$ is small, the transmission probability in epoch $t$ is often $q(t) = 0$ and the time-averaged transmission probability $q$ can be larger than 0, depending on $\mu$, as illustrated in Figs. 6-4, 6-6, and 6-7, meaning that the approximation $q(t) \approx q$ may be inaccurate. In addition, when $\lambda$ is small, the packet delay $z$ becomes an important factor in the NAoI analysis, and the upper bound $z \leq \tilde{z}$ becomes less tight. In contrast, when $\lambda$ is large, which is the *region of interest for the NAoI optimization*, both approximations are accurate.

The transmission probability $q$ of the sources is directly affected by $W$ and $\lambda$. In particular, a larger $W$ results in lower $\mu$ and $q$, and a lower $\lambda$ results in lower $q$. The results in Table 6.1 reflect this relationship and its impact on the NAoI performance. Notice that when the contention window is large, e.g. $W = 256$, the transmission probability is (relatively) low, and NAoI improves for larger $\lambda$. In contrast, when the contention window is small, e.g. $W = 8$, the transmission probability is (relatively) large, and NAoI improves for smaller $\lambda$. As expected, this behavior is observed in the experimental, simulation, and analytical results.

In Figs. 6-9 and 6-10, we display the optimal $\mu^*$ and the corresponding minimum $NAoI^*$, respectively, for different values of $\lambda \in (0, 0.05]$. For $\lambda \geq 0.05$, we note that the optimal conditional transmission probability remains constant at $\mu^* = 0.02$ and the value of $NAoI^*$ decreases with $\lambda$, achieving $NAoI^* = 5.58$ milliseconds when $\lambda = 1$. The optimal conditional transmission probability $\mu^*$ from experimental results is obtained using the measurements in Table 6.1. Figures 6-9 and 6-10 show that the analytical results in Secs. 6.2 and 6.3 closely follow simulation and experimental results.

In Sec. 6.3.3, we noted that *in practice* the global minimizer $\mu^*$ obtained using Proposition 6.6 displays a threshold structure. In particular, for each network configuration with parameters $(N, L)$, there exists a threshold $\lambda' \in (0, 1)$ such that, when $\lambda < \lambda'$, the minimizer is the candidate $\mu^{(2)}$ associated with (6.23b) and, when $\lambda \geq \lambda'$, the minimizer is the candidate $\mu^{(1)}$ associated with (6.23a). By matching the optimal values of $\mu^*$ from the analytical results in Fig. 6-9 with the solution candidates $\mu^{(j)}$ in Proposition 6.6, we find that the threshold for $N = 10$ and $L = 50$ is $\lambda' = 0.00187$. This threshold structure is

observed in various network configurations, and it can be used to reduce the computational complexity of finding the global minimizer $\mu^*$ using Proposition 6.6. Moreover, to further reduce the complexity, we provide an approximate solution for $\mu^{(1)}$. Notice from (6.22) and (6.23a) that when $\lambda$ is high, an approximate solution for $\mu^{(1)}$ and $q^{(1)}$ is given by

$$\mu^{(1)} \approx q^{(1)} \approx \frac{1}{N}\sqrt{\frac{2}{L}} = 0.02\,, \tag{6.24}$$

which coincides with $\mu^*$ in Fig. 6-9 for $\lambda > 0.01$.

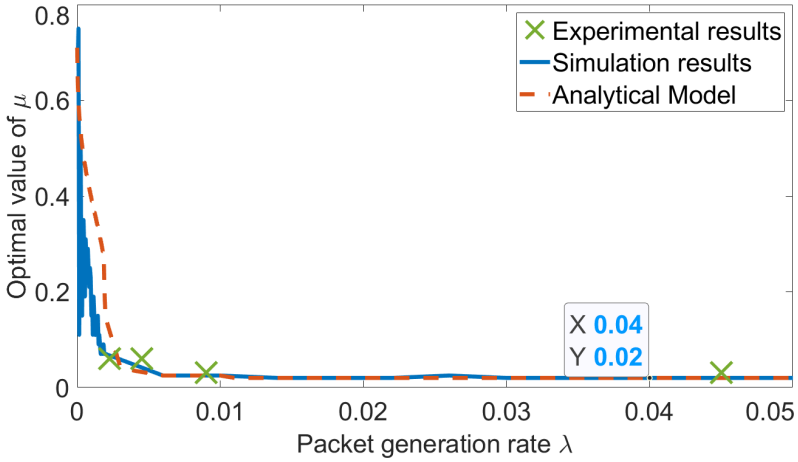



Figure 6-9: Optimal conditional transmission probability $\mu^*$ obtained from the experiments with the SDR testbed, from the simulation results, and from the analysis in Proposition 6.6.

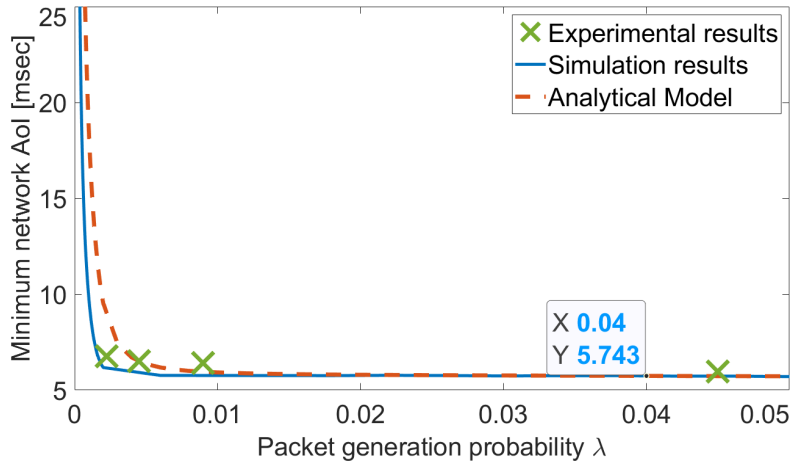



Figure 6-10: Optimal NAoI performance associated with the optimal $\mu^*$.

## 6.5 Summary

In this chapter, we studied AoI in networks employing Random Access mechanisms. We considered a wireless network with a number of nodes generating packets according to a Bernoulli process and employing Slotted-ALOHA or CSMA to transmit these packets to the BS. We proposed a framework to analyze and optimize the average AoI in the wireless network. In particular, we developed a discrete-time model and derived expressions for: the time-average transmission probability, a lower bound on the inter-delivery interval, an upper bound on the packet delay, and an (accurate) approximation for the average AoI in the network. We then used the analytical expressions to optimize the Random Access mechanism in terms of AoI. Furthermore, we implemented the optimized CSMA network in a Software Defined Radio testbed and compared the AoI measurements with analytical and numerical results in order to validate our framework. We showed that the analytical results accurately track both the simulation and experimental results. Our approach allowed us to evaluate the combined impact of the packet generation rate, transmission probability, and size of the network on the AoI performance.

