IMPLEMENTATION AND EVALUATION OF AGE-AWARE DOWNLINK SCHEDULING POLICIES IN COMMUNICATION NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TAHİR KEREM OĞUZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2021

Approval of the thesis:

IMPLEMENTATION AND EVALUATION OF AGE-AWARE DOWNLINK SCHEDULING POLICIES IN COMMUNICATION NETWORKS

submitted by **TAHİR KEREM OĞUZ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Elif Uysal
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Engin Tuncer
Electrical and Electronics Engineering, METU _____

Prof. Dr. Elif Uysal
Electrical and Electronics Engineering, METU _____

Prof. Dr. Ali Özgür Yılmaz
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Tolga Girici
Electrical and Electronics Engineering, TOBB ETU _____

Assist. Prof. Dr. Elif Tuğçe Ceran Arslan
Electrical and Electronics Engineering, METU _____

Date: 09.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: TAHİR KEREM OĞUZ

Signature :

ABSTRACT

IMPLEMENTATION AND EVALUATION OF AGE-AWARE DOWNLINK SCHEDULING POLICIES IN COMMUNICATION NETWORKS

OĞUZ, TAHİR KEREM
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Elif Uysal

September 2021, 96 pages

As communication systems evolve, advanced perspectives are needed to meet the requirements of emerging applications. The Age of Information metric has taken its place in the literature to examine the information freshness demands of various applications. With the Age of Information perspective, the freshness of information subject to communication is monitored from the destination's point of view. Within the scope of the thesis, a wireless network with a single transmitter (base station) and multiple receivers was realized using Software Defined Radios (SDRs). The Information Age metric was considered the main performance metric of the network. Low-complexity, age-sensitive scheduling policies were applied and compared with traditional scheduling policies. Then, the Query Age of Information (QAoI) metric, a version of the Age of Information concept suitable for pull-based scenarios, was examined. We modified Age-Aware policies to perform well with respect to QAoI. We modified the SDR testbed to study the performance of pull-based scheduling methods experimentally.

Keywords: Age of Information, Wireless Networks, Software Defined Radios

ÖZ

HABERLEŞME AĞLARINDAKİ BİLGİ-YAŞI FARKINDA ÇİZELGELEME POLİÇELERİNİN GERÇEKLENMESİ VE İNCELENMESİ

OĞUZ, TAHİR KEREM
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Elif Uysal

Eylül 2021 , 96 sayfa

İletişim sistemleri geliştikçe, ortaya çıkan uygulamaların gereksinimlerini karşılamak için gelişmiş perspektiflere ihtiyaç duyulmaktadır. "Bilgi Yaşı" metriği, uygulamaların bilgi tazeliği taleplerini incelemek için literatürde kendine yer edinmiştir. Bilgi Yaşı perspektifi ile, iletişime konu olan bilgilerin tazeliği, bilginin alıcısının gözünden takip edilmektedir. Tez kapsamında, Yazılım Tanımlı Telsizler (SDR'ler) kullanılarak tek verici (baz istasyonu) ve birden fazla alıcı içeren bir kablosuz ağ gerçekleştirilmiştir. Bilgi Yaşı metriği, ağın ana performans metriği olarak kabul edilerek, karmaşıklığı düşük, yaşa duyarlı çizelgeleme politikaları uygulanmış ve geleneksel çizelgeleme politikalarıyla karşılaştırılmıştır. Ardından, Bilgi Yaşı kavramının, bilgi çekme isteği odaklı senaryolara uygun bir versiyonu olan "Sorgulanan Bilgi Yaşı" (QAoI) metriği incelenmiştir. Bilgi Yaşı merkezli çizelgeleme algoritmaları modifiye edilerek, QAoI konseptinde iyi performans gösterebilecek çizelgeleme politikaları elde edildi. SDR'lar ile oluşturulan test ortamı, bilgi çekme isteği odaklı çizelgeleme politikalarının performansının deneysel olarak incelenebilmesi amacıyla geliştirildi.

Anahtar Kelimeler: Bilgi Yaşı, Kablosuz Ağlar, Yazılım Tabanlı Telsizler

To my dear family

ACKNOWLEDGMENTS

I would like to express my sincere appreciation and gratitude to my supervisor Prof. Elif Uysal for her continuous support and invaluable guidance throughout my thesis study.

I would like to express my gratitude to the Assoc. Prof. Tolga Girici and Assist. Prof. Elif Tuğçe Ceran Aslan for their guidance and insightful suggestions.

I would like to thank Prof. T. Engin Tuncer and Prof. Ali Özgür Yılmaz for for their valuable comments and critics about my thesis study.

I would like to express my gratitude to the fellow members of the METU Research Group on Communication Networks. I would like to thank Hasan Burhan Beytur and Sajjad Baghaee from the CNG group for their valuable support.

I would like to thank Aselsan Inc. for supporting my graduate study.

I would like to thank my friends Hüsrev Cilasun, Emin Yüksel and İbrahim İşler for their valuable support.

I would like to offer my special thanks to my wife and my family for their unwavering support.

All praise and gratitude are due to Allah, by whose favor good deeds are accomplished.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 The Motivation of the Thesis

As communication systems evolve, new perspectives are needed to meet the demands of various applications. The traditional approaches to emerging technologies may be inadequate to solve the encountered problems. For example, along with the high amount of data transfer requirements, the Internet of Things, Cyber-Physical Systems, Machine-Type communications, and Vehicular networks also require the freshness of the shared information. In this example, the freshness property gives an intuition about the value of the information in terms of usefulness. However, the traditional metrics are unable to capture the usefulness of the transferred information. The emerging concept of semantic-based reinterpretation to communication systems can offer new solutions for defining the "right piece of the information" that is valuable to transfer [29].

Metrics that prioritize the semantics of information center the significance and relevance of the information sent [29]. Among the semantic metrics, the Age of Information values the information concerning its freshness. The Age of Information (AoI) at the receiver of a status update flow is defined as the time elapsed since the generation of the newest status update that has been received by the destination [15]. AoI is gaining traction as a key performance indicator for Machine-Type Communications (MTC). This is owed to timely update requirements in many real-time and remote monitoring-based applications in growing networking paradigms such as the Internet of Things, Vehicular Networks, Cyber-Physical Systems.

Similar to the Age of Information, Query-Age of Information is another member

of the Semantic Metrics family, prioritizing the information's "freshness" property. However, it differentiates from the Age of Information metric since, in the Query-Age of Information, the freshness is important only at the query instants [4, 16, 33]. These queries are generated by the query source which can be a user or an application that demands the latest value of the tracked information. In contrast, the Age of Information measures the freshness of the information with giving equal importance to every time instant.

The main motivation of this thesis is to analyze the traditional networks with a new look provided by the arising semantic metrics named Age of Information and the Query age of Information. To have a deep understanding of these metrics, we implemented a network with Software Defined Radios and tried to observe the effects of real-life channel imperfections on the theoretically expected results.

1.2 Contributions

The main contribution of this thesis is to report one of the first-ever experimental studies of Age-aware MAC Layer scheduling policies. We implemented a multiuser downlink network with a single base station and multiple receivers using SDRs. This implementation allowed us to modify the scheduling policies and study their performance in real-life environment scenarios. We implemented the Age-aware and query-aware scheduling policies along with the other well-known policies and evaluated the results in the network in terms of Age of Information and Query-Age of Information.

Besides setting up the testbed, we have made the following contributions:

Inspired by [13], we modified the traditional Max-weight Policy for the Effective AoI system model. The resulting Query-Aware Max-Weight policy has a similar QAoI performance with the Whittle's Index. Besides, we observed that Max Weight Policy might yield higher throughput than Whittle's Index Policy in some cases.

We modified Whittle's Index policy to give the Scheduler the ability to induce higher reward to the successful transmissions. The resulting Policy is similar to the WIP Policy in [12] and enables the Scheduler to adjust the throughput via an additional

penalty parameter inserted into the cost function. Although it is possible to increase the throughput as a result of the modification, this increase also brings about a reduction in AoI performance.

We propose a lower bound for the proactive serving scenario of the Effective-AoI case. The lower bound is important for practical works since the channel reliabilities are unstable across experiments. The lower bound can be used as a benchmark to scale the results of the experiments conducted in varying channel conditions.

We propose a weight function for the Max-Weight Policy to obtain a policy for the Query-AoI system model. We tested the Policy and shared the results.

1.3 The Outline of the Thesis

In the second chapter, we present detailed information about the Age of Information concept. Then, we explain the system model that we examined. We give detailed information about the scheduling policies and share the results of the implementation. We extend the research we have done in [23].

In the third chapter, we introduce the Effective Age of Information and Query Age of Information concepts. We present the scheduling policies and share the results of the experiments conducted in the SDR testbed.

In the fourth chapter, we describe our implementation environment, which consists of Software Defined Radios. We share detailed information about our SDR testbed.

In the last chapter, we summarize the thesis, discuss the results and share our vision for future work.

CHAPTER 2

AGE OF INFORMATION

2.1 Literature Review

The Age of Information (AoI) at the receiver of a status update flow is defined as the time elapsed since the generation of the newest status update that has been received by the destination [15]. AoI is gaining traction as a key performance indicator for Machine-Type Communications (MTC). This is owed to timely update requirements in many real-time and remote monitoring-based applications in growing networking paradigms such as the Internet of Things, Vehicular Networks and Cyber-Physical Systems.

It follows from the definition above that AoI tracks the freshness of an entire flow of information from the receiver's point of view. It thus exhibits significantly different behavior than delay, which is measured per packet. Notably, a low average delay value does not imply a low average age [26, 32]. Age is a composite measure that is related to both throughput and delay. As a result, its consideration as the objective of a multiuser scheduling problem introduces a novel formulation. In [26], the authors point out that the zero-wait policy may not be age-optimal in the presence of FCFS queues.

To understand the concept of the Age of Information, we can consider a simple vehicular network as an example. In this network, we assume that a central processing system collects the necessary information for each vehicle. This information can include the status of the road or locations and velocities of other vehicles. Vehicles use these pieces of information for autonomous driving. These pieces of information vary with time. The central processing system generates an information packet for each

vehicle. Then, a common base station tries to send these packets to the vehicles. The base station can send only one packet at each time slot.

If we examine the problem from the perspective of an autonomous vehicle, we can notice that stale information is of no use in the driver assistant's decisions. As the freshness of the information obtained by the driver assistant increases, uncertainties on information such as road conditions or the position of other vehicles are reduced. In this way, it may be possible for the driver assistant to make more accurate decisions. Therefore, the base station must implement a scheduling policy to keep the vehicle's information as fresh as possible.

There have been several studies in the area of multiuser scheduling for minimum Age of Information. In [8], a scheduling problem with the objective of age minimization is revealed as an NP-hard problem. In [14], age-optimal scheduling principles were developed for a lossy multiuser channel, and optimality conditions for a Greedy scheduler were found. In [13], a Whittle's Index Policy was developed and contrasted with a Max-Weight Policy.

In the multiuser scheduling problem, the timing of the packets generated by the sources to be sent to the receivers has serious impacts on the AoI metric. In the active source system model, the sources generate a new packet at each time slot. In this model, each source has its own queue. These queues are managed with the LCFS policy. Moreover, the capacity of this LCFS queue is limited to one packet. Therefore, only the most up-to-date packet is available in the queues. In this model, it is assumed that packets are generated deterministically at each time slot. However, in real systems, there may be situations where the packet generation process is stochastic. In [10], a case is discussed where the packet generation of the sources is Bernoulli distributed random variable. Then, the paper examines the effect of different queuing policies applied to the generated packets from the AoI perspective.

While most AoI related studies reported to date have been theoretical, there are also implementation-based studies such as [1,2,7,11,23,25,27]. In [25,27], the effects of different wireless access technologies on end-to-end TCP/IP connections were measured. In the studies [1, 7, 11] novel age-based MAC layer algorithms, including a Max-Weight age policy, were tested on software-defined radio platforms (USRPs). A

simple estimator was utilized to compute the channel reliability during runtime. The work in [1] experiments on the effects of packet management policies on the performance of networked control systems. Lastly, [7] developed a testbed to evaluate various ALOHA-like random access protocols in terms of AoI. In [23], we implemented and evaluated Age-aware downlink scheduling policies in SDR testbed.

2.2 AoI at the Multiuser Downlink Channel

We study a wireless downlink where multiple receiver modules try to track time-sensitive information in the form of status update packet flows, sent to them by a common access point or base station, over orthogonal channels. System time is divided into fixed-length frames. In each frame, the base station is allowed to activate only one link, transmitting to a single receiver. Downlink channels are unreliable such that for each link, there is a constant probability that the packet reception will not be successful.

The packet generation model is that one that was considered in [14]: At the beginning of each frame, fresh packets for each flow are assumed to arrive at the BS, ready for possible transmission. One of those will be selected for transmission, and all others will be discarded. If a successful transmission happens, the base station is informed over an error-free channel in the same frame.

In the network, the total number of receivers is denoted as M. We consider a finite-horizon problem of K frames in duration. In the activated link, the probability of successful transmission (i.e., reliability) is p_i, where $i \in 1, \ldots, M$ is the receiver's index.

The $a_i(k)$ indicates the decision of the base station in frame k. If the receiver i is selected for transmission in frame k, then $a_i(k)$ is equal to 1. Otherwise, $a_i(k)$ is equal to 0.

$$a_i(k) = \begin{cases} 1 & \text{if the receiver } i \text{ is Selected} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

Similarly, the $c_i(k)$ is an indicator for the channel state in each frame. If the channel is ON, then the successful transmission can be made to i'th user in k'th frame; therefore $c_i(k)$ is equal to 1. Otherwise, if the successful transmission is unavailable, $c_i(k)$ is equal to 0. In the system model, we consider $c_i(k)$ as a Bernoulli distributed random variable.

$$c_i(k) = \begin{cases} 1 & \text{if the Channel is ON} \\ 0 & \text{Channel is OFF} \end{cases} \tag{2.2}$$

To have a successful transmission for the receiver i in frame k, both $a_i(k)$ and $c_i(k)$ must equal to 1. $u_i(k)$ gives the result of the transmission to user i in frame k. Calculation of $u_i(k)$ is given in (2.3).

$$u_i(k) = \begin{cases} 1 & \text{if } c_i(k)a_i(k) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

The instantaneous Age of Information for flow i (i.e. receiver i) at the beginning of the k^{th} frame is $\Delta_i(k)$. The Figure 2.1 shows an example $\Delta_i(k)$ evaluation through frames. In the figure, black arrows indicate the successful packet arrivals. As there is no buffering in the base station and new packets arrive at the BS at the beginning of each frame, the age of a successfully received packet is always unity. Thus, $\Delta_i(k)$ drops to 1 after successful reception. If receiver i was not selected or did not successfully decode a packet, $\Delta_i(k)$ increases by 1 at the end of the frame. Evaluation of $\Delta_i(k)$ is given in (2.4).

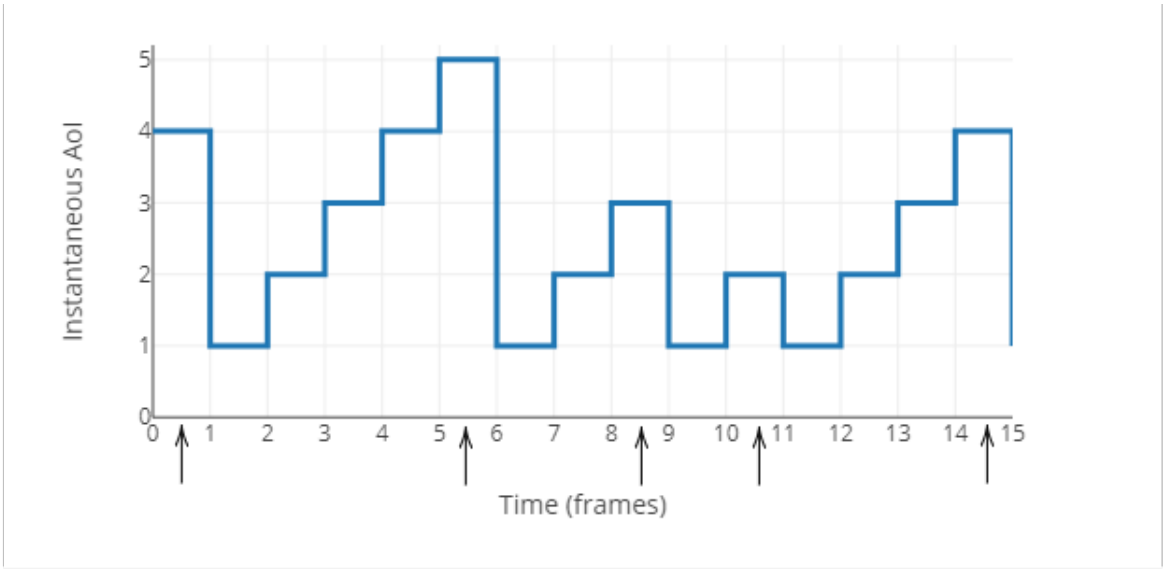

Figure 2.1: Evaluation of the Instantaneous AoI through frames

$$\Delta_i(k+1) = \begin{cases} 1 & \text{if } u_i(k) = 1 \\ \Delta_i(k) + 1 & \text{otherwise} \end{cases} \tag{2.4}$$

The asymmetry among the channel qualities of the receivers is an important statistic that may significantly affect the AoI of the network. This statistic is captured by the Coefficient of Variation C_V. High C_V indicates that the channel qualities highly differ among receivers. The Calculation of the C_V is given in (2.8). As [13] states that when the C_V increases, the performance of some scheduling policies (e.g., Greedy) may suffer.

Inter-arrival times between "ON" states of the channel are used to calculate the Coefficient of Variation. Since we assumed the channel status is independent Bernoulli-Distributed random variable, the corresponding inter-arrivals are i.i.d. random variables with geometric distribution. The mean value of the inter-arrivals can be calculated as the mean value of a geometric random variable with a success rate p_i. Inter-arrival times between the "ON" states of the channel are described as $\hat{I}_i^c$ where

i denotes the receiver's index.

$$\overline{\mathbb{M}}\left[\hat{I}_i^c\right] = \frac{1}{p_i} \tag{2.5}$$

To calculate C_V, we define a sample mean and the sample variance of the $\hat{I}_i^c$'s in the network at (2.6) and (2.7).

$$\overline{\mathbb{M}}\left[\hat{I}_i^c\right] = \frac{1}{M}\sum_{j=1}^{M}\hat{I}_i^c \tag{2.6}$$

$$\overline{\mathbb{V}}\left[\hat{I}_i^c\right] = \frac{1}{M}\sum_{j=1}^{M}\left(\hat{I}_i^c - \overline{\mathbb{M}}\left[\hat{I}_i^c\right]\right)^2 \tag{2.7}$$

Then, the definition of the C_V is given in equation (2.8).

$$C_V = \frac{\sqrt{\overline{\mathbb{V}}\left[\hat{I}_i^c\right]}}{\overline{\mathbb{M}}\left[\hat{I}_i^c\right]} \tag{2.8}$$

2.2.1 The Optimization Problem

The performance indicator for each receiver is the Age of Information of its corresponding packet flow, which increases linearly in time in between packet receptions and is updated at the time of successful reception of a new packet.

$$AoI_{avg} = \frac{1}{2} + \frac{1}{KM}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_i(k)\right] \tag{2.9}$$

Throughout the thesis, we assume that AoI changes in a discretized fashion and only changes at the beginning of the frame. The term $\frac{1}{2}$ is necessary only if the AoI continues to increase within the frame.

The utility function for the AoI, $J_A(\pi)$, which adopts the scheduling policy π over the K frames and M receivers, is calculated by omitting the constant term in the (2.9).

Throughout the thesis, we omit the $\frac{1}{2}$ term in AoI_{avg} and use $J_A(\pi)$ to denote the Average AoI of the network.

$$J_A(\pi) = \frac{1}{KM}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_i(k)\right] \tag{2.10}$$

The objective of the optimization is to find a scheduling policy π that minimizes the utility function.

$$\min_{\pi\in\Pi}\mathbb{E}\left[J_A(\pi)\right], \text{ where } J_A(\pi) = \frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_i(k) \tag{2.11}$$

2.2.2 Lower Bound for the Optimization Problem

In the literature, [13] and [3] proposed a lower bound for the system model described in Section 2.2. The proposed lower bound by the [13] is given in Equation (2.12), and the lower bound proposed by [3] is given in (2.13) . However, throughout this section, we show that the lower bound in Equation (2.12) becomes loose under certain circumstances. We observe that the lower bound in (2.13) is more strict than the (2.12). Therefore, we will use the lower bound in (2.13) as a benchmark.