# Appendices

## 6.A Proof of Proposition 6.1

In this appendix, we obtain the closed-form expression for the *transmission probability* $q_i$ displayed in Proposition 6.1. Consider the time interval between two consecutive packet deliveries from source $i$ illustrated in Fig. 6-1. To obtain $q_i$ in (6.4), we derive expressions for $\mathbb{E}[N_i^B]$ and $\mathbb{E}[N_i^A]$, and substitute them into the definition of $q_i$ in (6.2) which is rewritten below for convenience

$$q_i = \frac{(\mathbb{E}[N_i^A]+1)\mu_i}{\mathbb{E}[N_i^B]+(\mathbb{E}[N_i^A]+1)} \,.$$

We start by analyzing $\mathbb{E}[X_i^B(t)]$ and $\mathbb{E}[X_i^A(t)]$. Since $X_i^B(t)$ is the length of epoch $t$ within the interval $N_i^B$ in which source $i$ does not have a packet to transmit, it follows that

$$\mathbb{P}(X_i^B(t)=1) = \prod_{j=1,j\neq i}^{N}(1-q_j) = Q^{-i}\,, \tag{6.25a}$$

$$\mathbb{P}(X_i^B(t)=L) = 1-Q^{-i}\,. \tag{6.25b}$$

In addition, since $X_i^A(t)$ is the length of epoch $t$ within the interval $N_i^A$ in which source $i$ may transmit, but cannot deliver its packet, it follows that

$$\mathbb{P}(X_i^A(t)=1) = \frac{(1-\mu_i)Q^{-i}}{1-\mu_i Q^{-i}}\,, \tag{6.26a}$$

$$\mathbb{P}(X_i^A(t)=L) = \frac{1-Q^{-i}}{1-\mu_i Q^{-i}}\,, \tag{6.26b}$$

where the numerator in (6.26a) represents the probability of no source transmitting a packet, and the denominator represents the probability of source $i$ not delivering its packet. From (6.25a), (6.25b), (6.26a), and (6.26b), we obtain the expected values below

$$\mathbb{E}[X_i^B(t)] = Q^{-i}+L(1-Q^{-i})\,, \tag{6.27a}$$

$$\mathbb{E}[X_i^A(t)] = \frac{(1-\mu_i)Q^{-i}}{1-\mu_i Q^{-i}} + L\,\frac{1-Q^{-i}}{1-\mu_i Q^{-i}}\,. \tag{6.27b}$$

Next, we analyze $\mathbb{E}[N_i^B]$ and $\mathbb{E}[N_i^A]$. By the definition of the interval $N_i^B$, it follows that

$$\mathbb{P}(N_i^B = 0) = 1 - (1-\lambda_i)^L \,; \tag{6.28a}$$

$$\mathbb{P}(N_i^B = n | \{X(n)\}_{m=1}^n) = \left[1 - (1-\lambda_i)^{X(n)}\right] \times \times (1-\lambda_i)^L \prod_{m=1}^{n-1} (1-\lambda_i)^{X(m)}, \forall n \in \{1,2,\cdots\} \,. \tag{6.28b}$$

The first term on the RHS of (6.28b) represents the probability of a packet generation on the $n$th epoch of the interval $N_i^B$. The second and third terms on the RHS of (6.28b) represent the probability of no packet generation prior to the $n$th epoch. To obtain the expectation of $N_i^B$, we use the probabilities in (6.28a)-(6.28b), employ the law of iterated expectations, and then use the fact that $X_i^B(t)$ are i.i.d. with mean given by (6.27a), which yields

$$\mathbb{E}[N_i^B] = \frac{(1-\lambda_i)^L}{1-(1-\lambda_i)Q^{-i}-(1-\lambda_i)^L(1-Q^{-i})} \,. \tag{6.29}$$

In addition, by the definition of the interval $N_i^A$, it follows that

$$\mathbb{P}(N_i^A = n) = \left[1-\mu_i Q^{-i}\right]^n \mu_i Q^{-i}, \forall n \in \{0,1,\cdots\} \,, \tag{6.30}$$

which gives the expected value below

$$\mathbb{E}[N_i^A] = \frac{1-\mu_i Q^{-i}}{\mu_i Q^{-i}} \,. \tag{6.31}$$

Finally, substituting (6.29) and (6.31) into the transmission probability in (6.2) gives the closed-form expression in (6.4).

## 6.B Proof of Proposition 6.4

In this appendix, we derive the solution candidates $\mu \in (0,1]$ displayed in Proposition 6.4 for the NAoI minimization in symmetric Slotted-ALOHA networks. First, we analyze the expression of NAoI in (6.17) and then we analyze the expression of the transmission probability $q$ in (6.14).

### 6.B.1 Network AoI

The expression of NAoI in (6.17) is central to the analysis and is rewritten below for convenience

$$NAoI \approx \frac{1}{q(1-q)^{N-1}} + \frac{1-\lambda}{\lambda^2} q(1-q)^{N-1} .$$

The partial derivative of NAoI with respect to the transmission probability $q$ is given by

$$\frac{\partial NAoI}{\partial q} \approx \frac{(Nq-1)(\lambda^2 - q^2(1-q)^{2N-2}(1-\lambda))}{\lambda^2 q^2 (1-q)^N} . \tag{6.32}$$

This partial derivative has up to three roots $q^{(1)}$, $q_1^{(2)}$ and $q_2^{(2)}$, characterized by the equations below

$$q^{(1)} = \frac{1}{N} \quad \text{and} \quad q^{(2)}(1-q^{(2)})^{N-1} = \frac{\lambda}{\sqrt{1-\lambda}} , \tag{6.33}$$

which are dependent on the packet generation probability $\lambda$. Next, we divide $\lambda$ into three cases. **Case 1:** if $\lambda$ is high enough such that

$$\frac{1}{N}\left(1-\frac{1}{N}\right)^{N-1} < \frac{\lambda}{\sqrt{1-\lambda}} , \tag{6.34}$$

then the unique root $q^{(1)} = 1/N$ is the point of *global minimum* NAoI. **Case 2:** if $\lambda$ is such that

$$\frac{1}{N}\left(1-\frac{1}{N}\right)^{N-1} = \frac{\lambda}{\sqrt{1-\lambda}} , \tag{6.35}$$

then the three roots overlap $q^{(1)} = q_1^{(2)} = q_2^{(2)} = 1/N$ at the point of *global minimum* NAoI.
**Case 3:** if $\lambda$ is low enough such that

$$\frac{1}{N}\left(1-\frac{1}{N}\right)^{N-1} > \frac{\lambda}{\sqrt{1-\lambda}}\,, \tag{6.36}$$

then $q^{(1)} = 1/N$ is a point of *local maximum NAoI* and $q_1^{(2)}, q_2^{(2)}$ are points of *local minimum NAoI*. Notice that $q^{(1)} = 1/N$ is between $q_1^{(2)}$ and $q_2^{(2)}$.

**Summary.** For a given packet generation probability $\lambda \in (0,1]$, we use (6.33)-(6.36) to determine the transmission probabilities $q^{(1)}$ and $q^{(2)}$. Then, we substitute $q^{(1)}$ and $q^{(2)}$ into the expression that relates $\mu$ and $q$ in (6.14) to obtain the solution candidates $\mu^{(1)}$ and $\mu^{(2)}$ in (6.15a) and (6.15b), respectively. Next, we analyze the expression in (6.14) to derive the third solution candidate shown in Proposition 6.4.

### 6.B.2 Transmission Probability

The expression for the transmission probability (6.14) is rewritten below for convenience

$$q = \frac{1}{\frac{1-\lambda}{\lambda}(1-q)^{N-1} + \frac{1}{\mu}}\,.$$

This *iterated function* can be written as $q = g(q,\mu,\lambda)$, with $g$ continuous over $(q,\mu,\lambda) \in [0,1]\times(0,1]\times(0,1]$ and having extreme points

$$g(0,\mu,\lambda) = \frac{1}{\frac{1-\lambda}{\lambda}+\frac{1}{\mu}} \quad \text{and} \quad g(1,\mu,\lambda) = \mu\,. \tag{6.37}$$

Since $0 < g(0,\mu,\lambda) \le g(1,\mu,\lambda) < 1$ for all $(\mu,\lambda) \in (0,1]^2$, we conclude that $q = g(q,\mu,\lambda)$ has at least one fixed point in the interval $q \in [0,1]$.

Taking the first and second partial derivatives of $g(q,\mu,\lambda)$ with respect to $q$, we get

$$\frac{\partial g(q,\mu,\lambda)}{\partial q} = \frac{\frac{1-\lambda}{\lambda}(N-1)(1-q)^{N-2}}{\left(\frac{1-\lambda}{\lambda}(1-q)^{N-1}+\frac{1}{\mu}\right)^2}, \tag{6.38}$$

and

$$\frac{\partial^2 g(q,\mu,\lambda)}{\partial q^2} = \frac{(1-\lambda)\lambda\mu^2(N-1)(1-q)^N\left[(1-\lambda)\mu N(1-q)^{N-1}-\lambda(N-2)\right]}{\left[(1-\lambda)\mu(1-q)^N+\lambda(1-q)\right]^3}. \tag{6.39}$$

Combining the analysis of (6.14), (6.38) and (6.39), and taking into account the extreme points in (6.37), we conclude that $q = g(q,\mu,\lambda)$ has either

- one fixed point which is attracting, i.e. $\partial g(.)/\partial q < 1$
- two fixed points, one attracting and the other borderline, i.e. $\partial g(.)/\partial q = 1$; or
- three fixed points, two attracting and one repelling, i.e. $\partial g(.)/\partial q > 1$.

Hence, for any given $(\mu,\lambda)$, the iterated function has either one of two attracting fixed points. To obtain the attracting fixed points, we solve $q_{k+1} = g(q_k,\mu,\lambda)$ recursively using two initial values $q_0 = 0$ and $q_0 = \mu$.

**Interpretation.** When $\lambda \to 1$, sources often have packets in their transmission queues, resulting in a transmission probability $q$ that is comparable to the conditional transmission probability $\mu$. Substituting $\lambda = 1$ into the iterated function $q = g(q,\mu,\lambda)$ in (6.14), it is easy to see that $q = \mu$ is the unique attracting fixed point. Similarly, when $\mu \to 0$, sources rarely attempt to transmit and, thus, often have packets in their transmission queues, implying (again) that $q = \mu$ is a fixed point. In contrast, when $\lambda$ is low and $\mu$ is high, the network may be in one of two states: 1) *low_q state*, in which transmission queues are often empty and packet collisions are rare; or 2) *high_q state*, in which more than one source is attempting to transmit and packet collisions are frequent. The network *oscillates* between these two states. Intuitively, a transition to *high_q* may occur when two or more sources generate packets at the same time, and a transition to *low_q* may occur when all packets in the network are delivered to the BS. Each network state corresponds to an attractive fixed

point. For simplicity, in this chapter, we focus on the *high_q* state and neglect *low_q*. The numerical results in Sec. 6.2.3 show that the value of *high_q* obtained from the iterated function closely follows the simulated value of the transmission probability $q$ in a wide range of network parameters.

For a fixed $\lambda$, as $\mu$ increases from 0 to 1, the fixed points $q \in [0,1]$ tend to increase. Let $G(\lambda)$ be the set of all attracting fixed points $q$ associated with the iterated function $q = g(q,\mu,\lambda)$ in (6.14). From (6.14), we can see that $q \to 0$ and $q = 1$ are attracting fixed points associated with $\mu \to 0$ and $\mu = 1$, respectively. However, there is no guarantee that $G(\lambda)$ spans the entire interval $(0,1]$. Recall that as $\mu$ increases, the number of attracting fixed points may change from one to two, and vice versa. These changes may allow for gaps in $G(\lambda)$, leading to $G(\lambda)$ that does not span $(0,1]$. Notice that these *gaps are solution candidates* for the minimization of NAoI. A necessary condition for a gap to occur at the point $\mu^{(3)}$ is the existence of a *borderline fixed point* at $\mu^{(3)}$. Let $\mu^{(3)} \in (0,1]$ be the conditional transmission probability associated with a borderline fixed point. Then, it follows that there exists $q^{(3)} \in [0,1]$ such that

$$q^{(3)} = g(q^{(3)},\mu^{(3)},\lambda) \quad \text{and} \quad \frac{\partial g(q^{(3)},\mu^{(3)},\lambda)}{\partial q} = 1\,. \tag{6.40}$$

From the system of equations in (6.40), we obtain the solution candidate $\mu^{(3)}$ in (6.15c).

## 6.C Proof of Proposition 6.6

In this appendix, we derive the solution candidates $\mu \in (0,1]$ displayed in Proposition 6.6 for the NAoI minimization in symmetric CSMA networks with low packet generation rate $\lambda \ll 1$. To simplify the analysis, we use the binomial approximation

$$(1-\lambda)^L \approx 1-\lambda L\,, \tag{6.41}$$

which is accurate when $\lambda L \ll 1$. Substituting (6.41) into the expression for $q$ in (6.4), gives

$$q \approx \frac{1}{\frac{(1-\lambda L)Q}{\lambda(Q+L-QL)}+\frac{1}{\mu}}\,, \tag{6.42}$$

where $Q=(1-q)^{N-1}$ is the probability that all but one of the nodes are idle during an arbitrary slot $t$. Then, substituting (6.41) and (6.42) into the expression of NAoI in (6.11), yields

$$NAoI \approx \frac{Q+L-QL}{qQ}+\frac{5(L-1)}{2}+\frac{1}{2}\frac{\left(\frac{1}{\lambda}-L\right)\left(\frac{2}{\lambda}+L-1\right)-\left(\frac{1}{\mu}-1\right)(L-1)}{\frac{Q+L-QL}{qQ}+L-1}\,. \tag{6.43}$$