While strictly speaking, this lower bound is not for finite K, and also not valid when p_i can change in time, yet, need a benchmark to scale $J_A(\pi)$. Note that $J_A(\pi)$ could be used for direct comparison of scheduling policies when channel reliabilities are constant across channels. However, when there are channels with differing reliabilities, which will become important as output power drops, it may be best to measure the relative performances of algorithms to a benchmark. Hence, we will use AoI_{Norm} ((2.14)), which is the ratio of $J_A(\pi)$ to $L_B(p_1, p_2, ..p_M)$, as the performance metric.

$$\lim_{K\to\infty} J_A(\pi) \geq \frac{1}{2M}\left(\sum_{i=1}^{M}\sqrt{\frac{1}{p_i}}\right)^2 + \frac{1}{2} \quad w \cdot p.1 \tag{2.12}$$

$$\lim_{K\to\infty} J_A(\pi) \geq \frac{1}{2M}\left(\sum_{j=1}^{M}\sqrt{\frac{1}{p_j}}\right)^2 + \frac{1}{2M}\left(\frac{1-p_j}{p_j}\right) + \frac{1}{2} \quad \text{and } j^* \triangleq \arg\min_j \frac{1-p_j}{p_j} \tag{2.13}$$

$$AoI_{Norm} = \frac{J_A(\pi)}{L_B(p_1, p_2, ..p_M)} \tag{2.14}$$

The derivation of the lower bound in the Equation 2.12is obtained by rewriting the $\Delta_i(k)$ in terms of inter-deliveries. In one of the steps for derivation of this lower bound, $J_A(\pi)$ is obtained in terms of variance $\mathbb{V}[I_i]$ and the mean $\mathbb{M}[I_i]$ of the inter-arrivals (2.15).

$$\lim_{K\to\infty} J_A(\pi) = \frac{1}{2M}\sum_{i=1}^{M}\left[\frac{\bar{V}[I_i]}{\overline{\mathbb{M}}[I_i]} + \overline{\mathbb{M}}[I_i] + 1\right] \tag{2.15}$$

This equation gives an intuitive understanding of the underlying mechanism for minimizing the AoI. To minimize the AoI, **frequent** update packets must arrive **regularly** [13]. The $\frac{\bar{V}[I_i]}{\overline{\mathbb{M}}[I_i]}$ term, which can be described as **Variance to Mean Ratio(VMR)** of the inter-arrivals, indicates the **regularity**, the variation on the inter-arrival times of the packets must be minimized. The $\mathbb{M}[I_i]$ is aligned with the **frequent** term. If the updates occur more frequently, then the average inter-delivery time decrease.

In the next derivation step, the author of [13] removes the VMR term and turns the equation into inequality using the variance of inter-arrivals, $\bar{V}[I_i]$, is greater than or equal to 0.

$$\lim_{K\to\infty} J_A(\pi) \geq \frac{1}{2M}(\sum_{i=1}^{M}\overline{\mathbb{M}}[I_i]) + \frac{1}{2} \tag{2.16}$$

However, in our work, we observed that the removal of the VMR term reduces the tightness of the bound, especially in the cases where the number of the receivers are limited, and receivers with low p_i are present. Fig. 2.2 shows the growth of the VMR in symmetric networks when the channel reliabilities decrease. The effect of increasing the number of receivers in the network to the equation (2.16) is shown in Fig. 2.3. The figure shows the value of the VMR in the network as p_i of the receivers increased. The number of receivers, M, was 3, and each receiver had the same p_i, i.e.,

the network was symmetric. The p_i values were increased by 0.02 in each step, and in each step, the frame length K was 10000000. Greedy PolicyPolicy was used for scheduling decisions. As the channel reliabilities deteriorated, the value of the VMR increased significantly.

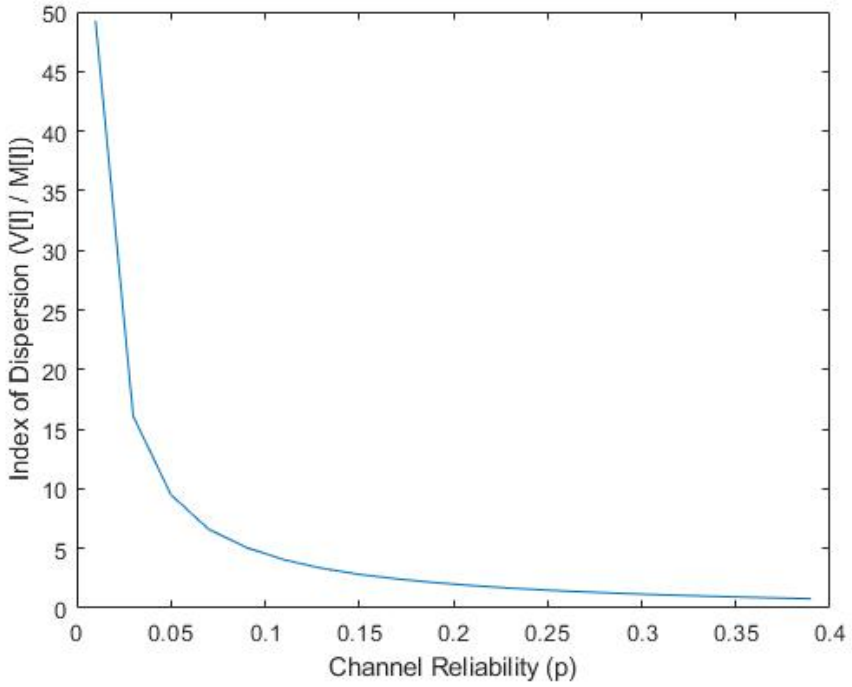

Figure 2.2: Evaluation of VMR with varying p_i (MATLAB Simulation)

The author of [13] points out that the performance of the Greedy Policy converges to the lower bound in the symmetric networks when the M is increased. However, we argue that this observation results from the looseness of the lower bound, rather than the performance of the Greedy. In the case of low M, since the lower bound becomes looser due to the phenomenon that can be seen in Fig. 2.3, the actual performance of the Greedy Policy was not observable. Fig. 2.3 shows the evaluation of the VMR and the result of the utilization function given in (2.15) as the number of the receivers M increased. When the M increases, the value of the $\frac{\bar{V}[I_i]}{\overline{\mathbb{M}}[I_i]}$ term in the (2.15) tends to stay unchanged. However, since the $\overline{\mathbb{M}}[I_i]$ increases with the M, the portion of $\frac{\bar{V}[I_i]}{\overline{\mathbb{M}}[I_i]}$ in the Eq.(2.15) decreases. Therefore, the effect of the $\frac{\bar{V}[I_i]}{\overline{\mathbb{M}}[I_i]}$ in the Equation(2.15) decreases when the M increases.

To overcome the looseness observed in the lower bound derived in [13], another lower

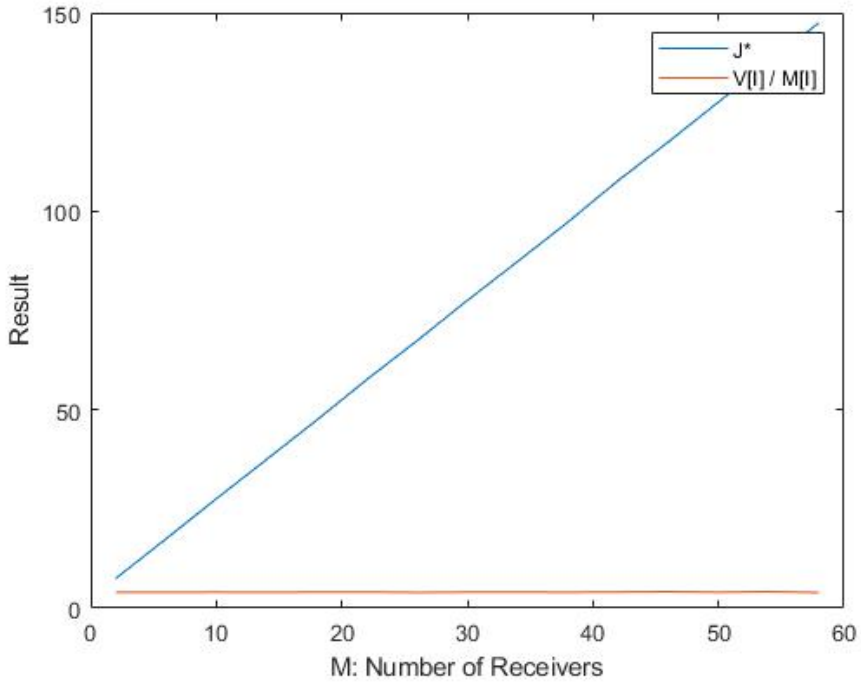

Figure 2.3: Evaluation of the Utility Function with varying M (MATLAB Simulation)

bound from the literature is investigated. The authors of [3] also proposed lower bound for the similar problem and both lower bounds in [13] and [3] are very similar except [3] has an additional $\frac{1-p_j^*}{p_j^*}$ term. We adapted the lower bound in [3] into the our system model and the result is given in the Equation (2.13).

We argue that the additional term from [3] aligns with the VMR and fills the gap that occurred by neglecting the VMR in the Equation (2.15).

To gain an insight about the additional term, we investigate a trivial system model with only one receiver and a base station. The optimal scheduling policy for this system model is choosing $a_i(k) = 1$ at all frames. Therefore, the base station always selects the only available receiver and $u_i(k) = c_i(k)$ at every frame k. The successful transmission happens at every frame where the $c_i(k) = 1$. Then, we can calculate the Variance to Mean Ratio for this system model when the optimal policy is utilized. The variance to mean ratio(VMR) of the inter-deliveries between possible transmissions for the receiver i is denoted as $\frac{\mathbb{V}[I_i^\star]}{\mathbb{M}[I_i^\star]}$. Since the channel status is an independent random variable with Bernoulli distribution, the corresponding inter-delivery time

is also a random variable with Geometric distribution. We use $I_i^{c\star}[n]$ notation to describe the n'th inter-delivery to the receiver i under $a_i(k) = c_i(k)$ assumption at every frame. The mean value and the variance of a Geometric distribution is well defined in the literature and we can easily define $\mathbb{V}\left[I_i^{c\star}\right] = \frac{1-p_i}{p_i^2}$ and $\mathbb{M}\left[I_i^{c\star}\right] = \frac{1}{p_i}$. Then, we calculate VMR in Equation (2.17) by using the mean value and the variance of $I_i^{c\star}[n]$,

$$\begin{aligned} \frac{\mathbb{V}\left[I_i^{c\star}\right]}{\mathbb{M}\left[I_i^{c\star}\right]} &= \frac{\frac{1-p_i}{p_i^2}}{\frac{1}{p_i}} \\ \frac{\mathbb{V}\left[I_i^{c\star}\right]}{\mathbb{M}\left[I_i^{c\star}\right]} &= \frac{1-p_i}{p_i} \end{aligned} \tag{2.17}$$

In this trivial system model, we can change the scheduling policy to make the inter-delivery times less varying. This would decrease VMR, but the resulting AoI performance would decrease since we would waste update attempts. This result shows that proposing a lower bound to VMR without considering $\mathbb{M}\left[I_i^{c\star}\right]$ causes that bound to become loose. Moreover, the result of VMR in the Equation (2.17) overlaps with the additional term from the lower bound of [3] given in the Equation (2.13). We comment that, in the multiple receiver case, the average VMR is bounded by the $\frac{\mathbb{V}\left[I_i^{c\star}\right]}{\mathbb{M}\left[I_i^{c\star}\right]}$. This comment is described in Equation (2.18).

$$\sum_{i=1}^{M} \frac{\mathbb{V}\left[I_i\right]}{\mathbb{M}\left[I_i\right]} \geq \frac{1-p_{j^*}}{p_{j^*}} \text{ where } j^* \triangleq \arg\min_j \frac{1-p_{j^*}}{p_{j^*}} \tag{2.18}$$

2.2.3 Scheduling Policies

We implement the Round Robin, Greedy, Whittle's Index, and Max-Weight Policies defined in [13] for the link activation decision. Both policies can operate without requiring high computational resources.

In Round-Robin Policy, neither ages of receivers nor channel reliabilities are taken into account. Links are activated sequentially, one in each frame. The only information used by the base station is the former activation decisions.

The Greedy Policy takes the results of the former transmissions into account and

tracks the AoI of the receivers. Then, selects the flow with the highest age in each frame, ignoring channel reliabilities. Note that it will keep picking the receiver with the largest AoI again and again until there is a successful reception. In the case of a link with low reliability, this phenomenon will cause the whole network to go into starvation.

Whittle's Index and Max-Weight Policies defined in [14] take care of the starvation problem by combining the ages of flows with the reliability of the corresponding links, summarized by a set of weights or indices $\{C_i\}$. In each frame, the receiver with the highest C_i will be selected for transmission [13]. The computation of this value for the Whittle Index Policy and the Max-Weight Policy are given, respectively, in (2.19) and (2.20).

$$C_i\left(\Delta_i(k)\right) = p_i\Delta_i(k)\left[\Delta_i(k) + \frac{2-p_i}{p_i}\right] \tag{2.19}$$

$$C_i\left(\Delta_i(k)\right) = p_i\Delta_i(k)(\Delta_i(k)+2) \tag{2.20}$$

2.3 Experiment Results

In this section, we share the results of the experiments conducted in the SDR network. The detailed information about the experimental setup is given in 4.

To test scheduling policies in different system configurations, we altered the channel reliabilities of the receivers. The power of the broadcasted signal from the base station, the signal gain of the receivers, and the distance between the receiver and the base station were the factors that caused the channel status of the receivers to vary. Since the gain of the receivers and the Output Power Gain of the base station are variables that we can manipulate, we altered these variables to create channel variations among experiments. The output power gain of the transmitter USRP is configured using the Labview software.

When the gain increases, stronger signal will be sent to the receivers and the probability of successful transmission of the packets will increase. Moreover, as the output

gain of the base station increases, the receiver further away from the base station is more likely to receive the update packet successfully, and asymmetry between the channels will be reduced. Similarly, we also use the Input Gain is also for modifying the channel reliability. The Input Gain of the receiver is directly proportional to the channel reliability. Increasing the Input Gain decreases the error probability and therefore increases the channel reliability.

In the experiments, evaluated scheduling policies run at least five times at each power gain level. The channel statistics corresponding to the experiment sets are given in Table 2.1 and Table 2.2.

The results of the experiments are given in terms of average AoI J_a, Normalized AoI AoI_{Norm}, and throughput. Calculation of the Average AoI is given in Equation (2.10), and calculation of Normalized AoI is given in Equation (2.14). Throughput is calculated as the ratio between total packets sent and the total packets obtained by all receivers. The channel statistics observed in the experiments are also shared along with the experiment.

A normalized performance indicator is equal to 1 means the AoI of the system reaches the theoretical lower bound. Therefore, as the normalized performance indicator approaches one, the scheduling policy approaches the optimal performance.

2.3.1 Increasing the Gain of an Individual Receiver

In the first experiment set, each of the 4 policies was tested ten times at each of the 4 plotted values of receiver gain level, and the two results were averaged. In each experiment, the frame length was $K = 7500$, and there were $M = 3$ receiver.

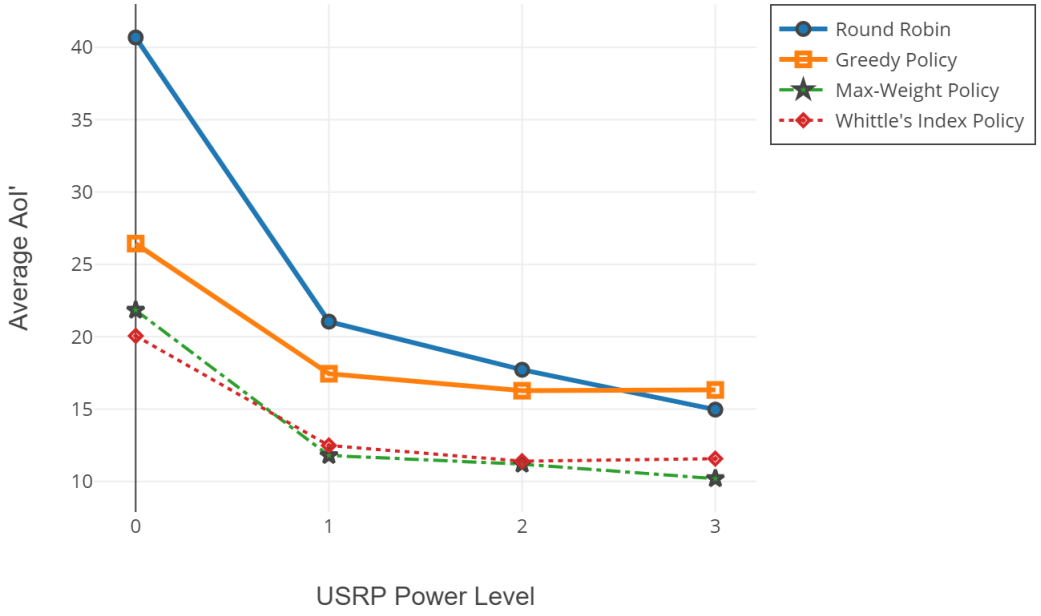

Figure 2.4: Evaluation of Average AoI J_A with varying Receiver Gain (SDR Testbed)

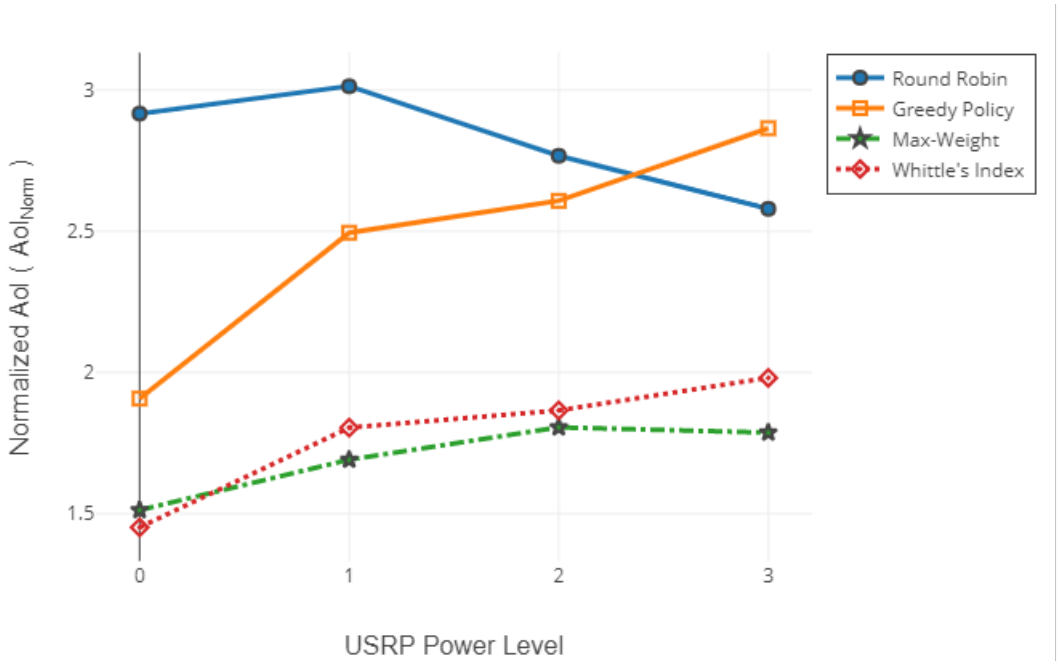

Figure 2.5: Evaluation of Normalized AoI AoI_{Norm} with varying Receiver Gain (SDR Testbed)

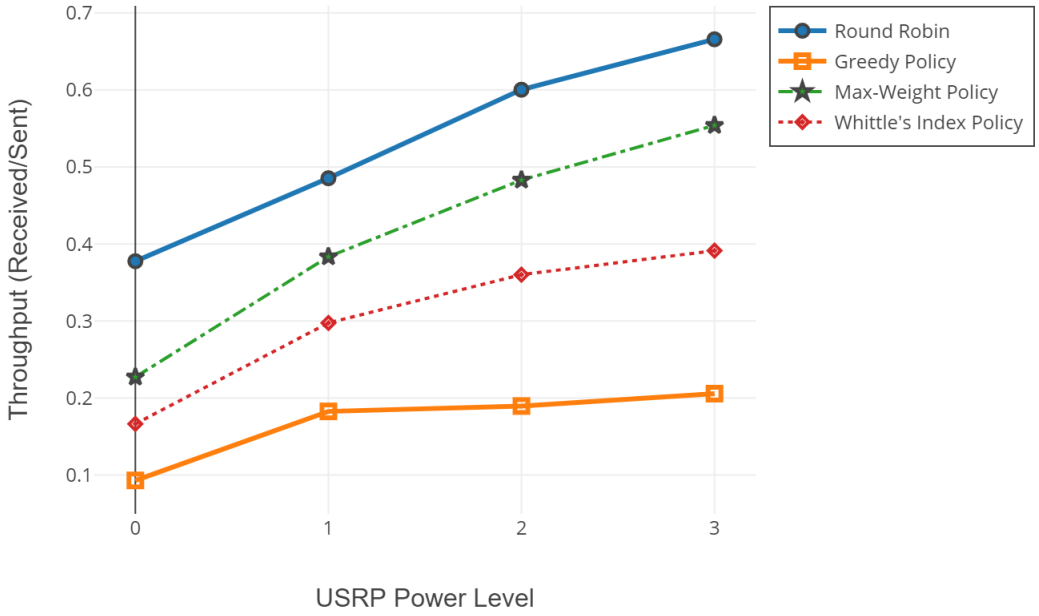

Figure 2.6: Evaluation of Throughput with varying Receiver Gain (SDR Testbed)

Table 2.1: Channel Statistics in the First Experiment Set

Gain	P1	P2	P3	C_V
0	0.9997	0.0517	0.0779	0.84
1	0.9997	0.3698	0.078	1.16
2	0.9997	0.7135	0.0747	1.337
3	0.9997	0.9139	0.0795	1.361

2.3.2 Increasing the Gain of the Base Station

In this experiment set, each of the 4 policies was tested five times at each of the 4 transmitter gain levels, and the two results were averaged. In each experiment, the frame length was $K = 7500$, and there were $M = 3$ receiver.