Notice that, for small $\lambda$, we have

$$\left(\frac{1}{\lambda}-L\right)\left(\frac{2}{\lambda}+L-1\right) \gg \left(\frac{1}{\mu}-1\right)(L-1)\,. \tag{6.44}$$

Hence, we can further simplify (6.43), and get

$$NAoI \approx \frac{Q+L-QL}{qQ}+\frac{5(L-1)}{2}+\frac{1}{2}\frac{\left(\frac{1}{\lambda}-L\right)\left(\frac{2}{\lambda}+L-1\right)}{\frac{Q+L-QL}{qQ}+L-1}\,, \tag{6.45}$$

which is a function of $\lambda$ and $q$. Next, we analyze the expression of NAoI in (6.45) and the expression of $q$ in (6.42) to determine the solution candidates $\mu \in (0,1]$ in Proposition 6.6

### 6.C.1 Network AoI

By taking the partial derivative of NAoI in (6.45) with respect to $q$ and setting it equal to zero, we obtain the following equations

$$(L-1)\left(1-q^{(1)}\right)^{N} = L\left(1-Nq^{(1)}\right) , \tag{6.46}$$

and

$$\begin{aligned} &2\lambda^2 L^2\left(1-q^{(3)}\right)^2 - 4\lambda^2(L-1)L\left(1-q^{(3)}\right)^{N+2} + \\ &\quad + \left(1-q^{(3)}\right)^{2N}\lambda^2(L-1)\left[L\left(3\left(q^{(3)}\right)^2 - 4q^{(3)}+2\right) - 2\left(1-q^{(3)}\right)^2\right] + \\ &\quad + \left(1-q^{(3)}\right)^{2N}\left(q^{(3)}\right)^2(\lambda(L+1)-2) = 0 \end{aligned} \tag{6.47}$$

For a given packet generation probability $\lambda \in (0,1]$, we use (6.46) and (6.47) to determine the transmission probabilities $q^{(1)}$ and $q^{(3)}$. Then, we substitute $q^{(1)}$ and $q^{(3)}$ into the expression that relates $\mu$ and $q$ in (6.22) to obtain the solution candidates $\mu^{(1)}$ and $\mu^{(3)}$ in Proposition 6.6. Notice that (6.46) and (6.47) are displayed in Proposition 6.6 as (6.23a) and (6.23c), respectively. Next, we analyze the expression of $q$ in (6.42) to derive the third solution candidate shown in Proposition 6.6.

### 6.C.2 Transmission Probability

The *iterated function* in (6.42) can be written as $q = g(q,\mu,\lambda)$, with $g$ continuous over $(q,\mu,\lambda) \in [0,1]\times(0,1]\times(0,1]$ and having extreme points

$$g(0,\mu,\lambda) = \frac{1}{\frac{1-\lambda L}{\lambda}+\frac{1}{\mu}} \quad \text{and} \quad g(1,\mu,\lambda) = \mu \, . \tag{6.48}$$

Assuming that $\lambda L < 1$, we have $0 < g(0,\mu,\lambda) \leq g(1,\mu,\lambda) < 1$ for all $(\mu,\lambda) \in (0,1]^2$. Hence, we conclude that $q = g(q,\mu,\lambda)$ has at least one fixed point in the interval $q \in [0,1]$.

Analyzing the first and second partial derivatives of $g(q,\mu,\lambda)$ with respect to $q$, and taking into consideration the extreme points in (6.48), we arrive at the same conclusions as Appendix 6.B, namely that:

- for any given $(\lambda,\mu)$ the iterated function $q = g(q,\mu,\lambda)$ has either one or two attracting fixed points; and
- the set $G(\lambda)$ of all attracting fixed points $q$ associated with the iterated function in (6.42) is not guaranteed to span the entire interval $(0,1]$ and the gaps are characterized by the system of equations

$$q^{(2)} = g(q^{(2)},\mu,\lambda) \quad \text{and} \quad \frac{\partial g(q^{(2)},\mu,\lambda)}{\partial q} = 1 \,. \tag{6.49}$$

By manipulating (6.49), we obtain the equation in (6.23b) for determining $q^{(2)}$.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

# Concluding Remarks

Future applications will increasingly rely on sharing time-sensitive information for monitoring and control. Examples are abundant: monitoring mobile ground-robots in automated fulfillment warehouses at Amazon [107, 111] and Alibaba [89]; collision prevention applications [61] for vehicles on the road [3, 16, 27, 63]; path planning, localization and motion control for multi-robot formations using drones [2, 4] and using ground-robots [106]; multi-drone system for tracking a mobile spectrum cheater [88]; multi-drone system for automated aerial cinematography [83]; multi-drone system for exploration of subterranean environments [79]; multi-robot simultaneous localization and mapping (SLAM) using drones [69, 77] and using ground-robots [75]; real-time surveillance system using a fleet of ground-robots [80]; and data collection from sensors, drones and cameras for agriculture using the Azure FarmBeats IoT platform [56, 108]. In such application domains, it is essential to keep the AoI low, as outdated information at the destination can lead to *system failures* and *safety risks*.

In this thesis, we addressed the problem of minimizing AoI in wireless networks. In particular, we considered a broadcast single-hop wireless network with a base station and a number of nodes sharing time-sensitive information through communication links. We formulated a discrete-time decision problem and used tools from stochastic control and mathematical optimization to find network control algorithms with *provable* performance guarantees and low computational complexity. Then, leveraging the theoretical results, we

proposed practical algorithms and implemented them in FPGA-based Software Defined Radios and/or Raspberry Pis to evaluate their performance in real operating conditions. Next, we summarize the main contributions of the thesis.

## 7.1 Summary of contributions

**Chapter 2. Age of Information in Wireless Networks**

In this chapter, we considered a broadcast single-hop wireless network with sources that generate fresh packets *on demand* and transmit them via unreliable communication links. We formulated the problem of optimizing transmission scheduling decisions with respect to the Expected Weighted Sum AoI in the network.

First, we obtained the AoI-optimal policy using Dynamic Programming. We showed that the computational complexity of such solution grows exponentially with the number of sources $N$, making it suitable for small networks. For large networks, we developed four low-complexity scheduling policies and derived performance guarantees for each of them as a function of network parameters, in particular the network size $N$, the channel reliabilities $\{p_i\}_{i=1}^N$ and the weights $\{w_i\}_{i=1}^N$. A summary of the main results follows:

- Maximum Age First policy is AoI-optimal for the case of symmetric networks, when all links have the same channel reliability $p_i = p$ and weight $w_i = w$. The performance guarantee for general networks is given in Theorem 2.9;
- Stationary Randomized policy with $\beta_i = \sqrt{w_i/p_i}$ is 2-optimal for any network configuration $(N, p_i, w_i)$;
- Max-Weight policy with $\tilde{\alpha}_i = \sqrt{w_i/p_i}$ is AoI-optimal for symmetric networks and 2-optimal for general networks; and
- Whittle's Index policy is AoI-optimal for symmetric networks. The performance guarantee for general networks is given in Theorem 2.21.