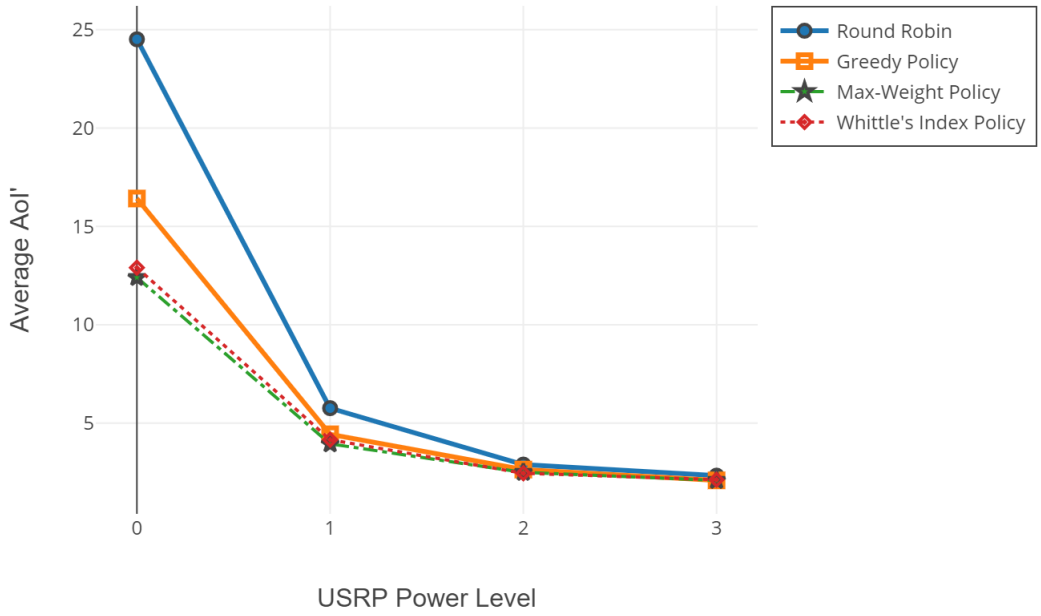

Figure 2.7: Evaluation of Average AoI J_A with varying BS Output Gain (SDR Testbed)

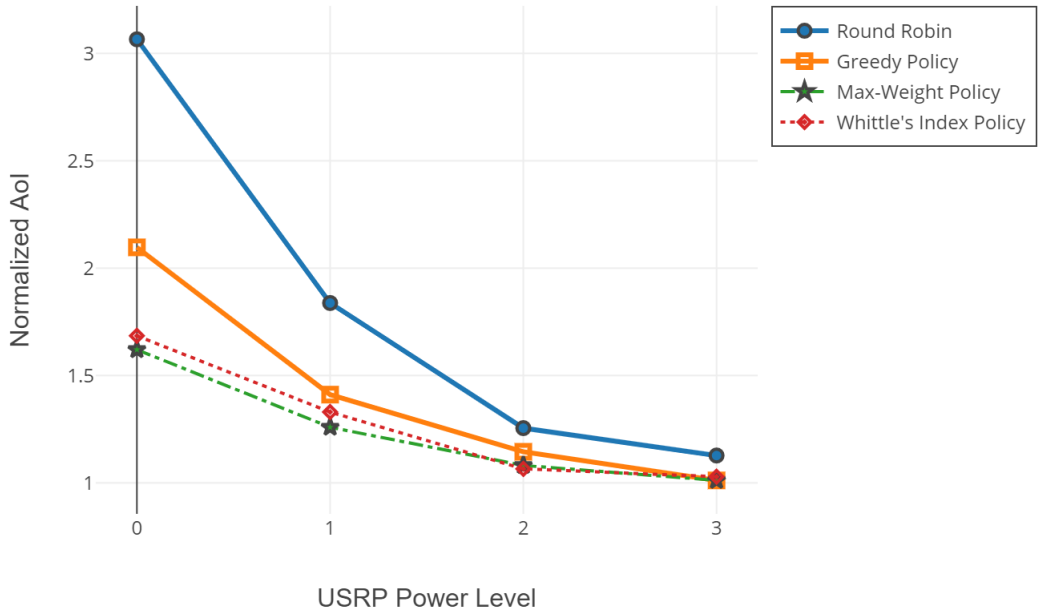

Figure 2.8: Evaluation of Normalized AoI AoI_{Norm} with varying BS Output Gain (SDR Testbed)

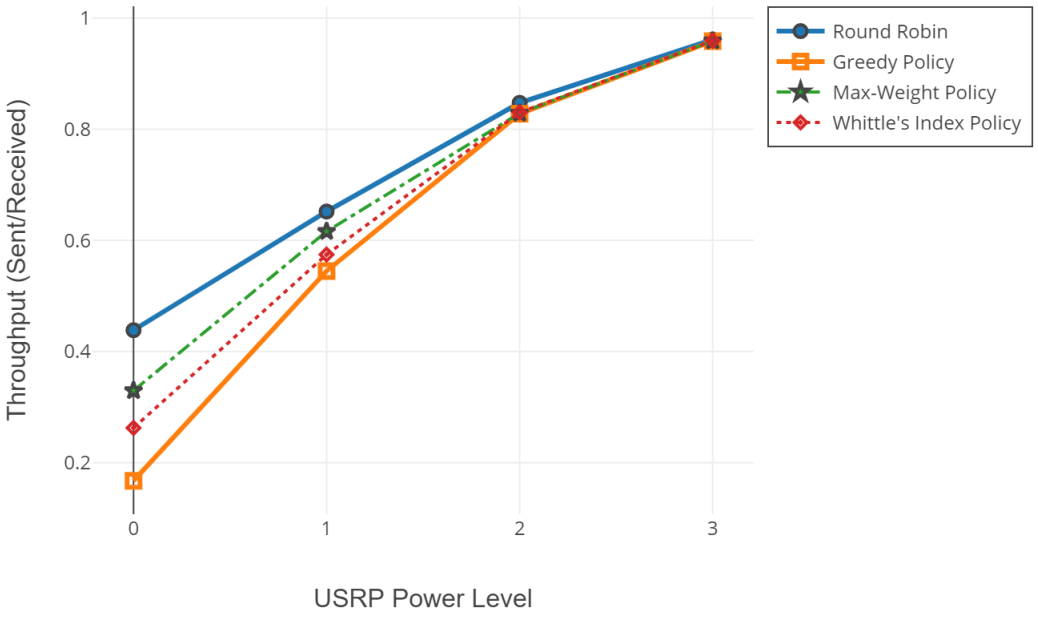

Figure 2.9: Evaluation of Throughput with varying BS Output Gain (SDR Testbed)

Table 2.2: Channel Statistics in the Second Experiment Set

Gain	P1	P2	P3	C_V
0	0.9997	0.0814	0.2317	0.988
1	0.9997	0.3566	0.5891	0.496
2	0.9997	0.6811	0.8587	0.196
3	0.9997	0.9055	0.9733	0.051

2.3.3 Interpretation of the Results

As channel reliability decreases, the performances of MW and WIP differ positively from the others. MW and WIP policies take channel reliability into account in the scheduling decision. This enables more efficient use of transmission attempts. For example, by reducing the number of attempts to update a receiver with very low channel reliability, starvation of the system may be prevented. On the other hand, Greedy Policy does not use channel reliability information. The continuous unsuccessful up-

date attempts to a receiver with very low channel reliability waste update packets; thus, the AoI increases. In the first experiment that considers the effects of increasing the input gain of an individual receiver, the performance drop of Greedy is more apparent. Greedy Policy constantly tries to send an update packet to the third receiver. However, the third receiver rarely receives packets successfully, and the base station gets stuck in that receiver until the packet is appropriately received. On the other hand, since the Round Robin policy proceeds by scanning all receivers one by one without using any information about whether the packet is appropriately received, the starvation problem does not occur. In both experiments, it is seen that as channel reliabilities of receivers improve and asymmetry of channels decreases, Greedy Policy shows better performance than Round Robin.

As the channel conditions improve and the asymmetry among the channels decreases, performances of both policies converge to the optimal. In case of 100 percent transmission success, all scheduling policies will behave like Round Robin and scans all receivers in order.

In the experiments, as the gain of the base station decreases, the normalized performance indicator for MW and WIP policies moves away from one as can be seen in Figure 2.8. We interpret this results as MW and WIP policies drift away from optimality as the channel reliabilities deteriorate. We mainly explain this results by the looseness of the lower bound at the low channel reliability region. Moreover, short-term variations in the channel reliability throughout the experiment run can also be a contributing factor. The channel reliability is an input for the MW and WIP policies and used to calculate the coefficient of each receiver. In the implementation, we observed that the channel reliability often converges to a fixed value after some time. However, at short intervals, it may not remain constant and vary continuously throughout the experiments. As the channel reliabilities decay, this variation may become more apparent, and as a result, the performances of the MW and WIP policies may be affected.

2.3.4 Comparison of SDR Testbed Results with Simulations

In this section, we share results of the comparison between simulation and implementation. We used the results of the experiment mentioned in Section 2.3.2 as a reference. We used same channel reliabilities from Table 2.2 for the simulation environment and evaluated the policies. Results of the comparison in terms of Average AoI is given in Figure 2.10, in terms of Normalized AoI is given in Figure 2.11 and in terms of Throughput is given Figure 2.12.

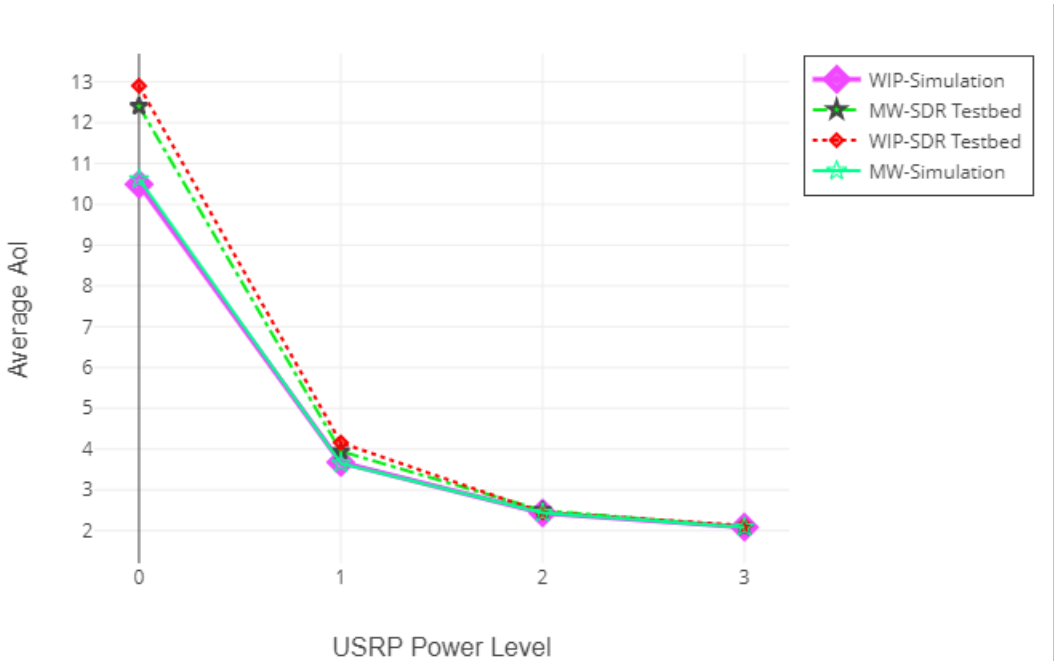

Figure 2.10: Comparison of Simulation and Implementation-Average AoI

We used the same average channel reliability values in the simulation and SDR implementation experiments. As a result, we did not observe any difference between experiments in terms of throughput. However, in terms of AoI, we found that the results in the simulation were at lower AoI values compared to the SDR implementation. In the simulation environment, the transmission availability of the channel for each time slot is determined as a Bernoulli random variable. However, in the SDR implementation, the channel state is formed by the real environment conditions and does not have to be following the Bernoulli distribution. Real channels may have memory. For example, consider a model where the noise in the channel is periodically very high for an interval and very low the rest of the time. In this noise profile,

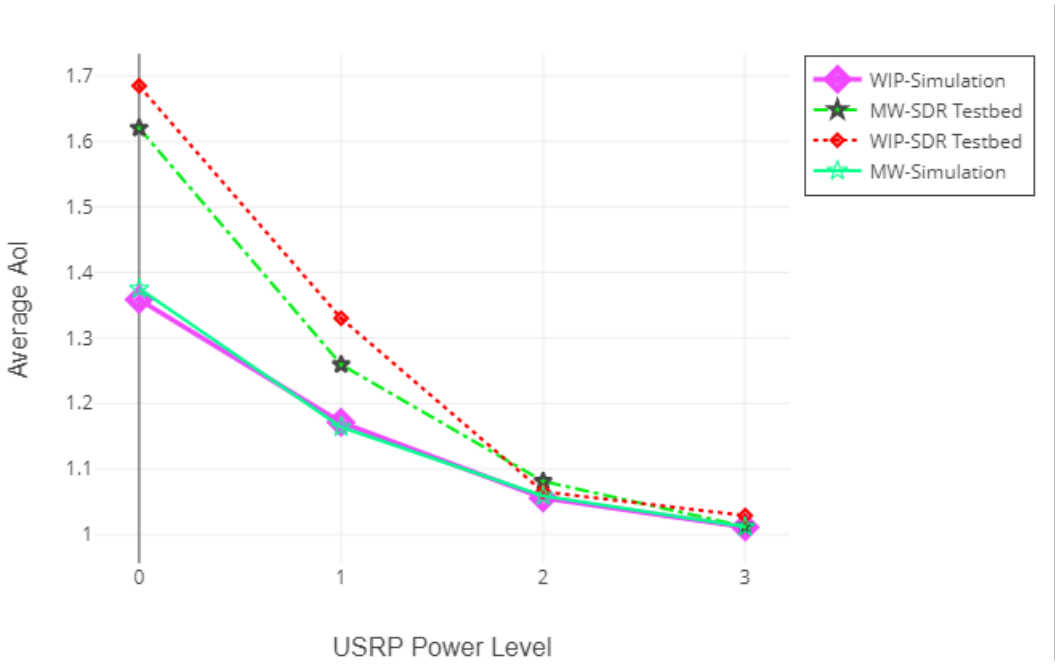

Figure 2.11: Comparison of Simulation and Implementation-Normalized AoI

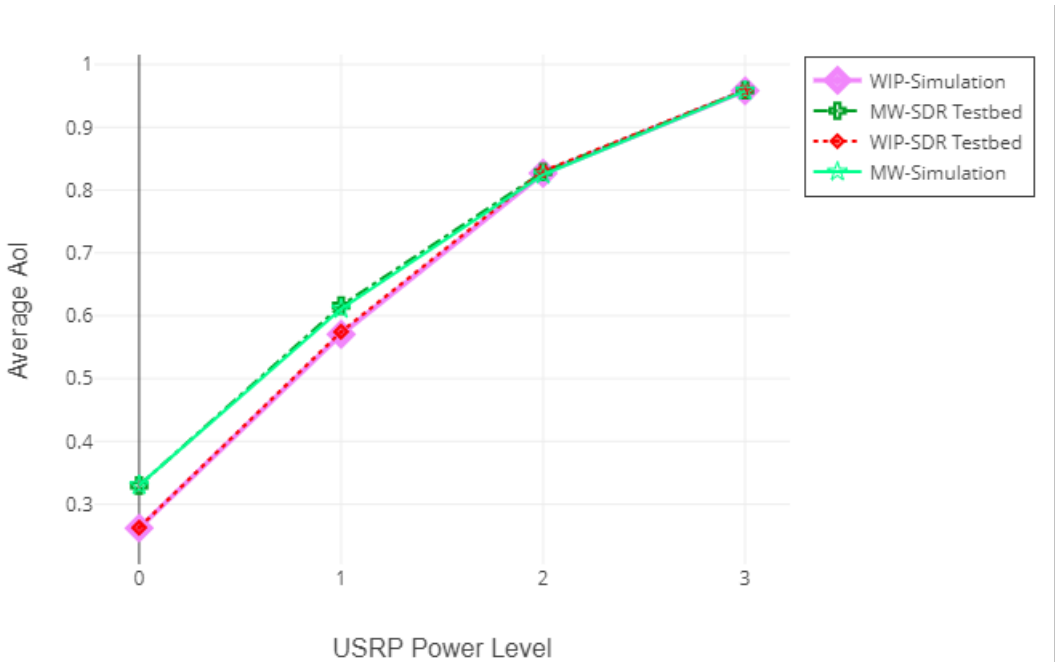

Figure 2.12: Comparison of Simulation and Implementation-Throughput

even if the channel reliability over time is the same as that of a channel with Bernoulli distribution, regular packet updates may be problematic. The regular arrival of packages is an essential factor for low AoI. If channels in the implementation environment cause more irregular packet transmission than Bernoulli for the same throughput, we expect AoI to increase.

2.3.5 Simulation Results for High Number of Receivers

In this section, we examine the effects of an increasing number of receivers on scheduling policies. Our motivation for this study is to investigate the requirements of new technologies such as IoT or Industry 4.0, which highlights situations where many receivers communicate with a central node. For this purpose, we created a MATLAB simulation to examine scenarios with such a high number of receivers. We chose the average channel reliability of each receiver as a randomly chosen value between 0.01 and 1.

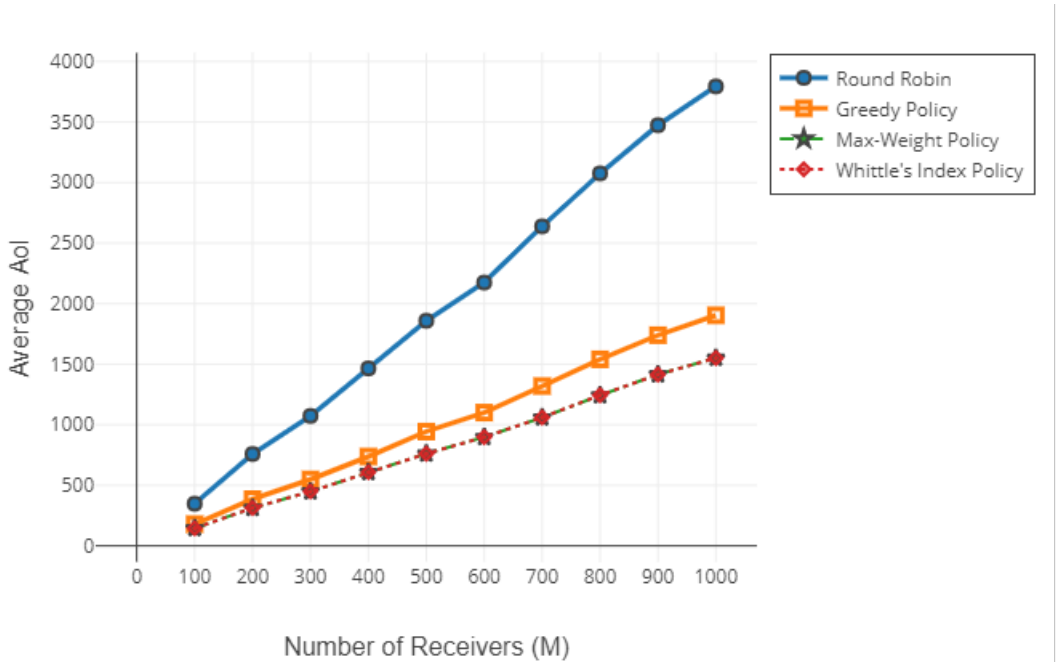

Figure 2.13: Evaluation of Average AoI as the number of receivers is increased-Matlab Simulation

According to the experiment results, as the number of receivers increases, the difference between the AoI values of Age-Aware policies and traditional policies increases.

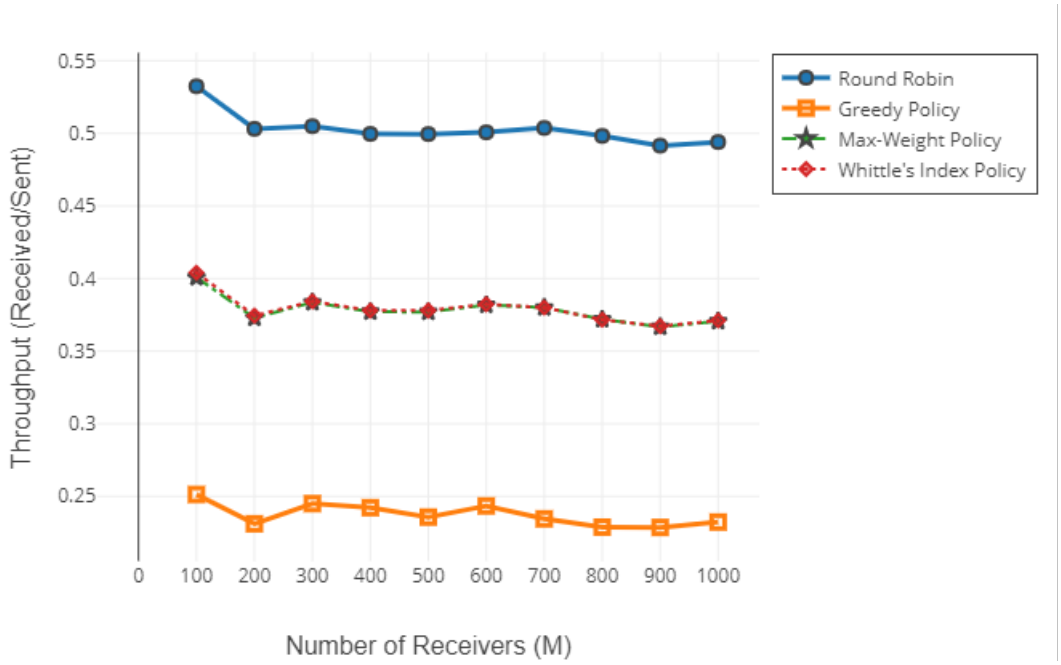

Figure 2.14: Evaluation of Throughput as the number of rReceivers is increased-Matlab Simulation

However, when we analyze the AoI performances of policies with respect to each other, we observe that the ratio of the AoI values of Greedy and AoI-Aware policies remains approximately unchanged through experiments. We consider that increasing the number of receivers does not significantly affect the asymmetry of the channels. If the asymmetry does not show a drastic change, we don't expect any performance degradation in Greedy's performance.

Since we assigned the channel reliability values of all receivers to a randomly selected value between 0.01 and 1, we did not encounter a significant change in throughput as the number of receivers increased. From the experiment results in Figure 2.13, we observe that increasing the number of receivers reduces the performance of the Round Robin policy the most. As the number of receivers increases, the AoI difference between Round Robin and other policies grows at a higher pace.