Simulation results show that both Max-Weight and Whittle's Index policies outperform the other scheduling policies in every configuration simulated, achieving near optimal information freshness.

To the best of our knowledge, this was the first work to derive performance guarantees for scheduling policies that attempt to minimize AoI in wireless networks with unreliable channels.

**Chapter 3. Throughput Constrained AoI Optimization**

In this chapter, we considered a broadcast single-hop wireless network with sources that generate fresh packets *on demand* and transmit them via unreliable communication links. We addressed the problem of minimizing the Expected Weighted Sum AoI in the network while simultaneously satisfying throughput requirements from the individual nodes. Throughput requirements can either capture an attribute of the nodes or be used to enforce fair allocation of resources in the network.

First, we derived a lower bound on the AoI performance achievable by any given network. Then, we developed two low-complexity transmission scheduling policies, namely Stationary Randomized and Drift-Plus-Penalty, and showed that both are 2-optimal for any network configuration, while simultaneously satisfying any feasible throughput requirements. Simulation results show that the Drift-Plus-Penalty policy outperforms other scheduling policies in every configuration simulated, achieving near optimal information freshness.

To the best of our knowledge, this was the first work to consider AoI-based policies that provably satisfy throughput constraints of multiple destinations simultaneously.

**Chapter 4. AoI in Wireless Networks with Stochastic Arrivals**

In this chapter, we considered a broadcast single-hop wireless network with sources that generate packets according to a stochastic process, enqueue them in separate (per source) queues, and transmit them via unreliable communication links. We addressed the problem of minimizing the Expected Weighted Sum AoI in the network.

First, we derived a lower bound on the AoI performance achievable by any given network, operating under any queueing discipline. Then, we considered three common queueing disciplines and developed both a Stationary Randomized policy and a Max-Weight policy under each discipline. A summary of the main results follows:

- Stationary Randomized policy for Single packet queues with optimal scheduling probability $\mu_i^S \propto \sqrt{w_i/p_i}$ is 4-optimal for any network configuration $(N, p_i, \lambda_i, w_i)$. Notice that, contrary to intuition, the optimal scheduling probability $\mu_i^S$ is independent of the packet arrival rate $\lambda_i$.

- Stationary Randomized policies for No queues and FCFS queues have optimal scheduling probabilities $\mu_i^N$ and $\mu_i^F$, respectively, that are sensitive to the packet arrival rate $\lambda_i$, as shown in Theorems 4.8 and 4.10.

- Max-Weight policies for Single packet queues and No queues are shown in Theorems 4.12 and 4.13 to outperform the corresponding Stationary Randomized Policies with the same queueing discipline.

We evaluated the AoI performance both analytically and using simulations. Our approach allowed us to evaluate the combined impact of the stochastic arrivals, queueing discipline and scheduling policy on AoI. Numerical results show that the Max-Weight policy with LCFS queues achieves near optimal performance in various network settings.

**Chapter 5. WiFresh: AoI from Theory to Implementation**

In this chapter, we proposed WiFresh: an unconventional architecture that achieves near optimal information freshness for wireless networks of any size. The superior performance of WiFresh is due to the combination of three elements: LCFS queues, Polling Multiple Access mechanism, and Max-Weight scheduling policy. The choice of each of these elements is underpinned by theoretical research. We proposed and realized two strategies for implementing WiFresh: 1) WiFresh Real-Time, in which our architecture is implemented at the MAC layer in a network of eleven FPGA-based Software Defined Radios using hardware-level programming; and 2) WiFresh App which is a customization of WiFresh implemented at the Application layer, without modifications to lower layers of the communication system, in a network of twenty five Raspberry Pis using Python 3. A key advantage of WiFresh App is that it can be easily integrated into time-sensitive applications that already run over WiFi such as [2–4, 16, 27, 56, 63, 69, 75, 77, 79, 80, 83, 88, 89, 106–108, 111]. Our experimental results showed that WiFresh can improve information freshness by two orders of

magnitude when compared to an equivalent standard WiFi network.

To the best of our knowledge, this was the first experimental evaluation of a networked system that scales gracefully in terms of information freshness.

**Chapter 6. AoI in Random Access Networks**

In this chapter, we studied AoI in networks employing Random Access mechanisms. We considered a wireless network with a number of nodes generating packets according to a Bernoulli process and employing Slotted-ALOHA or CSMA to transmit these packets to the BS. We proposed a framework to analyze and optimize the average AoI in the wireless network. In particular, we developed a discrete-time model and derived expressions for: the time-average transmission probability, a lower bound on the inter-delivery interval, an upper bound on the packet delay, and an (accurate) approximation for the average AoI in the network. We then used the analytical expressions to optimize the Random Access mechanism in terms of AoI. Furthermore, we implemented the optimized CSMA network in a Software Defined Radio testbed and compared the AoI measurements with analytical and numerical results in order to validate our framework. We showed that the analytical results accurately track both the simulation and experimental results. Our approach allowed us to evaluate the combined impact of the packet generation rate, transmission probability, and size of the network on the AoI performance.

To the best of our knowledge, this was the first work to provide theoretical results on the optimization of a CSMA network with stochastic packet generation and packet collisions.

THIS PAGE INTENTIONALLY LEFT BLANK

# Bibliography


[1] Norman Abramson. THE ALOHA SYSTEM: another alternative for computer communications. In *Proceedings of the Fall Joint Computer Conference*, AFIPS '70 (Fall), page 281–285, 1970.

[2] Evan Ackerman. This autonomous quadrotor swarm doesn't need GPS. IEEE Spectrum, online: https://spectrum.ieee.org/automaton/robotics/drones/this-autonomous-quadrotor-swarm-doesnt-need-gps, 2017.

[3] Thales T. Almeida, Lucas de C. Gomes, Fernando M. Ortiz, José G. R. Júnior, and Luís H. M. K. Costa. IEEE 802.11p performance evaluation: Simulations vs. real experiments. In *Proceedings of IEEE ITSC*, pages 3840–3845, 2018.

[4] Javier Alonso-Mora, Eduardo Montijano, Tobias Nägeli, Otmar Hilliges, Mac Schwager, and Daniela Rus. Distributed multi-robot formation control in dynamic environments. *Autonomous Robots*, 43:1079–1100, 2019.

[5] Eitan Altman, Rachid El-Azouzi, Daniel Sadoc Menasche, and Yuedong Xu. Forever young: Aging control for hybrid networks. In *Proceedings of ACM MobiHoc*, 2019.

[6] Diego Assencio. NMEA generator. online: https://nmeagen.org/, 2016.

[7] Baran Tan Bacinoglu, Elif Tugce Ceran, and Elif Uysal-Biyikoglu. Age of information under energy replenishment constraints. In *Proceedings of IEEE ITA*, 2015.

[8] Baran Tan Bacinoglu and Elif Uysal-Biyikoglu. Scheduling status updates to minimize age of information with an energy harvesting sensor. In *Proceedings of IEEE ISIT*, 2017.

[9] Andrea Baiocchi and Ion Turcanu. A model for the optimization of beacon message age-of-information in a VANET. In *29th International Teletraffic Congress*, 2017.

[10] Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff. The age of information in multihop networks. *IEEE/ACM Transactions on Networking*, 27(3):1248–1257, 2019.

[11] Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff. Minimizing the age of information through queues. *IEEE Transactions on Information Theory*, 65(8):5215–5232, 2019.

[12] Dimitri Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3 edition, 2005.

[13] Partha P. Bhattacharya and Anthony Ephremides. Optimal scheduling with strict deadlines. *IEEE Transactions on Automatic Control*, 34:721–728, July 1989.

[14] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[15] Giuseppe Bianchi and Ilenia Tinnirello. Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network. In *Proceedings of IEEE INFOCOM*, volume 2, pages 844–852, 2003.

[16] Bastian Bloessl, Michele Segata, Christoph Sommer, and Falko Dressler. Performance assessment of IEEE 802.11p with an open source SDR-based prototype. *IEEE Transactions on Mobile Computing*, 17(5):1162–1175, 2018.

[17] Byung-Jae Kwak, Nah-Oak Song, and Leonard E. Miller. Performance analysis of exponential backoff. *IEEE/ACM Transactions on Networking*, 13(2):343–355, 2005.

[18] He Chen, Yifan Gu, and Soung-Chang Liew. Age-of-information dependent random access for massive IoT networks. In *IEEE INFOCOM workshop on the Age of Information*, 2020.

[19] Kun Chen and Longbo Huang. Age-of-information in the presence of error. In *Proceedings of IEEE ISIT*, pages 2579–2583, 2016.

[20] Xingran Chen, Konstantinos Gatsis, Hamed Hassani, and Shirin Saeedi Bidokhti. Age of information in random access channels. In *Proceedings of IEEE ISIT*, 2020.

[21] Maice Costa, Marian Codreanu, and Anthony Ephremides. On the age of information in status update systems with packet management. *IEEE Trans. Inf. Theory*, 62(4):1897–1910, 2016.

[22] Antonio Franco, Emma Fitzgerald, Bjorn Landfeldt, Nikolaos Pappas, and Vangelis Angelakis. LUPMAC: a cross-layer MAC technique to improve the age of information over dense WLANs. In *Proceedings of IEEE ICT*, 2016.

[23] Robert G. Gallager. *Stochastic Processes: Theory for Applications*. Cambridge University Press, 2013.

[24] Anand Ganti, Eytan Modiano, and John N. Tsitsiklis. Optimal transmission scheduling in symmetric communication models with intermittent connectivity. *IEEE Trans. Inf. Theory*, 53(3), Mar. 2007.

[25] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed Bandit Allocation Indices*. Wiley, 2 edition, Mar. 2011.

[26] Sneihil Gopal and Sanjit Kaul. A game theoretic approach to DSRC and WiFi coexistence. In *Proceedings of IEEE INFOCOM workshop on the Age of Information*, 2018.

[27] Javier Gozalvez, Miguel Sepulcre, and Ramon Bauza. IEEE 802.11p vehicle to infrastructure communications in urban environments. *IEEE Communications Magazine*, 50(5):176–183, 2012.

[28] Yan Grunenberger, Martin Heusse, Franck Rousseau, and Andrzej Duda. Experience with an implementation of the idle sense wireless access method. In *Proceedings of ACM CoNEXT*, 2007.

[29] Xueying Guo, Rahul Singh, P.R. Kumar, and Zhisheng Niu. A high reliability asymptotic approach for packet inter-delivery time optimization in cyber-physical systems. In *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 197–206, 2015.

[30] Mor Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[31] Qing He, Di Yuan, and Anthony Ephremides. On optimal link scheduling with min-max peak age of information in wireless systems. In *Proceedings of IEEE ICC*, 2016.

[32] Qing He, Di Yuan, and Anthony Ephremides. Optimizing freshness of information: On minimum age link scheduling in wireless systems. In *Proceedings of IEEE WiOpt*, 2016.

[33] Martin Heusse, Franck Rousseau, Romaric Guillier, and Andrzej Duda. Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs. In *Proceedings of ACM SIGCOMM*, pages 121–132, 2005.

[34] I-Hong Hou, V. Borkar, and P. R. Kumar. A theory of QoS for wireless. In *Proceedings of IEEE INFOCOM*, pages 486–494, Apr. 2009.

[35] I-Hong Hou and P. R. Kumar. Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels. In *Proceedings of ACM MOBIHOC*, pages 175–184, May 2009.

[36] I-Hong Hou and P. R. Kumar. Scheduling heterogeneous real-time traffic over fading wireless channels. In *Proceedings of IEEE INFOCOM*, Mar. 2010.

[37] Yu-Pin Hsu. Age of information: Whittle index for scheduling stochastic arrivals. In *Proceedings of IEEE ISIT*, 2018.

[38] Yu-Pin Hsu, Eytan Modiano, and Lingjie Duan. Age of information: Design and analysis of optimal scheduling algorithms. In *Proceedings of IEEE ISIT*, 2017.

[39] Longbo Huang and Eytan Modiano. Optimizing age-of-information in a multi-class queueing system. In *Proceedings of IEEE ISIT*, 2015.

[40] National Instruments. LabVIEW communications 802.11 application framework 3.0. online: http://www.ni.com/manuals/, 2019.

[41] Zhiyuan Jiang, Bhaskar Krishnamachari, Sheng Zhou, and Zhisheng Niu. Can decentralized status update achieve universally near-optimal age-of-information in wireless multiaccess channels? In *Proceedings of IEEE ITC*, volume 1, pages 144–152, 2018.

[42] Changhee Joo and Atilla Eryilmaz. Wireless scheduling for information freshness and synchrony: Drift-based design and heavy-traffic analysis. In *Proceedings of IEEE WiOpt*, 2017.

[43] Igor Kadota, Andrea Baiocchi, and Alessandro Anzaloni. Kalman filtering: estimate of the numbers of active queues in an 802.11e EDCA WLAN. *Elsevier Computer Communications*, 2014.

[44] Igor Kadota and Eytan Modiano. Minimizing the age of information in wireless networks with stochastic arrivals. In *Proceedings of ACM MobiHoc*, 2019.

[45] Igor Kadota and Eytan Modiano. Minimizing the age of information in wireless networks with stochastic arrivals. *IEEE Trans. Mobile Comput.*, 2019.

[46] Igor Kadota and Eytan Modiano. Age of information in random access networks with stochastic arrivals. *[Under Review]*, 2020.