CHAPTER 3

QUERY AGE OF INFORMATION

3.1 Literature Review

Metrics that prioritize the semantics of information, including AoI, centers the significance and relevance of the information sent [29]. Among the semantic metrics, AoI examines the freshness of the information by assigning equal importance to the freshness at every time instant. Other metrics alter this value assignment. For example, the Age of Incorrect Information does not attach importance to the freshness at every moment, but only when the information is changed [17]. Similarly, in the Query-Age of Information, the freshness of the information is valuable only at the $query$ instants [4, 16, 33]. The query sources generate these queries. A query source can be a user or an application that requests the freshest information available.

The Query-Age of Information provides a way for analyzing the pull-based scenarios with the AoI perspective. In the traditional $push$ based network model, the packet generation at the source unit triggers the communication process. Then, the source unit pushes the packet to the network. In contrast, in the $pull$ based model, the query source proactively requests (or queries) the information. Then, the communication process begins. In this scenario, the initiator of the communication process is the query source that pulls the information. These requests may be generated by users or applications tracking an information source.

There are various studies in the literature related to Query-AoI. In [33], the Effective AoI (EAoI) metric, which is closely related to the Query-AoI, is presented. The EAoI metric aims to minimize the sum of the AoI's at the queried frames. According to this metric, the Instantaneous EAoI value of the frames without query is 0. In [33],

the EAoI metric is investigated in a downlink multiuser scheduling problem under the pull-based model concept. For the static query probabilities scenario, a query for each receiver was an independent and a Bernoulli-distributed random variable. The system model also included a proactive serving scenario, the response procedure to the query that aims to serve the freshest information as a response. For this model, Effective AoI aware Whittle's Index Policy was presented. In [4, 9], the Query-AoI metric was presented. Unlike EAoI, QAoI does not include non-queried frames in the average QAoI calculation. In [4, 9], a system model with one Receiver that has an energy constraint is investigated. Two different Query arrival processes are applied to the receiver. The first strategy was the Permanent Query (PQ) strategy. In the Permanent Query case, the Scheduler was unaware of the query process and acted as a query is present at every frame. In the end, the optimal scheduling for the PQ case turns out to be similar to the AoI scheduling without queries. The second strategy for the query arrivals was the Query Arrival Process-Aware (QAPA) strategy. In the QAPA case, the query process was deterministic, and the Scheduler knew query times.

3.2 Effective Age of Information

3.2.1 System Model

The system model we use to study the EAoI is similar to the AoI model described in Section 2.2. The base station (BS) needs to send time-sensitive packets to the receivers. In the meantime, multiple receiver modules track time-sensitive information in the form of status update packet flows sent to them by a common access point or base station over orthogonal channels. System time is divided into fixed-length frames. In each frame, the base station can activate only one link, transmitting to a single receiver. Downlink channels are unreliable such that for each link, there is a constant probability that the packet reception will not be successful.

Every Receiver has its query source, a user, or an application that tracks the obtained time-sensitive information by sending queries. The queries immediately reach the receiver without error. Responses to the queries are also transmitted to the query sources without error. Generated queries arrive at the beginning of the frame. The

Receiver answers this request by sending the latest information received from the base station. In a frame, the arrival of a query for a receiver is a Bernoulli distributed random variable.This random variable is independent for each receiver i and each frame k.

The $a_i(k)$, $c_i(k)$ and $u_i(k)$ notations have the same meaning with the Chapter 2. The $a_i(k)$ indicates the decision of the base station in frame k. The $c_i(k)$ is an indicator for the channel state in each frame. Lastly, the $u_i(k)$ gives the result of the transmission to user i in frame k. The detailed information about the variables are given in Section 2.2.

$$a_i(k) = \begin{cases} 1 & \text{if the receiver } i \text{ is Selected} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

$$c_i(k) = \begin{cases} 1 & \text{if the Channel is ON} \\ 0 & \text{Channel is OFF} \end{cases} \tag{3.2}$$

$$u_i(k) = \begin{cases} 1 & \text{if } c_i(k)a_i(k) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

For each frame, the existence of a query for each receiver is modeled as independent Bernoulli arrival. The presence of query to the receiver i in frame k is denoted with $d_i(k)$. If the receiver i is queried in frame k, then $d_i(k)$ is equal to 1. Otherwise, $d_i(k)$ is equal to 0. $d_i(k)$ is a Bernoulli random variable with a mean value q_i.

$$d_i(k) = \begin{cases} 1 & \text{Receiver is Queried} \\ 0 & \text{Receiver is not Queried} \end{cases} \tag{3.4}$$

The evaluation of the $\Delta_{q_i}(k)$ varies with the response scenario of the query. According to the query response scenario, receivers may immediately respond to the queries they receive or delay the response for a while. Throughout this chapter, we investigated the Instantaneous Serving scenario and the Proactive Serving Scenario.

Communication between Receivers and Query Sources proceeds through query sources'

requests to retrieve information. On the other hand, the communication between the base station and receivers proceeds through the effort of the base station to push information to the receivers. The receiver tries to receive up-to-date information from the base station. On the one hand, the receiver tries to share the most up-to-date information to the query source if a query arrives. If the receiver is queried, the query response scenario defines when the receiver will respond to this request. For example, when the receiver is queried, it can instantly respond and send the most up-to-date information available. However, in this case, if the receiver obtains more up-to-date information after the moment the query is responded until the end of the frame, it misses the chance to transmit the freshest information. Within the scope of the thesis, we study the instant response case under the name of the "Instantaneous Serving" scenario.

In the system model we examined, the receiver knows whether the base station chooses itself for transmission at the start of the frame. A more efficient query response scenario can be obtained if the receivers utilize this information in the query responding process. The receiver tries to receive the freshest packet until the end of the frame and waits for the query response if it sees that the base station selects it. If it can get the freshest packet before the frame ends, it transmits this new information as a query response. If it fails to receive the packet, it transmits the old information it already has. However, in the failed transmission case, as one more frame passes over the old information while trying to receive the new packet, the AoI value of the transmitted information increases by one frame. This query response framework is known as the "Proactive Serving" scenario in the literature.

The instantaneous EAoI of the receiver i at the frame k is denoted as $\Delta_{q_i}(k)$. In the Instantaneous Serving, the queries are answered immediately upon their arrival. The figure 3.1 shows the evaluation of the EAoI under the Instantaneous Serving scenario. The EAoI metric only considers the AoI at query instants. These instants are shown as red dashed lines in the figure. The closed-form definition of the $\Delta_{q_i}(k)$ in the Instantaneous Serving scenario is defined in (3.5).

$$\Delta_{q_i}(k) = d_i(k)\Delta_i(k) \tag{3.5}$$

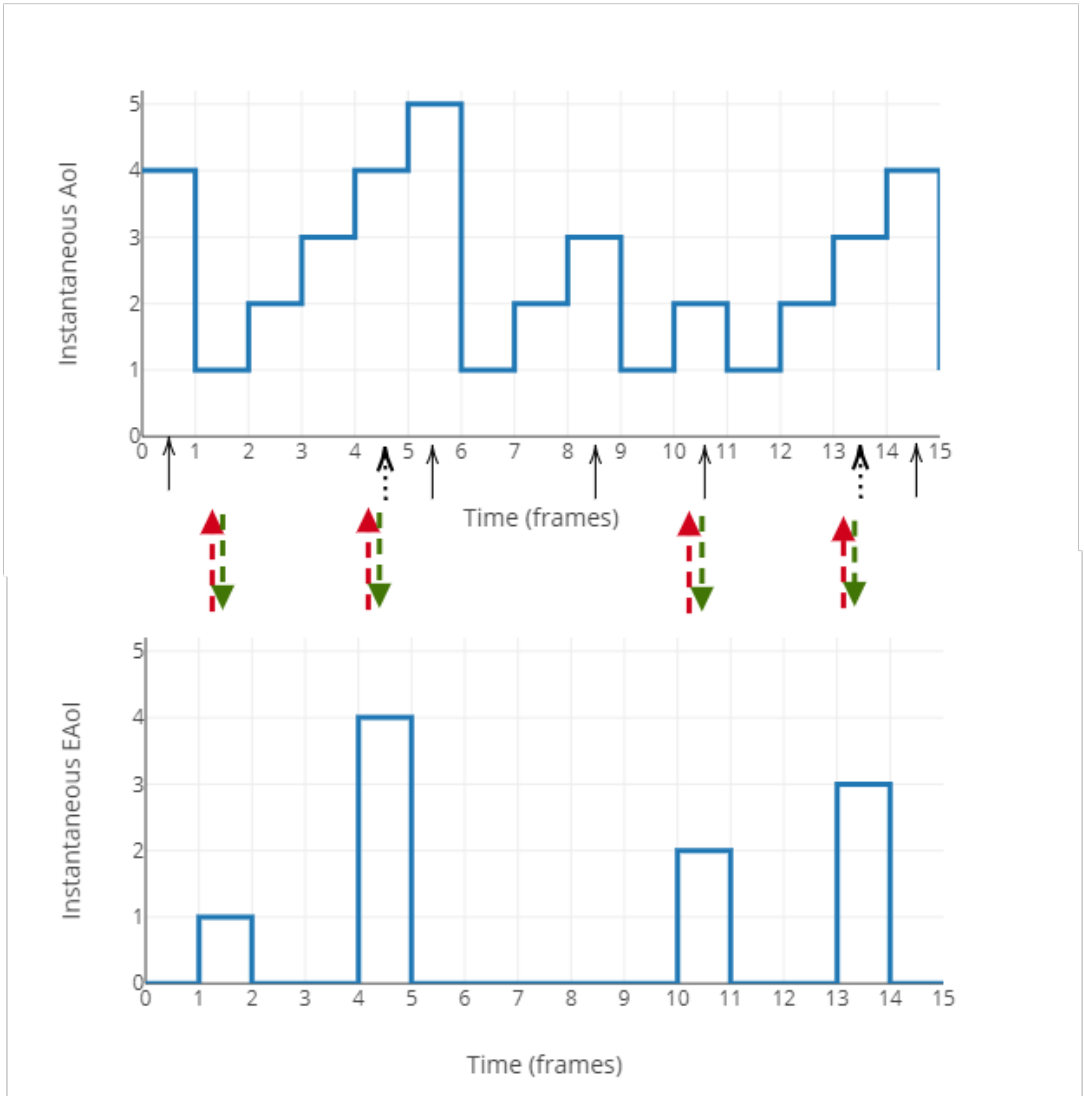

Figure 3.1: EAoI at the Instantaneous Serving scenario

The Proactive Serving scenario is defined in [33]. In the Proactive Serving scenario, the response to the query may be delayed by one frame or arrive before the queried frame has ended. The Figure 3.2 shows the evaluation of the EAoI under the Instantaneous Serving scenario. For example, if the receiver gets a packet within the queried frame, the receiver will immediately respond. In this case, the request and response of the query will be in the same frame. Also, if the queried Receiver is not selected for transmission in the queried frame, the query's response will appear immediately. A one-frame delay occurs only when the selected Receiver fails to obtain the packet in the queried frame. The closed-form definition of the instantaneous EAoI metric under the proactive serving scenario is given in Equation (3.6).

$$\Delta_{q_i}(k) = \begin{cases} d_i(k) & \text{if } u_i(k) = 1 \\ d_i(k)\left(\Delta_i(k) + a_i(k)\right) & \text{otherwise} \end{cases} \tag{3.6}$$

3.2.2 The Optimization Problem

The main objective in the network is the minimizing the average EAoI of the query sources. To provide a mathematical definition for this objective, we firstly define the Penalty function. The Penalty function calculates the average EAoI of M receivers throughout K frames, where the Policy π is adopted as a scheduling policy. The evaluation of the function is given in (3.7).

$$J_E(\pi) = \frac{1}{KM}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_{q_i}(k)\right] \tag{3.7}$$

The objective of the optimization problem is to minimize the expected value of the Penalty function.

$$\min_{\pi\in\Pi}\mathbb{E}\left[J_E(\pi)\right], \text{ where } J_E(\pi) = \frac{1}{KM}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_{q_i}(k)\right] \tag{3.8}$$

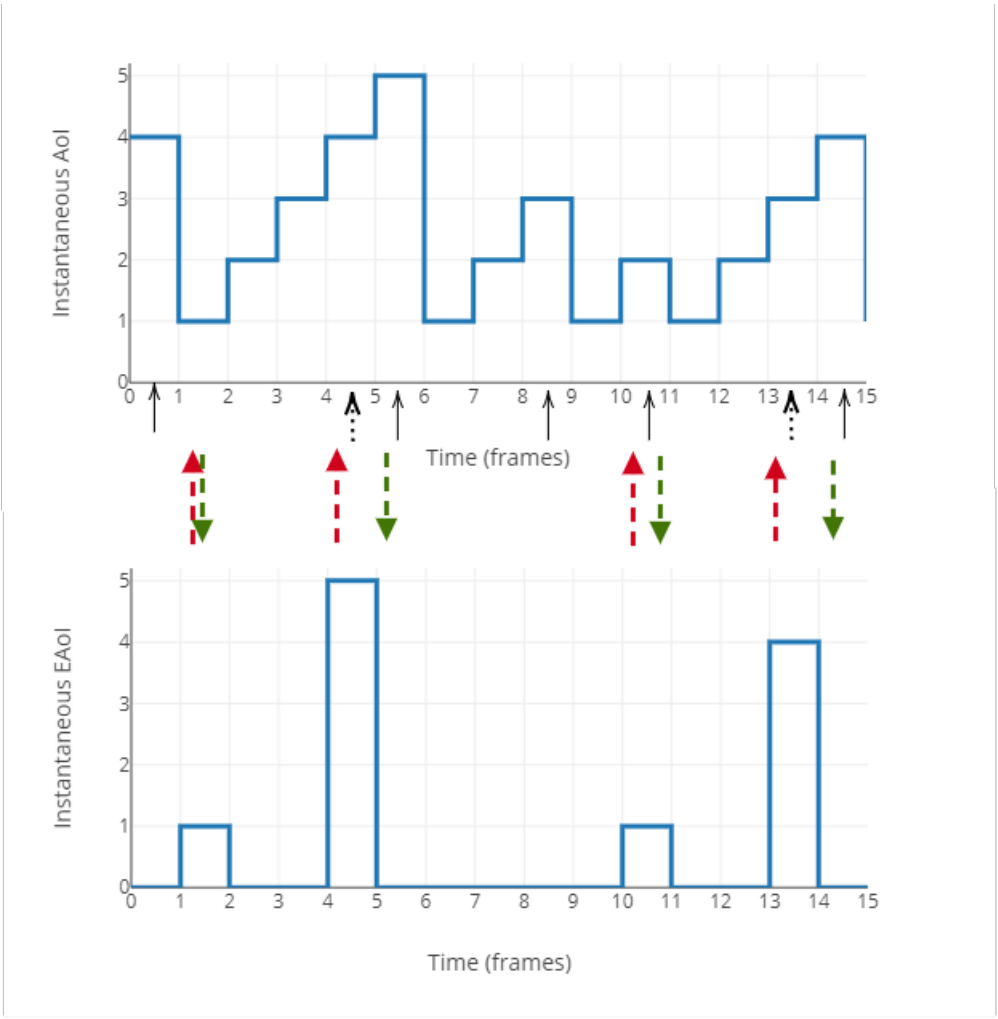

Figure 3.2: EAoI at the Proactive Serving scenario

3.2.3 Scheduling Policies

Along with the policies we implemented in the AoI case in Section 2.2.3, We implement the EAoI-Aware WIP proposed in [33]. We also implement EAoI-Aware Max-Weight Policy by modifying the Max-Weight Policy previously proposed in [13]. To obtain a policy to achieve higher throughput, we modify the Bellman Equations and examine the effects of this modification on Whittle's Index Policy.

3.2.4 EAoI-Aware Max-Weight Policy

The Max-Weight Policy we used in this section is the modified version of the Max-Weight Policy in [13]. We used this Policy under the Instantaneous Serving scenario. To obtain the Policy, We calculated the Lyapunov Drift of the Instantaneous EAoI's between consecutive frames. In line with [13], we selected Quadratic Lyapunov function to calculate the Lyapunov Drift. For each receiver, calculation of the Lyapunov Drift $Y_i(k)$ between the frames k and $k+1$ is given in (3.9).

$$\begin{aligned} Y_i(k) =& \mathbb{E}\left[\Delta_{q_i}^2(k+1) - \Delta_{q_i}^2(k)\right] \\ =& \mathbb{E}\left[d_i^2(k)\Delta_i^2(k+1) - d_i^2(k)\Delta_i^2(k)\right] \\ =& \mathbb{E}\left[d_i(k)\Delta_i^2(k+1) - d_i(k)\Delta_i^2(k)\right] \end{aligned} \tag{3.9}$$

Since we assume that policies are non-anticipative, which means the policies don't have an information about future channel or query status, we can argue that $\Delta_i(k)$ and $d_i(k)$ are independent. Therefore, we argue that $\mathbb{E}\left[d_i(k)\Delta_i(k)\right] = \mathbb{E}\left[d_i(k)\right]\mathbb{E}\left[\Delta_i(k)\right]$. We write the Lyapunov Drift using $\mathbb{E}\left[d_i(k)\right] = q_i$ and $\mathbb{E}\left[d_i(k+1)\right] = q_i$.

$$Y_i(k) = \mathbb{E}\left[q_i\Delta_i^2(k+1) - q_i\Delta_i^2(k)\right] \tag{3.10}$$

After this modification, the derivation process becomes identical with the [13]. We

write the transition of $\Delta_i(k)$ between consecutive frames is given in (3.11).

$$\begin{aligned}\Delta_i(k+1) =& u_i(k) + (1-u_i(k))(\Delta_i(k)+1) \\ =& a_i(k)c_i(k) + (1-a_i(k)c_i(k))(\Delta_i(k)+1)\end{aligned} \tag{3.11}$$

Then, we rewrite the Lyapunov Drift by expressing $\Delta_i(k+1)$ in terms of $\Delta_i(k)$.

$$Y_i(k) = \mathbb{E}\left[q_i\left[u_i(k) + (1-u_i(k))\left(\Delta_i(k)+1\right)\right]^2 - q_i\Delta_i^2(k)\right] \tag{3.12}$$

Since $u_i(k)$ is 0-or-1 variable, we can argue that $u_i^2(k) = u_i(k)$, $(1-u_i(k))^2 = (1-u_i(k))$ and $u_i(k)(1-u_i(k)) = 0$. With these simplifications, we can rewrite $Y_i(k)$.

$$\begin{aligned}Y_i(k) =& \mathbb{E}\left[q_i\left[u_i(k) + (1-u_i(k))(\Delta_i(k)+1)^2\right] - q_i\Delta_i^2(k)\right] \\ =& \mathbb{E}\left[q_i\left[\Delta_i^2(k) + 2\Delta_i(k) + 1 - u_i(k)\Delta_i^2(k) - 2u_i(k)\Delta_i(k)\right] - q_i\Delta_i^2(k)\right] \\ =& \mathbb{E}\left[q_i\left[2\Delta_i(k) + 1 - u_i(k)\Delta_i^2(k) - 2u_i(k)\Delta_i(k)\right]\right] \\ =& q_i\left[2\Delta_i(k) + 1 - \mathbb{E}\left[u_i(k)\right]\Delta_i^2(k) - 2\mathbb{E}\left[u_i(k)\right]\Delta_i(k)\right] \\ =& \left[2q_i\Delta_i(k) + q_i - q_ip_i\mathbb{E}\left[a_i(k)\right]\Delta_i^2(k) - 2q_ip_i\mathbb{E}\left[a_i(k)\right]\Delta_i(k)\right]\end{aligned} \tag{3.13}$$

The $a_i(k)$ is the decision variable that we can select zero or one. We only aim to investigate the effect of changing $a_i(k)$. Since the results of other terms in $Y_i(k)$ does not change as $a_i(k)$ change, we omit them and focus on the terms that have $a_i(k)$ as a coefficient.

$$C_i(k) = q_ip_i\left(\Delta_i^2(k) + 2\Delta_i(k)\right) \tag{3.14}$$

$$\begin{aligned}\hat{Y}_i(k) =& -q_ip_i\mathbb{E}\left[a_i(k)\right]\Delta_i^2(k) - 2q_ip_i\mathbb{E}\left[a_i(k)\right]\Delta_i(k) \\ =& -\mathbb{E}\left[a_i(k)\right]\left(q_ip_i\left(\Delta_i^2(k) + 2\Delta_i(k)\right)\right) \\ =& -\mathbb{E}\left[a_i(k)\right]C_i(k)\end{aligned} \tag{3.15}$$

The Max-Weight Policy aims to minimize the average Lyapunov Drift by attempting the eliminate the receiver i with maximum $C_i(k) = q_i p_i \left(\Delta_i^2(k) + 2\Delta_i(k)\right)$ at every frame k.

3.2.5 EAoI-Aware Whittle's Index Policy

Various approaches in the literature consider the AoI related optimization problems as a Markov Decision Process. Dynamic Programming based methods can be proposed to solve MDP problems. However, Backwards Induction, the primary solution method of dynamic programming, makes it harder to find a solution as the size of the system examined within the scope of the problem grows. This situation is known as the "curse of dimensionality." In contrast, the MAB-based approach provides a Forward Induction solution that demands less computational resources [18]. There are approaches in the literature that handles the AoI or EAoI related optimization problems as a relaxed version of the Restless Multi-Armed Bandit problem, then proposes a Whittle's Index Policy as a solution [14, 33]. The Multi-Armed Bandit is a problem that aims to optimize the reward in an unknown environment through a series of trials [30, 31]. In each turn, the decision-maker is allowed to activate one arm of the bandit. Each arm has an immediate reward(or penalty, for the minimization problem case) related to it. Nevertheless, the immediate reward is not known by the decision-maker before the arm selection decision. The immediate reward is a sample in time from an underlying reward generation process. This process could be following a specific distribution. In the Multi-Armed Bandit Problem, the distribution of the process is assumed to be not changing throughout the decisions [34]. In the beginning, the decision-maker has no information about the process. When the decision-maker selects an arm, it obtains an immediate reward from the decided arm. Besides, the decision-maker grabs a piece of information about the underlying process of the decided arm by observing the obtained reward. Therefore, as a result of the arm selection, the decision-maker receives an immediate reward and information about the utility process of the selected arm. In the next turn, the decision-maker may utilize the previous information pieces for the arm selection decision. The decision-maker has limited arm selection chances. The objective is to maximize the sum of the immediate utilities obtained from the decisions.

In the Multi-Armed Bandit problem, the decision-maker faces the "Exploration or Exploitation" dilemma. A good solution for the multi-armed bandit problem lies in bringing a balance between exploration and exploitation. The exploitation-centered Policy seeks to maximize short-term benefits. As reward maximization is the main objective, the Policy ignores the process of acquiring new knowledge about arms. Therefore, the future rewards of the arms are estimated with limited information. Due to the lack of information, this estimation may fail to discover the arms that provide higher rewards in the long term.

On the other hand, the Exploration-Centered Policy aims to improve the information about the reward processes of the arms, rather than the maximization of short-term rewards. However, applying The Exploration-Centered Policy comes with a cost. Since the main objective of the MAB is to maximize the overall reward, the Exploration-Centered Policy causes the selection of the arms with the low reward. Therefore, the overall reward may drift away from the optimal.

The Bellman Equations captures the transition of the overall cost between consecutive frames. With the Bellman Equation, states, actions, and rewards(or costs) are linked. The Bellman Optimality Equation from an eye of an agent is given in (3.16).

$$V(s) = \min_{a} \left(C(s,a) + \gamma \left[\sum_{s'} p(s'|s,a) V(s') \right] \right) \tag{3.16}$$

In the Equation (3.16), s denotes the current state of an agent, and the $V(s)$ is the value (cumulative cost) of the current state s. The equation describes the calculation of the value function in terms of actions, states and the future values. After taking the action a, the new state of the agent will be s'. $C(s,a)$ denotes the immediate cost of the decision a. $p(s'|s,a)V(s')$ is the expected future cost of the action a. The $p(s'|s,a)$ term is the probability of reaching s' state when a action is selected, and the $V(s')$ is the value(cumulative cost) of the new state s'.

The future cost is multiplied by the discount factor γ to indicate the weight of future costs. γ can be selected between 0 and 1. The future cost of $V(s+1)$ has γ coefficient. In the next future states, $V(s+2)$ will have γ^2, $V(s+3)$ will have γ^3, $V(s+n)$ will have γ^n coefficient and so on. Since the γ is not greater than one, γ^n will get closer

to zero as n goes to infinity. Therefore, as the states progress through actions, the weight of the future states will gradually decrease with the effect of the γ multiplier. Selecting γ equal to 1 will remove the decay of the future states, and every state will equal weight. As the γ value decreases, the weight of the future states in the overall sum also decreases.

With the Bellman Equation, the agent tries to minimize the overall cost through decision steps. The output of the Bellman Equation is an action set for all decision steps that minimize the total cost.

As a result of the relaxation of the Multi-Armed Bandit Problem, the Scheduler within the Base Station can evaluate the decision-making process independently for each receiver. We describe the individual decision-maker as agent. Each Receiver in the network has a corresponding agent that manages the transmission trials. The Scheduler divides the overall scheduling problem into smaller sub-problems by assigning a virtual agent to each receiver. Then, the Scheduler adopts the decision of the virtual agent that achieves the best result as the overall decision.

Every action of the agent incurs a cost. This cost varies according to the current state of the agent and the action taken. We can examine the cost of action in two parts as the immediate cost and the future cost. The Immediate cost is the cost that the agent directly encounters after the action it takes. Future cost is the overall cost of the new state reached as a result of the action. The cumulative cost of the state is often described as the value of that state.

In our system model, the EAoI minimization problem can be interpreted as a relaxed version of a Restless Multi-Armed Bandit Problem [13, 14, 33]. Restless property of the MAB indicates that a state change must happen after any action, even if the Idling action is selected. In our system model, AoI increases even if the receiver is not selected for transmission, which corresponds to a state change. In our case, the utility maximization term in the formal definition is mapped to the Penalty minimization, and the RMAB problem aims to minimize the penalty, the EAoI. The receivers are arms of the bandit and the relaxed version solves the problem by treating each arm individually. In this case, Bellman equations can be derived for each arm individually. Since each receiver is handled independently in the relaxed version of the RMAB, we

should also rearrange the objective function to calculate it over a single receiver. The average EAoI function in Equation (3.8) evolves to the Equation (3.17).

$$\hat{J}_E(\hat{\pi}) = \lim_{K\to\infty} \mathbb{E}\left[\frac{1}{KM}\left[\sum_{k=1}^{K}\sum_{i=1}^{M} \Delta_{q_i}(k) + \hat{C}a_i(k)\right]\right] \tag{3.17}$$

The Bellman Equations for the EAoI minimization problem is described in [33] for the proactive scenario and given in (3.19). The Instantaneous Serving scenario is analogous with the AoI minimization problem in [13], and the corresponding cost-to-go function is given in (3.22). Essential notations for understanding the Bellman Equations are provided in Table 3.1

Table 3.1: Table of Notation for the Bellman Equations

$f(.)$	Value function that gives the cumulative cost of the corresponding state
$f(\Delta)$	Value(cumulative cost) of the receiver at the Δ state
$f(1)$	Value of the state where $\Delta = 1$
λ	Lagrange multiplier
$\hat{C}$	Constant cost of one transmission attempt
Δ	AoI of the receiver
q	Expected Query Probability of the receiver
p	Expected Channel Reliability of the receiver

3.2.5.1 EAoI-Aware Q-WIP For the Proactive Scenario

We rewrite the objective function in Equation (3.21) by replacing the $\Delta_{q_i}(k)$ with the Equation (3.5). Since we are interested with the expected value of the (3.17), we

replace $d_i(k)$ with the expected value of $d_i(k)$, which is equal to q_i.

$$\begin{aligned}\hat{J}_E(\hat{\pi}) &= \lim_{K\to\infty} \mathbb{E}\left[\frac{1}{KM}\left[\sum_{k=1}^{K}\sum_{i=1}^{M}\Delta_{q_i}(k)\right]\right] \\ &= \lim_{K\to\infty} \mathbb{E}\left[\frac{1}{KM}\left[\sum_{k=1}^{K}\sum_{i=1}^{M} d_i(k)\left[u_i(k)+(1-u_i(k))(\Delta_i(k)+a_i(k)\right]+\hat{C}a_i(k)\right]\right]\end{aligned} \tag{3.18}$$

Then, the Bellman Equations are constructed with using the Average EAoI function at (3.18)

$$\begin{aligned}f(\Delta)+\lambda = \min\{&q\Delta + f(\Delta+1); \\ &\hat{C}+(1-p)(f(\Delta+1)+q(\Delta+1))+p(f(1)+q)\}\end{aligned} \tag{3.19}$$

The Equation (3.19) represents the costs faced by the agent during the state transition Process. The agent is allowed to take action at every frame, which triggers the state transition. This equation aims to minimize the total cost met by the agent by taking the proper actions.

In the equation (3.19), Δ is the AoI of the receiver that the agent handles and indicates the agent's state. $f(\Delta)$ indicates the cumulative cost faced by the agent from the $f(1)$ state through $f(\Delta)$. As a result of its decision, the agent may choose to stay idle or transmit the packet. Staying idle or trying transmission are the actions that the agent can take.

The Equation (3.19) states that the total cost of staying Idle is $q\Delta + f(\Delta+1)$. The agent who takes the idling action incurs cost of $q\Delta$ as the immediate cost and the value of the $f(\Delta+1)$ as the future cost. The resulting state of the agent will be $f(\delta+1)$ after staying idle.

When the agent takes the action of transmission attempt, it first incurs a constant transmission attempt cost of $\hat{C}$. In addition to this cost, it also incurs an immediate cost that varies with the result of the transmission. The expected cost for the successful transmission is $p(f(1)+q)$. The $f(1)$ indicates the value of the state where the AoI is equal to one. In this case, the expected immediate cost is pq, and the expected future cost is $pf(1)$. Note that Δ does not appear at the expected immedi-

ate cost since the Δ instantaneously drops to the 1 in the proactive serving scenario after successful transmission. The failed transmission case has an expected cost of $(1-p)(f(\Delta+1)+q(\Delta+1))$. In this case, the expected future cost is $(1-p)(f(\Delta+1))$ and the expected immediate cost is $p(f(1)+q)$.

In the literature, [33] proposes Whittle's Index Policy for the Bellman Equations in (3.19). The resulting Policy calculates Whittle's Index for each receiver. At each frame k, an index C_i will be calculated for each receiver according to (3.20) and the receiver with the highest C_i will be selected for transmission.

$$C_i\left(\Delta_i(k)\right) = q_i(p_i\Delta_i(k)+2)(\Delta_i(k)-1) \tag{3.20}$$

3.2.5.2 EAoI-Aware Q-WIP For the Instantaneous Serving Scenario

In Equation (3.21), we replace the $\Delta_{q_i}(k)$ with the corresponding equation given in Equation (3.5). Since we are interested with the expected value of the (3.17), we replace $d_i(k)$ with the expected value of $d_i(k)$, which is equal to q_i.

$$\hat{J}_E(\pi) = \lim_{K\to\infty} \mathbb{E}\left[\frac{1}{KM}\left[\sum_{k=1}^{K}\sum_{i=1}^{M} q_i\Delta_i(k) + \hat{C}a_i(k)\right]\right] \tag{3.21}$$

The Bellman Equations for the Average EAoI function in Equation (3.21) is given in Equation (3.22).

$$\begin{aligned} f(\Delta)+\lambda = \min\{&q\Delta + f(\Delta+1); \\ &\hat{C} + (1-p)\left(f(\Delta+1)+q\Delta\right) + p\left(f(1)+q\Delta\right)\} \end{aligned} \tag{3.22}$$

Similar with the Proactive serving case, the Equation (3.22) states that the total cost of staying Idle is $q\Delta + f(\Delta+1)$. The agent who takes the idling action incurs $q\Delta$ as the direct cost and the value of the $f(\Delta+1)$ as the future cost. The resulting state of the agent will be $f(\Delta+1)$ after staying idle. The term λ is a Lagrange multiplier and denotes the optimal average cost per frame [13].

When the agent takes the action of transmission attempt, along with the constant transmission cost $\hat{C}$, an immediate cost that varies with the result of the transmission

also incurs. In the successful transmission case, the expected cost is $p\left(f(1)+q\Delta\right)$. In this case, the expected immediate cost is $pq\Delta$, and the expected future cost is $pf(1)$. The value of the first state $f(1)$ is equal to zero. In the failed transmission attempt case, the overall expected cost is $(1-p)(f(\Delta+1)+q(\Delta+1))$. The expected immediate cost is $(1-p)(f(\Delta+1)+q\Delta)$ and the expected future cost is $(1-p)(f(\Delta+1))$. The Whittle's Index Policy in [13] aligns with our system model. We replace the weights of the receivers with q_i. The resulting index is given in Equation (3.23). In each frame, the base station selects the receiver with the highest C_i for transmission.

$$C_i\left(\Delta_i(k)\right)=p_iq_i\Delta_i(k)\left[\Delta_i(k)+\frac{2-p_i}{p_i}\right] \tag{3.23}$$

We argue that increasing the cost of failed transmission attempts in (3.22) may yield higher throughput. We altered the Bellman Equations by adding constant penalty H into the expected immediate cost of failed transmission to test this argument. Resulting Equation is given in (3.24).

$$\begin{aligned} f(\Delta)+\lambda = &\min\{q\Delta+f(\Delta+1);\\ &\hat{C}+(1-p)\left(f(\Delta+1)+q\Delta+H\right)+p\left(f(1)+q\Delta\right)\} \end{aligned} \tag{3.24}$$

Then, we aim to derive Whittle's Index policy for (3.24). To derive a threshold policy, we followed the derivation steps from [13]. We firstly divide the state space into two categories as non-transmit region and transmit region. If the action is in the non-transmit region, then staying idle is the best action. Otherwise, a transmission attempt is the best action. Two regions are separated with a threshold of T. If the value of a state $f(\Delta)$ is greater than the T, the action will be in the transmit region. Else, the action will be in the non-transmit region. We denote the transmission region as $T\leq\Delta$ and non-transmission region as $1\leq\Delta<T$. Note that as Δ increases, the value of the corresponding state also increases. Then, updating the receiver with high Δ would yield a higher drop in the overall cost. Intuitively we can interpret this as updating the receiver with high AoI would yield a more significant drop in the Average AoI than updating the receiver with low AoI.

We investigate the transmit region by assuming the cost of Idling is greater than the

cost of transmission attempt. Note that the value of the $\Delta = 1$ state, $f(1) = 0$. We rewrite the Bellman Equations in the $\Delta > T$ case.

$$\begin{aligned} f(\Delta) + \lambda =& \hat{C} + (1-p)\left(f(\Delta+1) + q\Delta + H\right) + p\left(f(1) + q\Delta\right) \\ f(\Delta) + \lambda =& \hat{C} + q\Delta + (1-p)H + (1-p)\left(f(\Delta+1)\right) \\ f(\Delta) =& \hat{C} - \lambda + q\Delta + (1-p)H + (1-p)\left(f(\Delta+1)\right) \end{aligned} \tag{3.25}$$

Similarly, $f(\Delta+1)$ and $f(\Delta+2)$ to $f(\Delta+n)$ can be written with the same formula in (3.25).

$$\begin{aligned} f(\Delta+1) =& \hat{C} - \lambda + q(\Delta+1) + (1-p)H + (1-p)\left(f(\Delta+2)\right) \\ f(\Delta+2) =& \hat{C} - \lambda + q(\Delta+2) + (1-p)H + (1-p)\left(f(\Delta+3)\right) \\ f(\Delta+n) =& \hat{C} - \lambda + q(\Delta+n) + (1-p)H + (1-p)\left(f(\Delta+n+1)\right) \end{aligned} \tag{3.26}$$

We write $f(\Delta+1)$ into $f(\Delta)$ and continue this process continuously. The result is given in (3.27)

$$\begin{aligned} f(\Delta) =& \hat{C} - \lambda + q\Delta + (1-p)H + \\ &+ (1-p)\left(\hat{C} - \lambda + q(\Delta+1) + (1-p)H + (1-p)\left(f(\Delta+2)\right)\right) \\ f(\Delta) =& \left[\hat{C} - \lambda + (1-p)H\right]\left(1 + (1-p) + (1-p)^2 + (1-p)^3 + ..\right) + \\ &+ q\Delta\left(1 + (1-p) + 2(1-p)^2 + 3(1-p)^3 + ..\right) \end{aligned} \tag{3.27}$$

Since the p is positive real value with $0 < p < 1$, we can use the infinite sum of geometric series formula.

$$\begin{aligned} \sum_{n=0}^{\infty}(1-p)^n &= \frac{1}{p} \\ \sum_{n=0}^{\infty} n(1-p)^n &= \frac{1-p}{p^2} \end{aligned} \tag{3.28}$$

We combine the (3.27) and (3.28) to calculate the generalized version of the value function $f(\Delta)$ in $T \leq \Delta$ region. Note that unlike the $\hat{C} - \lambda + (1-p)H$ term, the $q\Delta$ term increases as the Δ increases among states. Therefore, unchanging terms among states, $\hat{C} - \lambda + q\Delta + (1-p)H$ will be the subject of $\sum_{n=0}^{\infty}(1-p)^n$ and the increasing

part of $q\Delta$ will be part of $\sum_{n=0}^{\infty} n(1-p)^n$ as n goes to infinity.

$$f(\Delta) = \frac{\left(\hat{C} - \lambda + q\Delta + (1-p)H\right)}{p} + \frac{q(1-p)}{p^2} \tag{3.29}$$

If we assume that the value of the current state is in the non-transmit region, then the Bellman Equation transforms to Equation (3.30). Note that the derivation steps are closely similar to the [13].

$$\begin{aligned} f(\Delta) + \lambda &= q\Delta + f(\Delta + 1) \\ f(\Delta) &= q\Delta + f(\Delta + 1) - \lambda \end{aligned} \tag{3.30}$$

Similar with the (3.27), we go from $f(T)$ to $f(1)$ by interlacing the equations.

$$\begin{aligned} f(T-1) &= f(T) + q(T-1) - \lambda \\ f(T-2) &= f(T) + q(T-1) + q(T-2) - 2\lambda \\ f(T-3) &= f(T) + q(T-1) + q(T-2) + q(T-3) - 3\lambda \\ f(T-3) &= f(T) + 3qT - q - 2q - 3q - 3\lambda \\ f(T-n) &= f(T) + n\left(qT - \lambda\right) - q\sum_{j=0}^{n} j \\ f(T-n) &= f(T) + n\left(qT - \lambda - q\frac{(n+1)}{2}\right) \\ f(1) = f(T-(T-1)) &= f(T) + (T-1)\left(qT - \lambda - \frac{qT}{2}\right) \end{aligned} \tag{3.31}$$

By taking advantage of $f(1) = 0$, we can obtain a closed-form expression for $f(T)$.

$$\begin{aligned} 0 &= f(T) + (T-1)f(T) + (T-1)\left(qT - \lambda - \frac{qT}{2}\right) \\ f(T) &= (1-T)\left(qT - \lambda - \frac{qT}{2}\right) \end{aligned} \tag{3.32}$$

Lower and Upper bound for the Threshold's value can be obtained by approaching the Threshold from the transmit and non-transmit regions. The upper bound from the Transmission region is given in (3.33), and the lower bound from the Idling region is

given in (3.34).

$$\begin{aligned} q\Delta + f(\Delta+1) &> \hat{C} + (1-p)\left(f(\Delta+1) + q\Delta + H\right) + p\left(f(1) + q\Delta\right) \\ f(\Delta+1) &> \frac{\hat{C} + H(1-p)}{p} \end{aligned} \tag{3.33}$$

$$\begin{aligned} q\Delta + f(\Delta+1) &\leq \hat{C} + (1-p)\left(f(\Delta+1) + q\Delta + H\right) + p\left(f(1) + q\Delta\right) \\ f(\Delta+1) &\leq \frac{\hat{C} + H(1-p)}{p} \end{aligned} \tag{3.34}$$

Then, the resulting bound for the threshold's value is given in (3.35).

$$f(T) \leq \frac{\hat{C} + H(1-p)}{p} < f(T+1) \tag{3.35}$$

To evaluate the λ, we investigate the transmission region. To substitute the inequality in (3.35) with equality, we use a dummy variable δ that $0 \leq \delta < 1$. Since the value function in (3.29) is a monotonic increasing function as Δ increases, we can use $f(T+\delta)$ instead of $f(T+1)$ to obtain an equality. Because of the monotonic increasing property, $f(T+\delta)$ will be slightly smaller than $f(T+1)$, and it will enable us to reach the $\frac{\hat{C}+H(1-p)}{p}$.

$$\begin{aligned} \frac{\hat{C} + H(1-p)}{p} &= f(T+\delta) \\ \frac{\hat{C} + H(1-p)}{p} &= \frac{\left(\hat{C} - \lambda + q(T+\delta) + (1-p)H\right)}{p} + \frac{q(1-p)}{p^2} \end{aligned} \tag{3.36}$$

By modifying the Equation (3.36), we evaluate the λ in (3.37).

$$\lambda = \frac{q(1-p)}{p} + q(\delta + T) \tag{3.37}$$

At the intersection of transmit and non-transmit regions, we calculate the value of the threshold state from the perspective of each region. We equalize the $f(T)$ expressions obtained from each region to find a general solution. Then, we evaluate results at

$\lambda = \frac{q(1-p)}{p} + q(\delta + T)$.

$$(1-T)\left(qT - \lambda - \frac{qT}{2}\right) = \frac{\left(\hat{C} - \lambda + qT + (1-p)H\right)}{p} + \frac{q(1-p)}{p^2} \tag{3.38}$$

We rewrite the Equation (3.38) in terms of C and H.

$$\begin{aligned}
&\frac{-2\hat{C} + 2H(p-1) + pq(T-1)(2\delta + T - 2) + 2q(\delta + T - 1)}{p} = 0 \\
&\hat{C} = H(p-1) + \frac{1}{2}pq(T-1)(2\delta + T - 2) + q(\delta + T - 1) \\
&T(\delta) = \frac{3}{2} - \frac{1}{p} - \delta + \frac{\sqrt{\left(8\hat{C} - 8H(-1+p)\right)p + q\left(4 - 4p + p^2(-1+2\delta)^2\right)}}{2p\sqrt{q}}
\end{aligned} \tag{3.39}$$

To ensure the intersection operation was valid, we investigate the dummy variable δ. Since the $\frac{\partial T}{\partial \delta}$ is negative, and $T(0) > T(1)$, we can say that $T(\delta)$ is a decreasing function. Therefore, pushing back the transmit region's Threshold with δ to intersect the non transmit and transmit regions was a proper operation. We will use $\delta = 0$ to obtain the Whittle's Index.

$$\begin{aligned}
T(1) &= \frac{\sqrt{q\left[p(8C - 8H(p-1)) + (p-2)^2 q\right]}}{2pq} - \frac{1}{p} + \frac{1}{2} \\
T(0) &= \frac{\sqrt{q\left[p(8C - 8H(p-1)) + (p-2)^2 q\right]}}{2pq} - \frac{1}{p} + \frac{3}{2} \\
T &= \left\lfloor \frac{\sqrt{q\left[p(8C - 8H(p-1)) + (p-2)^2 q\right]}}{2pq} - \frac{1}{p} + \frac{3}{2} \right\rfloor
\end{aligned} \tag{3.40}$$

The Whittle's Index calculates the $\hat{C}$ for each arm (Receiver in our case) at the beginning of the transmission region. The Receiver's state is $\Delta + 1$ at the beginning of the transmission region. Therefore, we will select $T = \Delta + 1$.

$$\hat{C} = \frac{1}{2}\Delta pq(\Delta - 1) + \Delta q - H(1-p) \tag{3.41}$$

Then, the receiver with the largest $\hat{C}$ will be selected for the transmission. If the transmission attempt succeeds, $\hat{C}$ of the receiver will be reduced. Intuitively, $\hat{C}$ denotes the minimum service charge of the arm at the threshold state [13]. To obtain

maximum drop-in service charge, the Policy attempts to transmit to the receiver with the highest $\hat{C}$ at every frame.

$$\begin{aligned}\hat{C}_i(k) &= \frac{1}{2}\Delta_i(k)p_iq_i(\Delta_i(k)-1)+\Delta_i(k)q_i - H_i(1-p_i)\\ &= \frac{1}{2}\Delta_i(k)p_iq_i\left(\Delta_i(k)+\frac{2-p_i}{p_i}\right) - H_i(1-p_i)\end{aligned} \tag{3.42}$$

To ensure the validity of the Whittle's Index, we check the Indexability Property. The Indexability is defined in [13]. To provide this property, we look at the non-transmit region. In this region, as $\hat{C}$ increases, the number of states that lie in the non-transmit region must also increase monotonically. We can observe from (3.40) that the threshold increases as $\hat{C}$ increases.