[47] Igor Kadota, Muhammad Shahir Rahman, and Eytan Modiano. Poster: Age of information in wireless networks: from theory to implementation. In *Proceedings of ACM MobiCom*, 2020.

[48] Igor Kadota, Muhammad Shahir Rahman, and Eytan Modiano. WiFresh: age-of-information from theory to implementation. *[Under Review]*, 2020.

[49] Igor Kadota, Abhishek Sinha, and Eytan Modiano. Optimizing age of information in wireless networks with throughput constraints. In *Proceedings of IEEE INFOCOM*, 2018.

[50] Igor Kadota, Abhishek Sinha, and Eytan Modiano. Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints. *IEEE/ACM Trans. Netw.*, 2018.

[51] Igor Kadota, Abhishek Sinha, Elif Uysal-Biyikoglu, Rahul Singh, and Eytan Modiano. Scheduling policies for minimizing age of information in broadcast wireless networks. *IEEE/ACM Trans. Netw.*, 2018.

[52] Igor Kadota, Elif Uysal-Biyikoglu, Rahul Singh, and Eytan Modiano. Minimizing the age of information in broadcast wireless networks. In *Proceedings of IEEE Allerton*, 2016.

[53] Clement Kam, Sastry Kompella, and Anthony Ephremides. Experimental evaluation of the age of information via emulation. In *Proceedings of IEEE MILCOM*, pages 1070–1075, 2015.

[54] Clement Kam, Sastry Kompella, Gam D. Nguyen, and Anthony Ephremides. Effect of message transmission path diversity on status age. *IEEE Trans. Inf. Theory*, 62:1360–1374, 2016.

[55] Clement Kam, Sastry Kompella, Gam D. Nguyen, Jeffrey E. Wieselthier, and Anthony Ephremides. Controlling the age of information: Buffer size, deadline, and packet replacement. In *Proceedings of IEEE MILCOM*, pages 301–306, 2016.

[56] Zerina Kapetanovic, Deepak Vasisht, Jongho Won, Ranveer Chandra, and Mark Kimball. Experiences deploying an always-on farm network. *ACM GetMobile: Mobile Computing and Communications*, 21(2):16–21, 2017.

[57] Sanjit Kaul, Marco Gruteser, Vinuth Rai, and John Kenney. Minimizing age of information in vehicular networks. In *Proceedings of IEEE SECON*, pages 350–358, 2011.

[58] Sanjit Kaul and Roy D. Yates. Status updates over unreliable multiaccess channels. In *Proceedings of IEEE ISIT*, 2017.

[59] Sanjit Kaul, Roy D. Yates, and Marco Gruteser. Real-time status: How often should one update? In *Proceedings of IEEE INFOCOM*, pages 2731–2735, 2012.

[60] Sanjit Kaul, Roy D. Yates, and Marco Gruteser. Status updates through queues. In *Proceedings of IEEE CISS*, 2012.

[61] John B. Kenney. Dedicated short-range communications (DSRC) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[62] Kyu Seob Kim, Chih-Ping Li, Igor Kadota, and Eytan Modiano. Optimal scheduling of real-time traffic in wireless networks with delayed feedback. In *Proceedings of IEEE Allerton*, pages 1143–1149, Oct. 2015.

[63] Martin Klapez, Carlo Augusto Grazia, and Maurizio Casoni. Application-level performance of IEEE 802.11p in safety-related V2X field trials. *IEEE Internet of Things Journal*, 7(5):3850–3860, 2020.

[64] Leonard Kleinrock and Fouad Tobagi. Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, 1975.

[65] Antzela Kosta, Nikolaos Pappas, and Vangelis Angelakis. Age of information: A new concept, metric, and tool. *Foundations and Trends in Networking*, 12(3):162–259, 2017.

[66] Antzela Kosta, Nikolaos Pappas, Anthony Ephremides, and Vangelis Angelakis. Age and value of information: Non-linear age case. In *Proceedings of IEEE ISIT*, 2017.

[67] Antzela Kosta, Nikolaos Pappas, Anthony Ephremides, and Vangelis Angelakis. Age of information performance of multiaccess strategies with packet management. *Journal of Communications and Networks*, 21(3):244–255, 2019.

[68] Kyung Sup Kwak, Sana Ullah, and Niamat Ullah. An overview of IEEE 802.15.6 standard. In *Proceedings of the IEEE ISABEL*, pages 1–6, 2010.

[69] Pierre-Yves Lajoie, Benjamin Ramtoula, Yun Chang, Luca Carlone, and Giovanni Beltrame. DOOR-SLAM: distributed, online, and outlier resilient slam for robotic teams. *IEEE Robotics and Automation Letters*, 5(2):1656–1663, 2020.

[70] Bin Li, Atilla Eryilmaz, and R. Srikant. Emulating round-robin in wireless networks. In *Proceedings of ACM MobiHoc*, 2017.

[71] Bin Li, Ruogu Li, and Atilla Eryilmaz. Throughput-optimal wireless scheduling with regulated inter-service times. *IEEE/ACM Trans. Netw.*, 23:1542–1552, 2015.

[72] Bin Li, Ruogu Li, and Atilla Eryilmaz. Wireless scheduling design for optimizing both service regularity and mean delay in heavy-traffic regimes. *IEEE/ACM Trans. Netw.*, 24:1867–1880, 2016.

[73] Keqin Liu and Qing Zhao. Indexability of restless bandit problems and optimality of Whittle index for dynamic multichannel access. *IEEE Trans. Inf. Theory*, 56:5547–5567, Nov. 2010.

[74] Ning Lu, Bo Ji, and Bin Li. Age-based scheduling: Improving data freshness for wireless real-time traffic. In *Proceedings of ACM MobiHoc*, 2018.

[75] María T. Lázaro, Lina M. Paz, Pedro Pinies, José A. Castellanos, and Giorgio Grisetti. Multi-robot SLAM using condensed measurements. In *Proceedings of IEEE/RSJ IROS*, pages 1069–1076, 2013.

[76] Ali Maatouk, Mohamad Assaad, and Anthony Ephremides. On the age of information in a CSMA environment. *IEEE/ACM Transactions on Networking*, 28(2):818–831, 2020.

[77] Nesrine Mahdoui, Vincent Frémont, and Enrico Natalizio. Communicating multi-UAV system for cooperative SLAM-based exploration. *Journal of Intelligent & Robotic Systems*, 98:325–343, 2020.

[78] Parisa Mansourifard, Tara Javidi, and Bhaskar Krishnamachari. Optimality of myopic policy for a class of monotone affine restless multi-armed bandits. In *Proceedings of IEEE CDC*, pages 877–882, Dec. 2012.

[79] Frank Mascarich, Huan Nguyen, Tung Dang, Shehryar Khattak, Christos Papachristos, and Kostas Alexis. A self-deployed multi-channel wireless communications system for subterranean robots. In *Proceedings of IEEE AeroConf*, 2020.