There is one more requirement to prove that WIP is Indexable, which is challenging to meet in our system model. As the cost increases from 0 to infinity, the number of states in the Non-transmit region should also increase from 0 to infinity. We can see from the Equation (3.40) that Cost and Threshold increase in parallel, but when the cost is 0, we need to constrain the H value to ensure that no state remains in the non-transmit region.

To find the interval of H, we limit T that it should be less than two if $\hat{C}$ is zero, then we find the interval of H.

$$\begin{aligned}T = 1 &= \left\lfloor \frac{\sqrt{q\left[p(8H(1-p))+(p-2)^2q\right]}}{2pq} - \frac{1}{p} + \frac{3}{2}\right\rfloor\\ 2 &> \frac{\sqrt{q\left[p(8H(1-p))+(p-2)^2q\right]}}{2pq} - \frac{1}{p} + \frac{3}{2}\end{aligned} \tag{3.43}$$

To provide the conditions in (3.43), variable H must satisfy the condition in (3.44).

$$0 \leq H < \frac{q}{1-p} \tag{3.44}$$

The range we found for H limits the effect of this variable.

Intuitively, when the service charge $\hat{C}$ is zero, we expect that all states would stay in the transmission region. However, the penalty variable H may cause the Threshold to be larger than one even if there is no service charge. We can observe this from

the Bellman Equation in (3.24). When the $\hat{C}$ is equal to zero, the Bellman equations will transform to (3.45). It can be seen from the (3.45) that the cost of transmission region is increased by $(1-p)H$ with our penalty modification, and it may cause the decision-maker to select the Idling action even if there is no service charge for transmission.

$$\begin{aligned} f(\Delta)+\lambda = q\Delta + \min\{&f(\Delta+1); \\ &(1-p)f(\Delta+1)+(1-p)H)\} \end{aligned} \tag{3.45}$$

To ensure that there will be no receiver in the non-transmit region when $\hat{C}$ is zero, we redefined the Bellman Equation. Rather than assigning a penalty to the unsuccessful transmission case, we added a reward term (which is a negative penalty) to the successful transmission. With the reward term, we aimed to reduce the cost when a successful transmission has happened. We reduced the cost of successful transmission by pH. Renewed Bellman Equations are given in (3.46).

$$\begin{aligned} f(1) &= 0 \\ f(\Delta)+\lambda =& \min\{q\Delta + f(\Delta+1); \\ &\hat{C}+(1-p)(f(\Delta+1)+q\Delta)+p\left(f(1)+q\Delta-H\right)\} \end{aligned} \tag{3.46}$$

Since we are not increasing the transmission cost but decreasing it, we argue that the added term will not cause the non-transmit region to expand. At $\hat{C}=0$, we will not leave any space for the non-transmit region. Therefore, the Indexability Property can be established.

The value of the threshold from the non-transmit region's perspective is given in (3.47).

$$f(T) = (1-T)\left(qT-\lambda-\frac{qT}{2}\right) \tag{3.47}$$

The value obtained near the Threshold from the transmit region's perspective is given in (3.48).

$$f(T+\delta) = \frac{\left(\hat{C}-\lambda+q(T+\delta)-pH\right)}{p} + \frac{q(1-p)}{p^2} \tag{3.48}$$

The boundaries of the threshold's value is given in (3.50).

$$\begin{aligned} q\Delta + f(\Delta+1) &\leq \hat{C} + (1-p)\left(f(\Delta+1) + q\Delta\right) + p\left(f(1) + q\Delta - H\right) \\ f(\Delta+1) &\leq \frac{\hat{C}}{p} - H \end{aligned} \tag{3.49}$$

$$f(T) \leq \frac{\hat{C}}{p} - H \leq f(T+\delta) \tag{3.50}$$

We look up at the transmit region to find an expression for λ. The resulting equation for λ is given in (3.53).

$$\frac{\hat{C}}{p} - H = \frac{\left(\hat{C} - \lambda + q(T+\delta) - pH\right)}{p} + \frac{q(1-p)}{p^2} \tag{3.51}$$

$$\lambda = \frac{q(1-p)}{p} + q(\delta + T) \tag{3.52}$$

Then, we get an equation by approaching the intersection of the Transmit region and the Non-transmit region from both directions. Then, we obtain closed-form equations for Threshold and Cost by typing the value of λ obtained in (3.52) into the corresponding places in the equation. The resulting equation for threshold is given in (3.54), and the equation for the $\hat{C}$ is given in (3.55).

$$(1-T)\left(qT - \lambda - \frac{qT}{2}\right) = \frac{\left(\hat{C} - \lambda + qT - pH\right)}{p} + \frac{q(1-p)}{p^2} \tag{3.53}$$

$$\begin{aligned} T(0) =& \sqrt{\frac{2C}{pq} - \frac{2H}{q} + \frac{(p-2)^2}{4p^2}} - \frac{1}{p} + \frac{3}{2} \\ T(1) =& \sqrt{\frac{2C}{pq} - \frac{2H}{q} + \frac{(p-2)^2}{4p^2}} - \frac{1}{p} + \frac{1}{2} \\ T(\delta) =& \left\lfloor \sqrt{\frac{2C}{pq} - \frac{2H}{q} + \frac{(1-2\delta)^2p^2 - 4p + 4}{4p^2}} - \delta - \frac{1}{p} + \frac{3}{2} \right\rfloor \end{aligned} \tag{3.54}$$

We investigate the dummy variable δ. Since the $\frac{\partial T}{\partial \delta}$ is negative, and $T(0) > T(1)$, we can say that $T(\delta)$ is a decreasing function.Therefore, we argue that intersecting the

non transmit and transmit regions with dummy variable δ was a valid operation. We will use $\delta = 0$ to obtain the Whittle's Index.

$$\hat{C} = Hp + \frac{q(T-1)(p(T-2)+2)}{2} \tag{3.55}$$

In the end, we look at the Indexability. We can observe from (3.54) that the Threshold T is monotonically increasing with $\hat{C}$. Moreover, H can only reduce the cost corresponding to the Threshold since the coefficient of H is negative in the equation. If we select $\hat{C} = 0$ and $H = 0$, the resulting Threshold is equal to one, which satisfies the Indexability constraint. As [12] applied, we assume that if $\hat{C} < H$, then the $T = 1$. Therefore, to keep the problem Indexable, $H \geq 0$ must be provided.

The resulting Whittle's Index Policy is given in Equation (3.56). At each frame k, the base station selects the receiver i with the highest cost $C_i(k)$.

$$C_i(k) = H_i p_i + \frac{q_i p_i \Delta_i(k)(\Delta_i(k) + \frac{2-p_i}{p_i})}{2} \tag{3.56}$$

The resulting Policy is similar to Whittle's Index Policy in [12]. In [12], the system model includes throughput constraints for the receivers. Throughput incentives were added to the cost function to meet the constraints. The $H_i p_i$ term we added to the cost function can also be considered as a throughput incentive.

3.2.6 Lower Bound for the Average EAoI Function

The lower bound is a valuable tool for implementation studies. Although the lower bounds provided in this section are not for finite K, and also not valid when p_i can change in time, yet, we can use them as a benchmark to scale $J_E(\pi)$. If we use $J_E(\pi)$ for direct comparison of scheduling policies, we may fail to compare the results obtained in different channel reliabilities. Therefore, to provide a performance evaluation for the policies under various channel conditions, it may be best to measure the relative performances of the policies to a benchmark.

3.2.6.1 Lower Bound in the Proactive Serving Scenario

To obtain a Lower bound, we will use the framework provided by [13], and adapt it into our case. In Table 3.2, we give the necessary notations to make the equations more understandable. The closed-form expression of the $\Delta_{q_i}(k)$ is given in Equation 3.57.

Table 3.2: Table of Notation For Lower Bound Derivation

K	Total number of frames in within an experiment i
M	Total number of receivers
n	index of an inter-delivery time between packets
$I_i(n)$	Inter-delivery time of the n'th delivery of a packet to the receiver i
L_i	Remaining frames after the last delivery to receiver i
$A_i(K)$	Total number of transmission attempts to receiver i up to frame K
$U_i(K)$	Total number of successful transmissions to receiver i up to frame K
$u_i(k)$	Transmission succession status of the receiver i at frame k
q_i	Query Probability of the receiver i
p_i	Channel Reliability for the receiver i

$$\Delta_{q_i}(k) = d_i(k)\left[u_i(k) + (1 - u_i(k))(\Delta_i(k) + a_i(k)\right] \tag{3.57}$$

Since we aim to find a lower bound for the $J_E(\pi)$, we write the expression of the $\Delta_{q_i}(k)$ into the Average EAoI function given in Equation 3.58.

$$\begin{aligned} J_E(\pi) &= \mathbb{E}\left[\frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M} d_i(k)\left[u_i(k) + (1 - u_i(k))(\Delta_i(k) + a_i(k)\right]\right] \\ &= \mathbb{E}\left[\frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M} d_i(k)\left[u_i(k) + \Delta_i(k) - \Delta_i(k)u_i(k) + a_i(k) - u_i(k)a_i(k)\right]\right] \end{aligned} \tag{3.58}$$

The $u_i(k)$ and $a_i(k)$ are binary variables that can only take values 0 or 1. This property

enables us that we can say the $a_i(k)^2 = a_i(k)$ and the $c_i(k)^2 = c_i(k)$. Moreover, as defined in (3.3), the $u_i(k)$ is the multiplication of the $a_i(k)$ and $a_i(k)$. Then, the multiplication of $u_i(k)$ and $a_i(k)$ will be equal to the $a_i(k)^2 c_i(k)$, which is equivalent to the $a_i(k)c_i(k)$, then, to the $u_i(k)$. In the Equation (3.59), we redefine the $J_E(\pi)$ by using this result.

$$J_E(\pi) = \mathbb{E}\left[\frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M} d_i(k)\left[\Delta_i(k) - u_i(k)\Delta_i(k) + a_i(k)\right]\right] \tag{3.59}$$

The expected value of the $d_i(k)$ is equal to the q_i, and the $d_i(k)$ is independent from any other variable in the Equation (3.59). Therefore, we replace the $d_i(k)$ with the q_i. Moreover, since q_i is constant over frames, we can exclude the q_i from the summation over K frames.

$$J_E(\pi) = \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}\frac{q_i}{K}\sum_{k=1}^{K}\left(\Delta_i(k) - u_i(k)\Delta_i(k) + a_i(k)\right)\right] \tag{3.60}$$

To express the $\Delta_i(k) - u_i(k)\Delta_i(k)$ in terms of inter-delivery times of the packets, we firstly define the number of frames within an experiment in terms of inter-deliveries. The total frame number K can be expressed as the summation of all inter-deliveries plus the remaining frames between the last delivery and the last frame K. The resulting equation in (3.61) is taken from [13] and identical with the "Age Conservation Law" from [28].

$$K = (\sum_{n=1}^{U_i(K)} I_i[n]) + L_i, \forall i \in \{1, 2, \cdots, M\} \tag{3.61}$$

The sum of $u_i(k)\Delta_i(k)$ can be defined in terms of the inter-deliveries.

$$\sum_{k=1}^{K} u_i(k)\Delta_i(k) = \sum_{n=1}^{U_i(K)} I_i[n] \tag{3.62}$$

$$\sum_{k=1}^{K}\Delta_i(k) = \sum_{n=1}^{U_i(K)}\left(\frac{(I_i[n]+1)\, I_i[n]}{2}\right) + \frac{(L_i+1)\, L_i}{2} \tag{3.63}$$

$$
\begin{aligned}
\frac{q_i}{K}\left[\sum_{k=1}^{K}\Delta_i(k)-u_i(k)\Delta_i(k)\right] &= \frac{q_i}{K}\left[\sum_{k=1}^{K}\Delta_i(k)\right]-\frac{q_i}{K}\left[\sum_{k=1}^{K}u_i(k)\Delta_i(k)\right] \\
&= \frac{q_i}{K}\left[\sum_{n=1}^{U_i(K)}\left(\frac{(I_i[n]+1)\,I_i[n]}{2}\right)+\frac{(L_i+1)\,L_i}{2}-\sum_{n=1}^{U_i(K)}I_i[n]\right] \\
&= q_i\left[\sum_{n=1}^{U_i(K)}\left(\frac{(I_i^2[n]+I_i[n])}{2K}\right)+\frac{(L_i^2+L_i)}{2K}-\frac{1}{K}\sum_{n=1}^{U_i(K)}I_i[n]\right] \\
&= \frac{q_i}{2}\left[\sum_{n=1}^{U_i(K)}\left(\frac{(I_i^2[n]}{K}-\frac{I_i[n]}{K}\right)+\frac{(L_i^2+L_i)}{K}\right]
\end{aligned}
\tag{3.64}
$$

Similarly, the sum of $\Delta_i(k)$ is redefined in terms of the inter-deliveries. The $U_i(K)$ is the total number of successful arrivals to the receiver i. The index number, n, of the last arrival up to frame K is equal to the $U_i(K)$. We can say that the L_i is equal to the $K - I_i[U_i(K)]$. As the frame index K goes to infinity, the $\frac{\left(L_i^2+L_i\right)}{2K}$ term converges to 0. This result holds if the $U_i(K)$ continues to increase as the K increases. Hence, the scheduling policy must not starve any receiver to keep $U_i(K)$ increasing as K goes to infinity. A non-starving policy guarantees that the $U_i(K)$ will not stop increasing as K goes to infinity. Hence, the L_i will be constrained and could not reach infinity as K increases.

The starvation case is defined in [13]. The term "starvation" refers to a case where the scheduling policy gives up from updating a receiver forever. Hence, the packet arrival probability for this Receiver drops to zero as the frame number K goes to infinity. In the scope of the thesis, the scheduling policies are assumed to be not starving any receiver.

As K goes to infinity, the evaluation of the $\frac{q_i}{K}\left[\sum_{k=1}^{K}\Delta_i(k)-u_i(k)\Delta_i(k)\right]$ is given in the Equation (3.65).

$$
\begin{aligned}
\lim_{K\to\infty}\frac{q_i}{K}\left[\sum_{k=1}^{K}\Delta_i(k)-u_i(k)\Delta_i(k)\right] &= \frac{q_i}{2}\sum_{n=1}^{U_i(K)}\left(\frac{I_i^2[n]}{K}-\frac{I_i[n]}{K}\right) \\
&= \frac{q_i}{2K}\left[\sum_{n=1}^{U_i(K)}I_i^2[n]-\sum_{n=1}^{U_i(K)}I_i[n]\right]
\end{aligned}
\tag{3.65}
$$

We will use the sample mean and sample variance operators from the [13] to describe $I_i[n]$ and $I_i^2[n]$ in terms of their mean values.

$$\mathbb{M}\left[I_i\right] = \frac{1}{U_i(K)} \sum_{n=1}^{U_i(K)} I_i[n] \tag{3.66}$$

$$\mathbb{M}\left[I_i^2\right] = \frac{1}{U_i(K)} \sum_{n=1}^{U_i(K)} I_i^2[n] \tag{3.67}$$

$$\begin{aligned} \bar{V}\left[I_i\right] &= \frac{1}{U_i(K)} \sum_{n=1}^{U_i(K)} \left(I_i[n] - \mathbb{M}\left[I_i\right]\right)^2 \\ \bar{V}\left[I_i\right] &= \mathbb{M}\left[I_i^2\right] - \left(\mathbb{M}\left[I_i\right]\right)^2 \end{aligned} \tag{3.68}$$

We write $\mathbb{M}\left[I_i\right]$ and $\mathbb{M}\left[I_i^2\right]$ into (3.65). To do this, we modify the equation by expanding the terms with $U_i(k)$.

$$\begin{aligned} \lim_{K\to\infty} \frac{q_i}{K} \left[\sum_{k=1}^{K} \Delta_i(k) - u_i(k)\Delta_i(k)\right] &= \frac{U_i(k)}{2KU_i(k)} \left[\sum_{n=1}^{U_i(K)} I_i^2[n] - \sum_{n=1}^{U_i(K)} I_i[n]\right] \\ &= \frac{U_i(k)}{2K} \left[\frac{1}{U_i(k)} \sum_{n=1}^{U_i(K)} I_i^2[n] - \frac{1}{U_i(k)} \sum_{n=1}^{U_i(K)} I_i[n]\right] \\ &= \frac{U_i(k)}{2K} \left[\mathbb{M}\left[I_i^2\right] - \mathbb{M}\left[I_i\right]\right] \end{aligned} \tag{3.69}$$

to find an expression for the $\mathbb{M}\left[I_i\right]$, we modify the Equation (3.61) by dividing both sides with $U_i(K)$.

$$\begin{aligned} \frac{K}{U_i(K)} &= \frac{1}{U_i(K)} \left(\sum_{n=1}^{U_i(K)} I_i[n]\right) + \frac{L_i}{U_i(K)} \\ \frac{K}{U_i(K)} &= \mathbb{M}\left[I_i\right] + \frac{L_i}{U_i(K)} \end{aligned} \tag{3.70}$$

At the infinite horizon of K, a non-starving policy and $p_i > 0$ for each receiver guarantees that the $\frac{L_i}{U_i(K)}$ term converges to zero since L_i has an upper bound for non-

starving policies with $p_i > 0$, but $U_i(K)$ has no bound and continues to increase as K goes to infinity.

$$\lim_{K\to\infty} \frac{K}{U_i(K)} = \mathbb{M}\left[I_i\right] \tag{3.71}$$

We write the result of Equation (3.71) into the Equation (3.69) and obtain a solution for $\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \Delta_i(k)$ in terms of inter-deliveries.

$$\begin{aligned} \lim_{K\to\infty} \frac{q_i}{K} \left[\sum_{k=1}^{K} \Delta_i(k) - u_i(k)\Delta_i(k)\right] &= \frac{q_i}{2\mathbb{M}\left[I_i\right]} \left[\mathbb{M}\left[I_i^2\right] - \mathbb{M}\left[I_i\right]\right] \\ \lim_{K\to\infty} \frac{q_i}{K} \left[\sum_{k=1}^{K} \Delta_i(k) - u_i(k)\Delta_i(k)\right] &= \frac{q_i}{2} \left(\frac{\mathbb{M}\left[I_i^2\right]}{\mathbb{M}\left[I_i\right]} - 1\right) \end{aligned} \tag{3.72}$$

With using Equation (3.68), we can rewrite $\frac{\mathbb{M}\left[I_i^2\right]}{\mathbb{M}\left[I_i\right]}$.

$$\begin{aligned} \lim_{K\to\infty} \frac{q_i}{K} \left[\sum_{k=1}^{K} \Delta_i(k) - u_i(k)\Delta_i(k)\right] &= \frac{q_i}{2} \left(\frac{\mathbb{V}\left[I_i\right]}{\mathbb{M}\left[I_i\right]} + \mathbb{M}\left[I_i\right] - 1\right) \\ \lim_{K\to\infty} \frac{q_i}{K} \left[\sum_{k=1}^{K} \Delta_i(k) - u_i(k)\Delta_i(k)\right] &= \frac{q}{2} \left(\frac{\mathbb{V}\left[I_i\right]}{\mathbb{M}\left[I_i\right]}\right) + \frac{q}{2} \left(\mathbb{M}\left[I_i\right]\right) - \frac{q}{2} \end{aligned} \tag{3.73}$$

Therefore, the average EAoI function $J_E(\pi) = \lim_{K\to\infty} \mathbb{E}\left[J_E(\pi)\right]$ can be redefined.

$$\begin{aligned} \lim_{K\to\infty} \mathbb{E}\left[J_E(\pi)\right] = &\frac{1}{2M} \sum_{i=1}^{M} q_i \frac{\mathbb{V}\left[I_i\right]}{\mathbb{M}\left[I_i\right]} + \frac{1}{2M} \sum_{i=1}^{M} q_i \mathbb{M}\left[I_i\right] - \frac{1}{2M} \sum_{i=1}^{M} q_i + \\ &+ \frac{1}{KM} \sum_{k=1}^{K} \sum_{i=1}^{M} q_i a_i(k) \end{aligned} \tag{3.74}$$

The evaluation of the $\frac{1}{KM} \sum_{k=1}^{K} \sum_{i=1}^{M} a_i(k)$ is given in Equation (3.75). Since the policies we investigate are work-conserving, and the base station can select only one receiver in each frame, the sum of all transmission attempts are equal to K. Including q_i to the summation can't increase the result because q_i is bounded between 0 and 1. The minimum value of $\frac{1}{KM} \sum_{k=1}^{K} \sum_{i=1}^{M} a_i(k)$ can be achieved in a case where all the

transmission attempts are made to the receiver with minimum q_i.

$$\begin{aligned}
&\frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M} a_i(k) = \frac{1}{M} \\
&\geq \frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M} q_i a_i(k) = \frac{1}{M}\sum_{i=1}^{M}\frac{q_i}{K}\sum_{k=1}^{K} a_i(k)\frac{1}{M}\sum_{i=1}^{M}\frac{q_i}{K}A_i(k) \\
&\geq \frac{q_{t^*}}{M} \text{ and } t^* \triangleq \arg\min_t(q_t)
\end{aligned} \tag{3.75}$$

The lower bound for the $\frac{1}{2M}\sum_{i=1}^{M} q_i \mathbb{M}\left[I_i\right]$ was derived in [13], and the result of the derivation is given in (3.76).

$$\frac{1}{2M}\sum_{i=1}^{M} q_i \mathbb{M}\left[I_i\right] = \frac{1}{2M}\left(\sum_{i=1}^{M}\sqrt{\frac{q_i}{p_i}}\right)^2 \tag{3.76}$$

In the literature, [3] also derives a lower bound for the similar problem by utilizing a different method. The resulting lower bound is very similar to the one in [13], but has an additional term. We interpret that this term is related with the $\frac{1}{2M}\sum_{i=1}^{M} q_i \frac{\mathbb{V}[I_i]}{\mathbb{M}[I_i]}$ term which [13] discards it throughout the derivation. However, our analysis in Section 2.2.2 reveals that the effect of $\frac{1}{2M}\sum_{i=1}^{M} q_i \frac{\mathbb{V}[I_i]}{\mathbb{M}[I_i]}$ term increases dramatically at the networks with a few receivers where some of them has low p_i values. We discuss this situation in more detail at the Section 2.2.2. Therefore, we will include the additional term from [3] to provide a lower bound for $\frac{1}{2M}\sum_{i=1}^{M} q_i \frac{\mathbb{V}[I_i]}{\mathbb{M}[I_i]}$.

$$\frac{1}{2M}\sum_{i=1}^{M} q_i \frac{\mathbb{V}\left[I_i\right]}{\mathbb{M}\left[I_i\right]} \geq \frac{q_{j^*}\left(1-p_{j^*}\right)}{p_{j^*}} \text{ where } j^* \triangleq \arg\min_j \frac{q_{j^*}\left(1-p_{j^*}\right)}{p_{j^*}} \tag{3.77}$$

To sum up, the resulting lower bound for the EAoI under the Proactive Serving scenario is given in Equation (3.78).

$$\begin{aligned}
\lim_{K\to\infty}\mathbb{E}\left[J_E(\pi)\right] \geq & \frac{1}{2M}\left(\sum_{i=1}^{M}\sqrt{\frac{q_i}{p_i}}\right)^2 + \frac{1}{2M}\frac{q_{j^*}\left(1-p_{j^*}\right)}{p_{j^*}} - \frac{1}{2M}\sum_{i=1}^{M} q_i + \frac{q_{t^*}}{M} \\
& \text{where } j^* \triangleq \arg\min_j \frac{q_{j^*}\left(1-p_{j^*}\right)}{p_{j^*}} \\
& \text{and } t^* \triangleq \arg\min_t(q_{t^*})
\end{aligned} \tag{3.78}$$

3.2.6.2 Lower Bound for the Instantaneous Serving Scenario

$\Delta_{q_i}(k)$ under the Instantaneous Serving scenario is given in Equation (3.79). The Average EAoI function for this $\Delta_{q_i}(k)$ is also given in (3.80)

$$\Delta_{q_i}(k) = d_i(k)\Delta_i(k) \tag{3.79}$$

$$J_E(\pi) = \mathbb{E}\left[\frac{1}{KM}\sum_{k=1}^{K}\sum_{i=1}^{M} d_i(k)\Delta_i(k)\right] \tag{3.80}$$

Similar with the proactive serving case, we replace the $d_i(k)$ with the q_i since the expected value of the $d_i(k)$ is equal to the q_i, and the $d_i(k)$ is independent from any other variable in the Equation (3.80). The result is given in Equation (3.81)

$$J_E(\pi) = \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}\frac{q_i}{K}\sum_{k=1}^{K}\Delta_i(k)\right] \tag{3.81}$$

The authors of [13] and [3] proposes more generalized version of the lower bound that also fits to our Average EAoI function in Equation (3.81). By following similar steps in Proactive serving case, the result of Equation (3.81) can be derived as a combination of the lower bounds in [13] and [3]. The resulting lower bound is given in Equation (3.82).

$$\begin{aligned}\lim_{K\to\infty}\mathbb{E}\left[J_E(\pi)\right] \geq & \frac{1}{2M}\left(\sum_{i=1}^{M}\sqrt{\frac{q_i}{p_i}}\right)^2 + \frac{1}{2M}\frac{q_{j^*}\left(1-p_{j^*}\right)}{p_{j^*}} + \frac{1}{2M}\sum_{i=1}^{M} q_i \\ & \text{and } j^* \triangleq \arg\min_j \frac{q_{j^*}\left(1-p_{j^*}\right)}{p_{j^*}}\end{aligned} \tag{3.82}$$

3.2.7 Experiments and Results on Effective AoI

Several experiments are conducted to understand the behavior of the scheduling policies and evaluate their performances under different environments. The results of the experiments are examined in terms of AoI_{Norm} (AoI), J_E (EAoI), and the throughput.

3.2.7.1 Testing the EAoI-Aware Q-MW With Throughput Modification

In this section, we test the throughput Adjusted Whittle's index policy, which is given in (3.56). We compare this Policy with the EAoI-Aware Whittle's Index policy. The experiment is conducted in the Matlab environment. In the experiment, there were eight receivers. Half of the receivers have a channel reliability value equal to 0.9, and the other half's channel reliability is equal to 0.1. The query probabilities were also selected as 0.9 or 0.1. The detailed information about experiment setup is given in 3.3.

Table 3.3: Channel and Query Statistics-EAoI-Aware Q-MW With Throughput Adjustment Experiment

Receiver Index:	1	2	3	4	5	6	7	8
Channel Reliabilities:	0.9	0.9	0.9	0.9	0.1	0.1	0.1	0.1
Query Probabilities:	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1

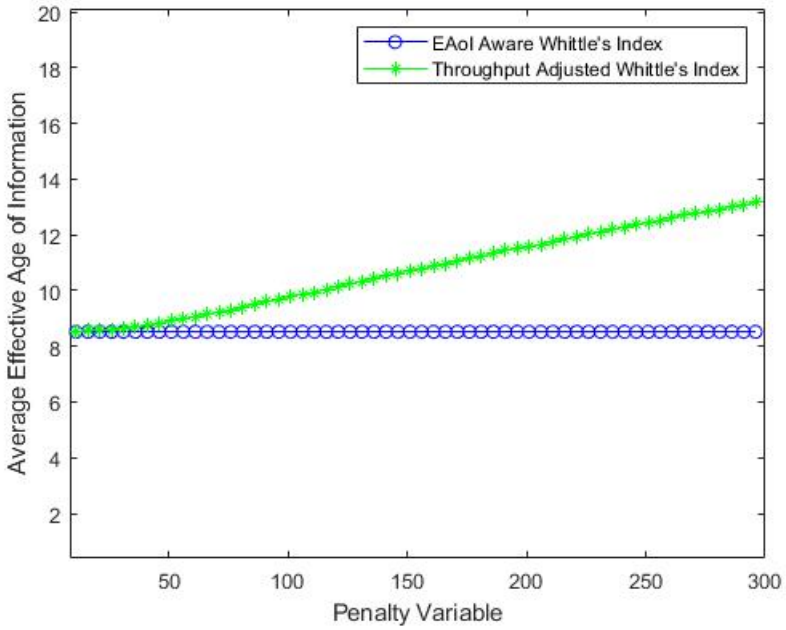

Figure 3.3: Comparison of Throughput Adjusted WIP with the EAoI-Aware WIP in terms of EAoI (MATLAB Simulation)

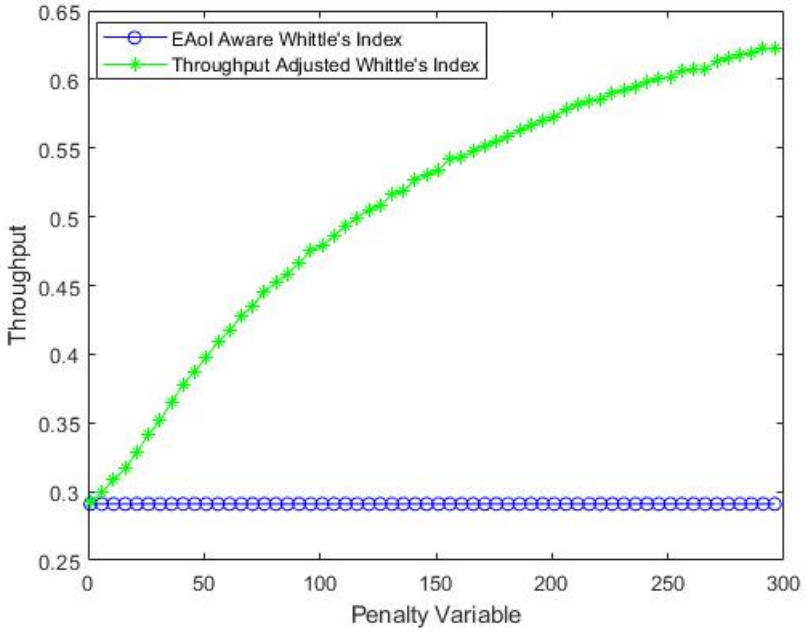

Figure 3.4: Comparison of Throughput Adjusted WIP with the EAoI-Aware WIP in terms of Throughput (MATLAB Simulation)

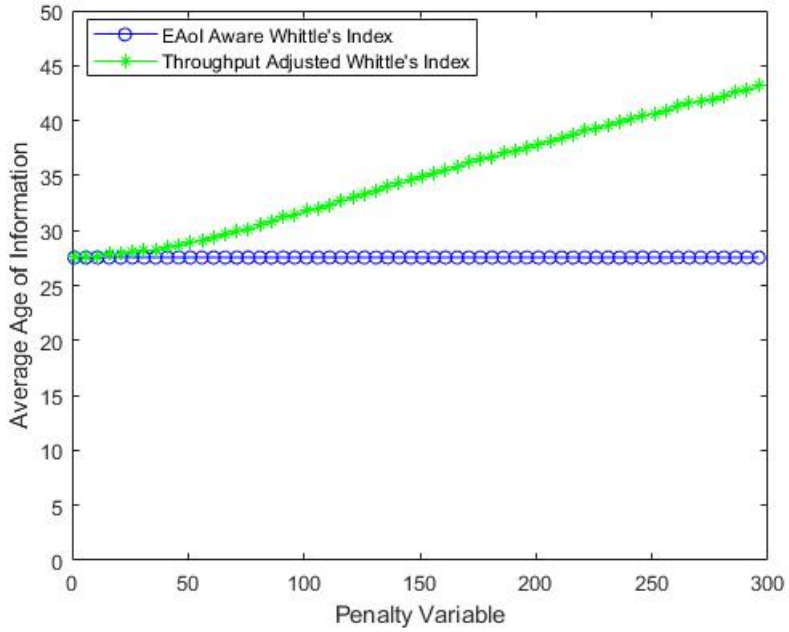

Figure 3.5: Comparison of Throughput Adjusted WIP with the EAoI-Aware WIP in terms of AoI (MATLAB Simulation)

3.2.7.2 Adjusting an Individual Receiver's Gain in SDR Network

Within the scope of the examination, 240 experiments were conducted for each of the five experiment sets, each experiment lasting $K = 7500$ frames. There were three receivers ($M = 3$) and one transmitter in the SDR network. Experiments were carried out at four different levels of the Second Receiver's Input power. At these power levels, we examined each Policy ten times. The performance of the Policy is calculated by taking the average of the ten experiments performed. Throughout the experiment sets, the query probabilities and the channel reliabilities are changed. At every experiment, we used proactive serving query response scenario. For the Q-Max Weight in the experiments, we used the EAoI-Aware Max-Weight policy derived under the instantaneous serving scenario. For the Q-Whittle's Index, we used the EAoI-Aware Whittle's Index Policy derived under proactive serving scenario.

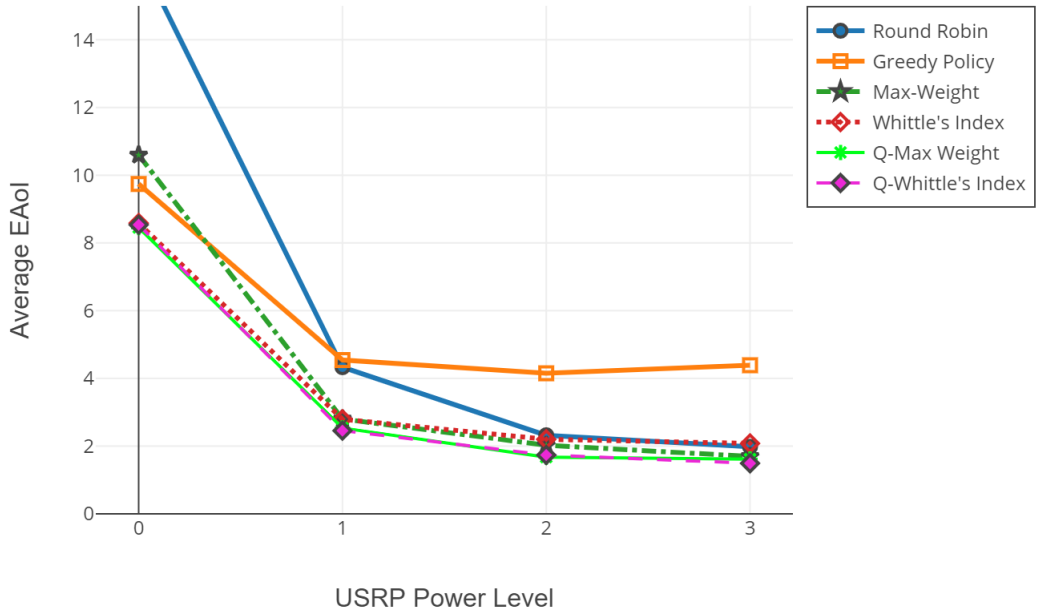

Figure 3.6: First Experiment Set-Evaluation of Effective AoI with varying Input Gain of Second Receiver (SDR Testbed)

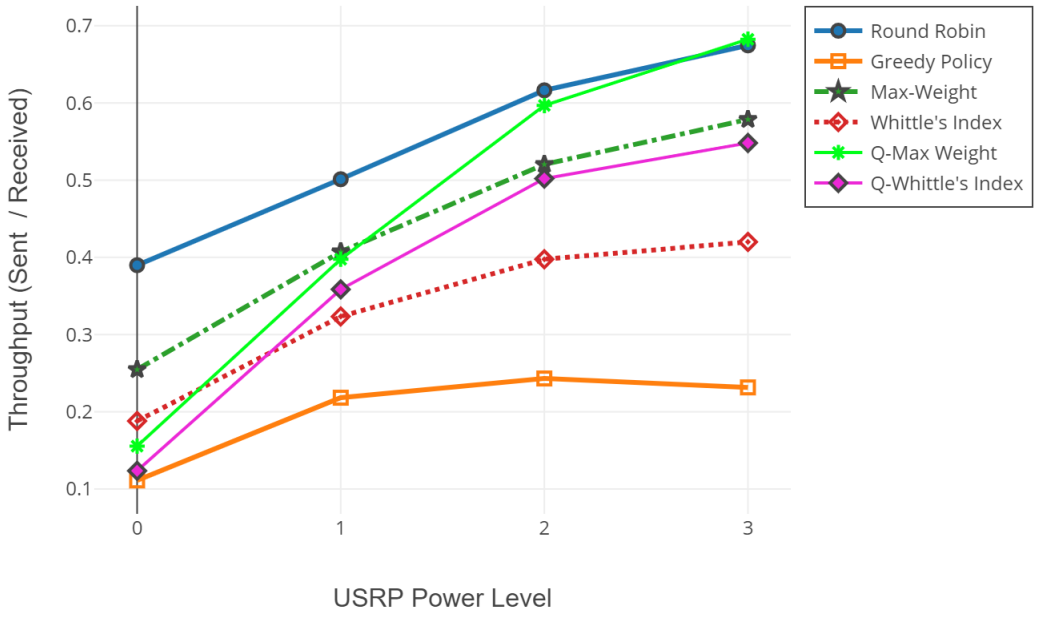

Figure 3.7: First Experiment Set-Evaluation of Throughput with varying Input Gain of Second Receiver (SDR Testbed)

Table 3.4: First Experiment Set-Channel Statistics

Gain	P1	P2	P3
0	0.9996	0.07049	0.095
1	0.9996	0.39666	0.096
2	0.9996	0.739	0.0994
3	0.9996	0.9231	0.0925

Table 3.5: First Experiment Set-Query Statistics

q_1	q_2	q_3
0.1	0.1	0.9

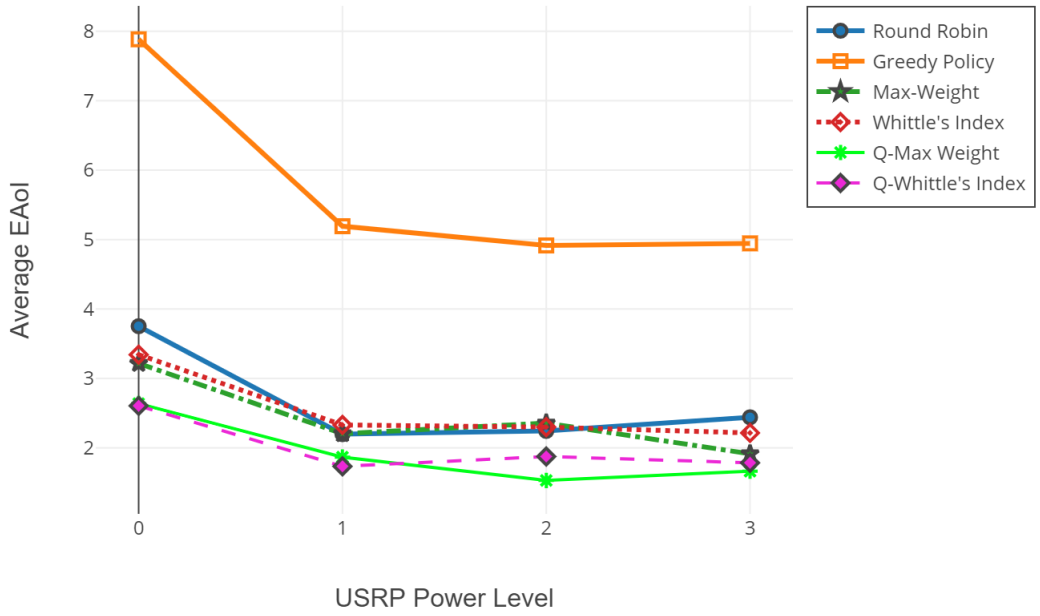

Figure 3.8: Second Experiment Set-Evaluation of Effective AoI with varying Input Gain of Second Receiver (SDR Testbed)

Figure 3.9: Second Experiment Set-Evaluation of Throughput with varying Input Gain of Second Receiver (SDR Testbed)

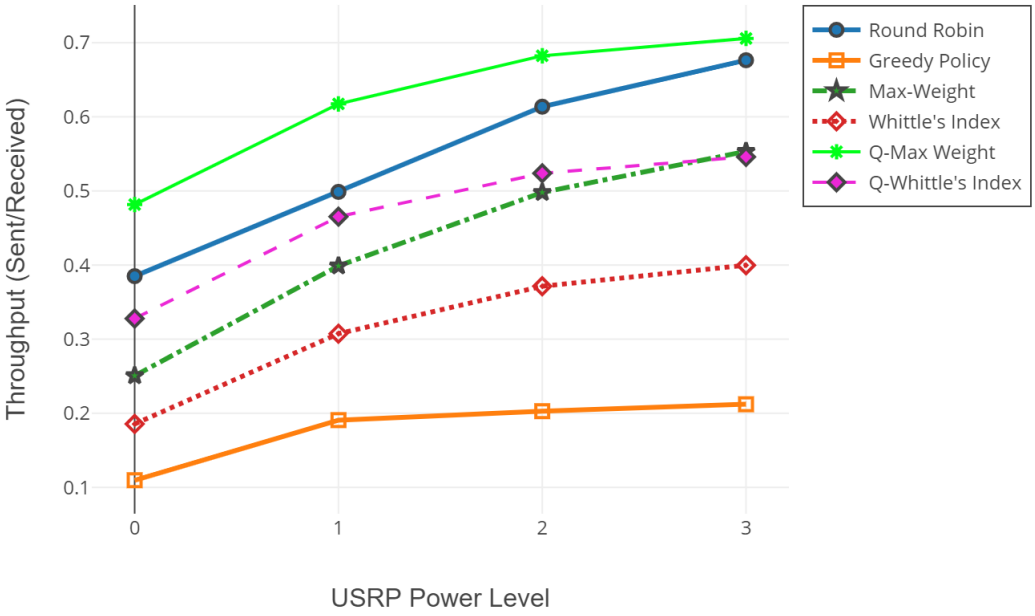

Table 3.6: Second Experiment Set-Channel Statistics

Gain	P1	P2	P3
0	0.999	0.069	0.083
1	0.999	0.413	0.080
2	0.999	0.750	0.081
3	0.999	0.934	0.081