[80] Laetitia Matignon and Olivier Simonin. Multi-robot simultaneous coverage and mapping of complex scene - comparison of different strategies. In *Proceedings of ACM AAMAS*, page 559–567, 2018.

[81] Prathamesh Mayekar, Parimal Parag, and Himanshu Tyagi. Optimal lossless source codes for timely updates. In *Proceedings of IEEE ISIT*, 2018.

[82] David Mills, Jim Martin, Jack Burbank, and William Kasch. Network time protocol version 4: Protocol and algorithms specification. RFC 5905, 2010.

[83] Tobias Nägeli, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics*, 36(4), 2017.

[84] Elie Najm and Rajai Nasser. Age of information: The gamma awakening. In *Proceedings of IEEE ISIT*, pages 2574–2578, 2016.

[85] Michael J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[86] Gam D. Nguyen, Sastry Kompella, Clement Kam, Jeffrey E. Wieselthier, and Anthony Ephremides. Information freshness over an interference channel: A game theoretic view. In *Proceedings of IEEE INFOCOM*, 2018.

[87] N. Pappas, J. Gunnarsson, L. Kratz, M. Kountouris, and V. Angelakis. Age of information of multiple sources with queue management. In *Proceedings of IEEE ICC*, 2015.

[88] Riccardo Petrolo, Yingyan Lin, and Edward Knightly. ASTRO: Autonomous, sensing, and tetherless networked drones. In *Proceedings of ACM MobiSys – DroNet workshop*, 2018.

[89] Jasper Pickering. Take a look inside Alibaba's smart warehouse where robots do 70% of the work. online: https://www.businessinsider.com/inside-alibaba-smart-warehouse-robots-70-per-cent-work-technology-logistics-2017-9, 2017.

[90] Vivek Raghunathan, Vivek Borkar, Min Cao, and P. R. Kumar. Index policies for real-time multicast scheduling for wireless broadcast systems. In *Proceedings of IEEE INFOCOM*, Apr. 2008.

[91] Lawrence G. Roberts. ALOHA packet system with and without slots and capture. *ACM SIGCOMM Computer Communication Review*, 5(2):28–42, 1975.

[92] Tanya Shreedhar, Sanjit Kaul, and Roy D. Yates. An age control transport protocol for delivering fresh updates in the internet-of-things. In *Proceedings of IEEE WoWMoM*, 2019.

[93] Rahul Singh, Xueying Guo, and P.R. Kumar. Index policies for optimal mean-variance trade-off of inter-delivery times in real-time sensor networks. In *Proceedings of IEEE INFOCOM*, pages 505–512, Jan. 2015.

[94] Rahul Singh and Alexander Stolyar. Maxweight scheduling: "smoothness" of the service process. In *Proceedings of IEEE INFOCOM*, 2016.

[95] Canberk Sönmez, Sajjad Baghaee, Abdussamed Ergişi, and Elif Uysal-Biyikoglu. Age-of-information in practice: Status age measured over TCP/IP connections through WiFi, ethernet and LTE. In *Proceedings of IEEE BlackSeaCom*, 2018.

[96] Dietrich Stoyan. *Comparison Methods for Queues and Other Stochastic Models*. Wiley Series in Probability and Statistics. Wiley, 1983.

[97] Yin Sun. The ongoing history of the age of information. online: http://webhome. auburn.edu/ yzs0078/, 2019.

[98] Yin Sun, Igor Kadota, Rajat Talak, and Eytan Modiano. *Age of Information: A New Metric for Information Freshness*. Morgan & Claypool, 2019.

[99] Yin Sun, Elif Uysal-Biyikoglu, and Sastry Kompella. Age-optimal updates of multiple information flows. In *IEEE INFOCOM workshop on the Age of Information*, 2018.

[100] Yin Sun, Elif Uysal-Biyikoglu, Roy D. Yates, C. Emre Koksal, and Ness B. Shroff. Update or wait: How to keep your data fresh. *IEEE Trans. Inf. Theory*, 2017.

[101] Rajat Talak, Igor Kadota, Sertac Karaman, and Eytan Modiano. Scheduling policies for age minimization in wireless networks with unknown channel state. In *Proceedings of IEEE ISIT*, 2018.

[102] Rajat Talak, Sertac Karaman, and Eytan Modiano. Minimizing age-of-information in multi-hop wireless networks. In *Proceedings of IEEE Allerton*, 2017.

[103] Rajat Talak, Sertac Karaman, and Eytan Modiano. Distributed scheduling algorithms for optimizing information freshness in wireless networks. In *Proceedings of IEEE SPAWC*, 2018.

[104] Rajat Talak, Sertac Karaman, and Eytan Modiano. Optimizing information freshness in wireless networks under general interference constraints. In *Proceedings of ACM MobiHoc*, 2018.

[105] Vishrant Tripathi and Sharayu Moharir. Age of information in multi-source systems. In *Proceedings of IEEE Globecom*, 2017.

[106] Pablo Urcola, María T. Lázaro, José A. Castellanos, and Luis Montano. Cooperative minimum expected length planning for robot formations in stochastic maps. *Robotics and Autonomous Systems*, 87:38–50, 2017.

[107] Pablo Valerio. Amazon robotics: IoT in the warehouse. online: https://www.informationweek.com/strategic-cio/amazon-robotics-iot-in-the-warehouse/d/d-id/1322366, 2015.

[108] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. Farmbeats: An IoT platform for data-driven agriculture. In *Proceedings of NSDI*, pages 515–529, 2017.

[109] Richard R. Weber and Gideon Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990.

[110] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988.

[111] Peter R. Wurman, Raffaello D'Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. In *Proceedings of IAAI*, pages 1752–1759, 2007.

[112] Yuanzhang Xiao and Yin Sun. A dynamic jamming game for real-time status updates. In *Proceedings of IEEE INFOCOM workshop on the Age of Information*, 2018.

[113] Roy D. Yates. Lazy is timely: Status updates by an energy harvesting source. In *Proceedings of IEEE ISIT*, pages 3008–3012, June 2015.

[114] Roy D. Yates, Philippe Ciblat, Aylin Yener, and Michele Wigger. Age-optimal constrained cache updating. In *Proceedings of IEEE ISIT*, 2017.

[115] Roy D. Yates and Sanjit Kaul. Real-time status updating: Multiple sources. In *Proceedings of IEEE ISIT*, 2012.

[116] Roy D. Yates, Elie Najm, Emina Soljanin, and Jing Zhong. Timely updates over an erasure channel. In *Proceedings of IEEE ISIT*, 2017.

[117] Se-Young Yun, Yung Yi, Jinwoo Shin, and Do Young Eun. Optimal CSMA: A survey. In *Proceedings of ICCS*, pages 199–204, 2012.

[118] Xu Zheng, Zhipeng Cai, Jianzhong Li, and Hong Gao. Scheduling flows with multiple service frequency constraints. *IEEE Internet of Things Journal*, 4:496–504, 2017.