Table 3.7: Second Experiment Set-Query Statistics

q_1	q_2	q_3
0.9	0.1	0.1

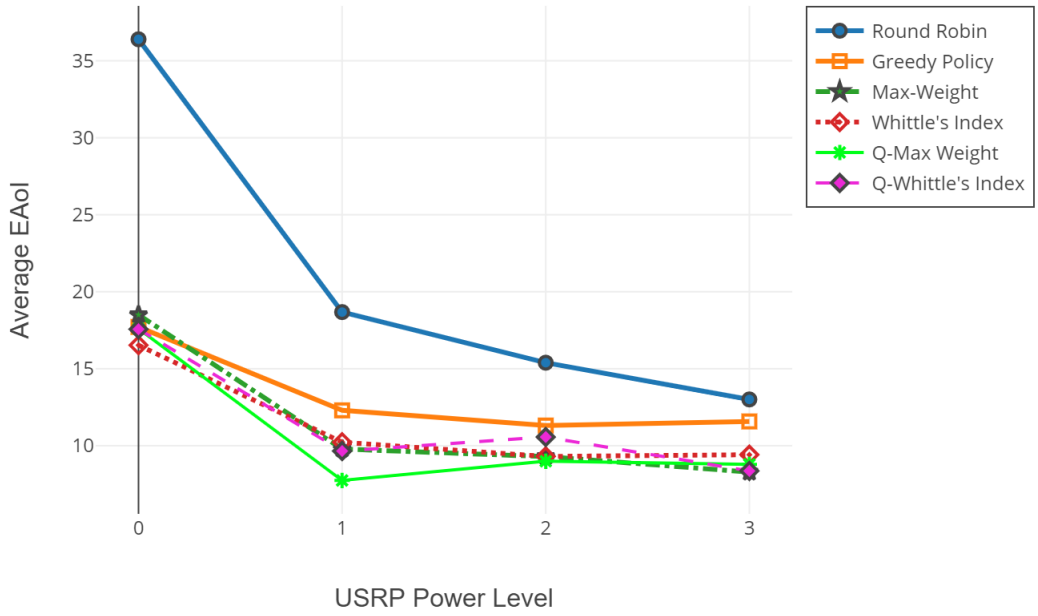

Figure 3.10: Third Experiment Set-Evaluation of Effective AoI with varying Input Gain of Second Receiver (SDR Testbed)

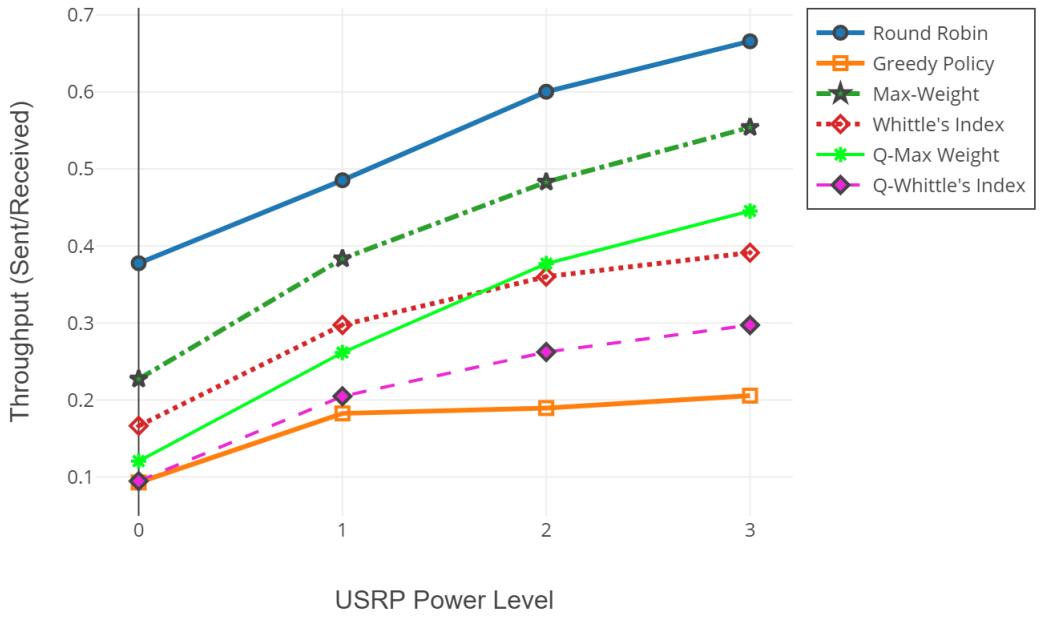

Figure 3.11: Third Experiment Set-Evaluation of Throughput with varying Input Gain of Second Receiver (SDR Testbed)

Table 3.8: Third Experiment Set-Channel Statistics

Gain	P1	P2	P3
0	0.999	0.051	0.077
1	0.999	0.369	0.077
2	0.999	0.713	0.074
3	0.999	0.913	0.0795

Table 3.9: Third Experiment Set-Query Statistics

q_1	q_2	q_3
0.1	0.9	0.9

Figure 3.12: Fourth Experiment Set-Evaluation of Effective AoI with varying Input Gain of Second Receiver (SDR Testbed)

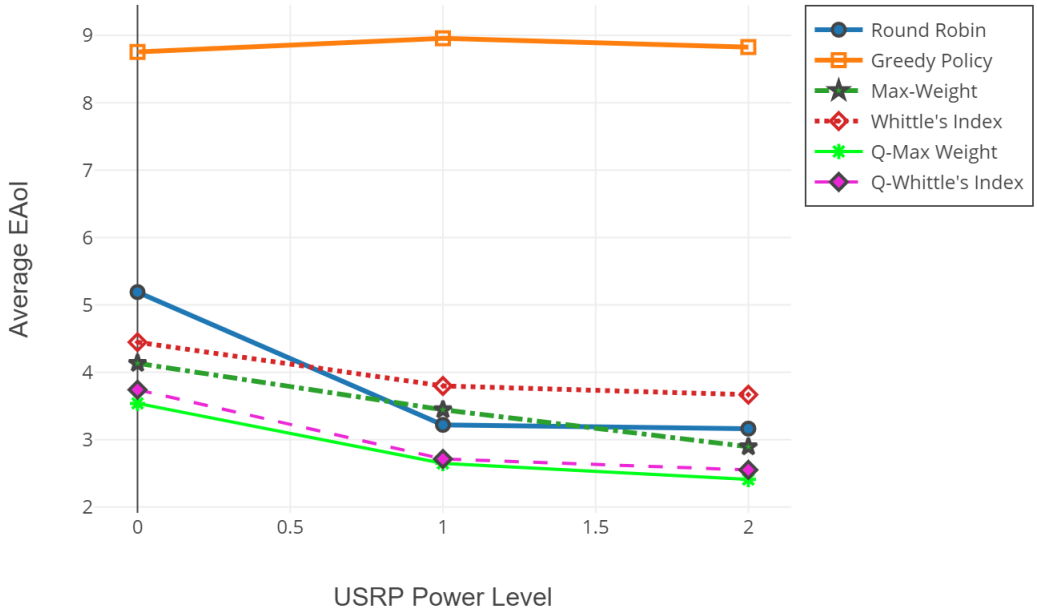

Figure 3.13: Fourth Experiment Set-Evaluation of Throughput with varying Input Gain of Second Receiver (SDR Testbed)

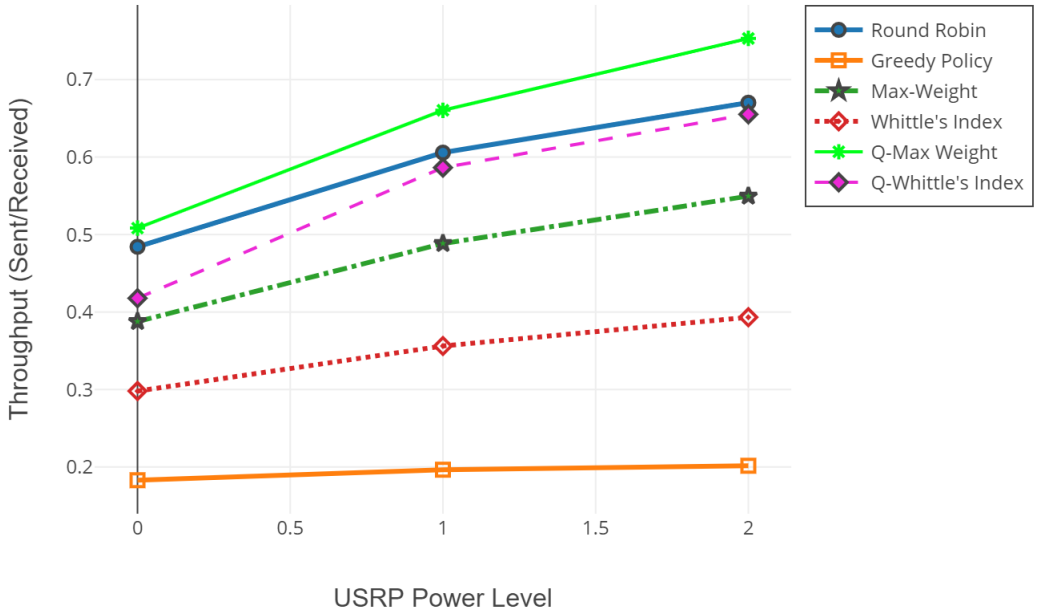

Table 3.10: Fourth Experiment Set-Channel Statistics

Gain	P1	P2	P3
0	0.999	0.363	0.078
1	0.999	0.735	0.076
2	0.999	0.930	0.077

Table 3.11: Fourth Experiment Set-Query Statistics

q_1	q_2	q_3
0.9	0.9	0.1

3.2.7.3 Adjusting the Output Power of BS in SDR Network

Within the scope of the examination, three sets of experiments are conducted where each of them consists of at least 120 individual experiments. The individual experiments run for 7500 frames. There were three receivers and one transmitter in the SDR network. Experiments were carried out at different levels of the Base Station's output gains. At these power levels, we examined each Policy at least five times. The performance of the Policy is calculated by taking the average of the experiments performed.

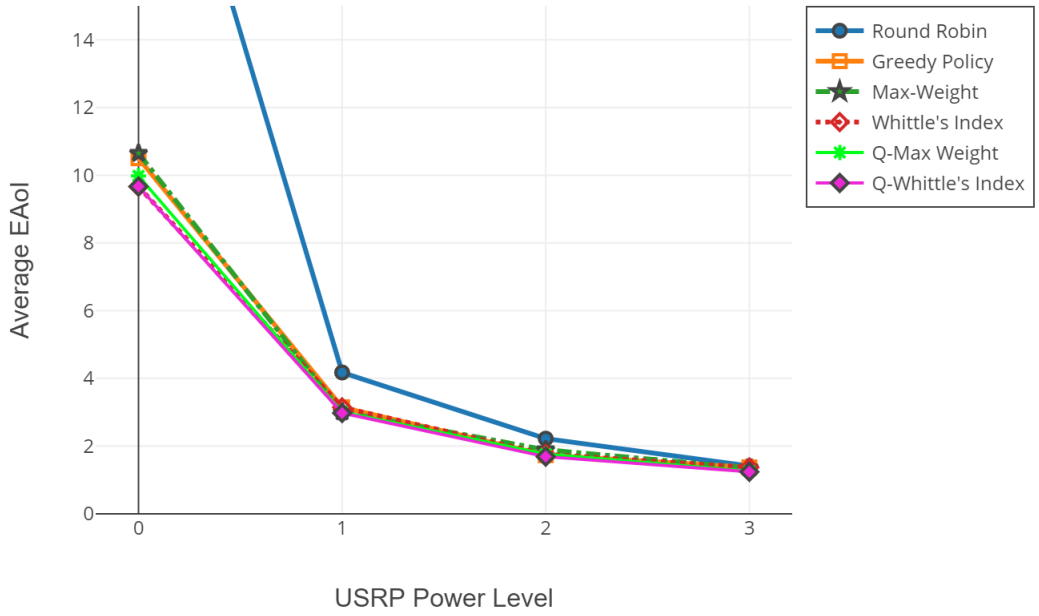

Figure 3.14: First Experiment-Evaluation of Effective AoI with varying Output Gain of BS (SDR Testbed)

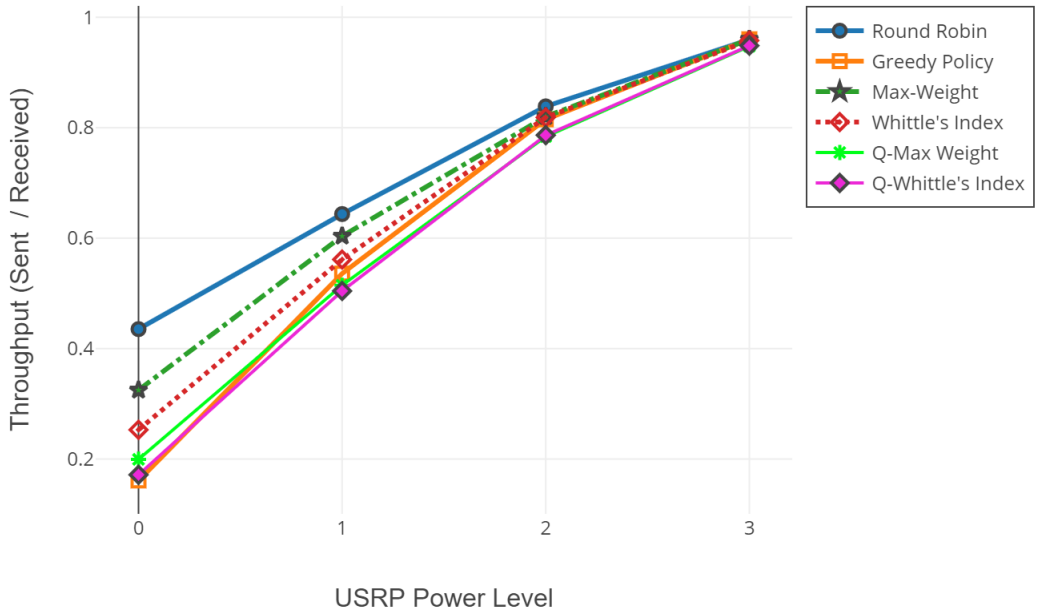

Figure 3.15: First Experiment-Evaluation of Throughput with varying Output Gain of BS (SDR Testbed)

Table 3.12: First Experiment-Channel Statistics

Gain	P1	P2	P3
0	0.999	0.075	0.231
1	0.999	0.339	0.587
2	0.999	0.662	0.854
3	0.999	0.905	0.973

Table 3.13: First Experiment-Query Statistics

q_1	q_2	q_3
0.1	0.9	0.9

Figure 3.16: Second Experiment-Evaluation of Effective AoI with varying Output Gain of BS (SDR Testbed)

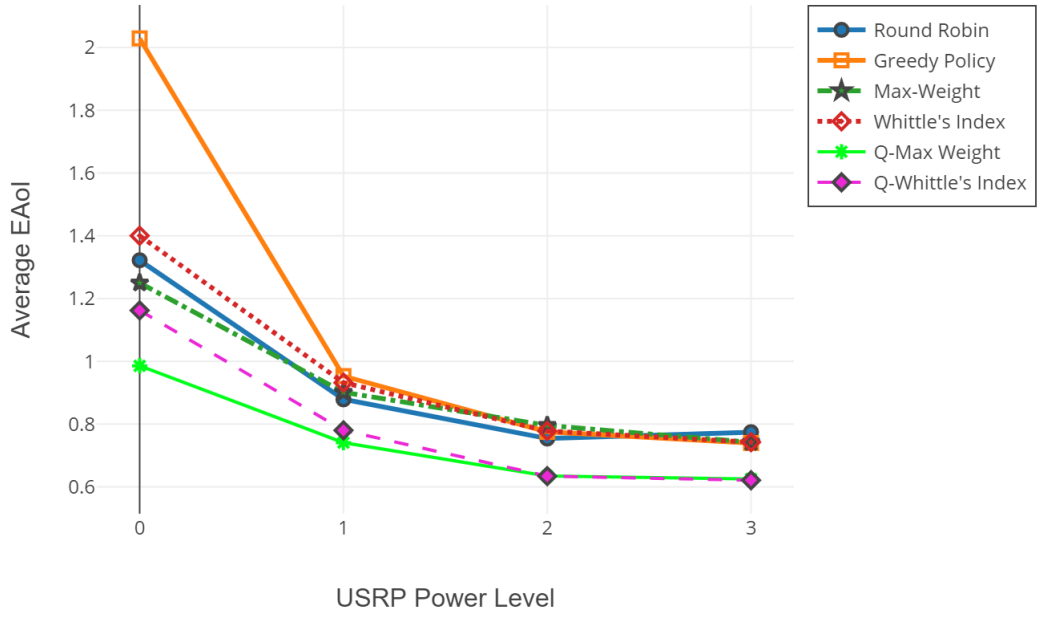

Figure 3.17: Second Experiment-Evaluation of Throughput with varying Output Gain of BS (SDR Testbed)

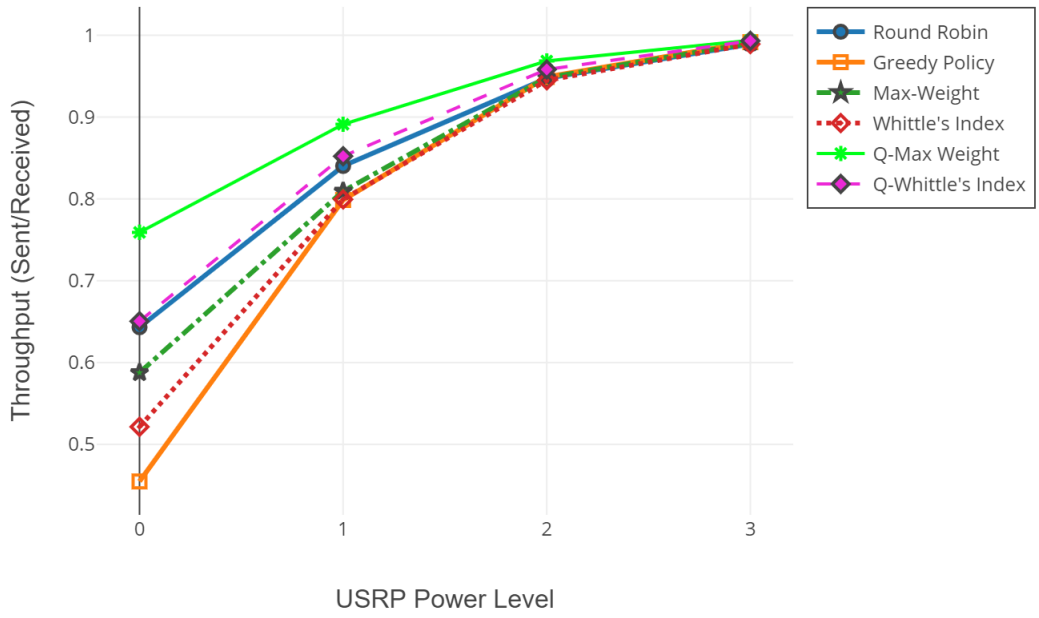

Table 3.14: Second Experiment-Channel Statistics

Gain	P1	P2	P3
0	0.9997	0.6652	0.2439
1	0.9997	0.9073	0.5983
2	0.9997	0.9830	0.8651
3	0.9997	0.9980	0.9736

Table 3.15: Second Experiment-Query Statistics

q_1	q_2	q_3
0.9	0.1	0.1

3.2.8 Interpretation of the Results

When we examine the experiments in general, we can see that the EAoI-Aware Q-MW and EAoI-Aware Q-MW Policies exhibit a more successful EAoI performance than those that do not use query information.

If we minimize the $\frac{q_i}{p_i}$ ratio, assuming we keep the sum of the q_i's unchanged, we can observe that EAoI-aware policies give better EAoI performance than AoI-Aware Policies. The test results show that the difference between EAoI policies and AoI policies narrows when a high query probability is given to the receiver with low channel reliability. The experiments show that when the query probabilities are aligned with the channel reliabilities, the EAoI-Aware Policies tend to obtain better EAoI results than AoI-aware Policies.

EAoI-Aware Q-MW Policy shows similar EAoI performance as EAoI-Aware Q-MW. However, it is seen that EAoI-Aware Q-MW policy provides higher throughput than EAoI-Aware Q-MW. This difference may be caused by the weights of the channel reliability values in the cost functions. The channel reliability in the cost function of the EAoI-Aware Q-MW Policy has a higher weight than the channel reliability in the cost function of the EAoI-Aware Q-MW. As a result, EAoI-Aware Q-MW policy is more inclined to choose the receiver with high channel reliability, which is reflected in the throughput.

3.3 Query Age of Information

Similar to EAoI, Query Age of Information(QAoI) can be used to examine pull-based scenarios. The difference between the metrics is due to their calculation methods. While calculating the EAoI metric, the instant EAoI value of the non-queried frames was taken as zero, and these values were used to calculate the average EAoI. Therefore, in the Average EAoI calculation, the sum of all Instantaneous AoIs at queried frames was divided by $\frac{1}{KM}$. The EAoI provides consistent results for the experiments performed under exact query probabilities. However, EAoI may provide misleading results under some circumstances, especially in comparing the experiments per-

formed at different query probabilities. For example, in an experimental setup with very low query probabilities, no matter how poor is the Scheduling Policy we choose, the EAoI metric will take a low value because non-query frames are included in the average as 0. In another experimental setup with very high query probabilities, since EAoI takes a value greater than 0 at query moments, no matter how good the scheduling policy we choose, the average EAoI will be higher than the experiment with low query probability. Besides the EAoI, the QAoI metric does not include non-queried frames in the average QAoI calculation and focuses on the average AoI per query.

In QAoI section, the System model and the variables are similar to the EAoI case. Throughout this section, we use the Instantaneous serving for the Query Response scenario. Unlike the EAoI model, we examine the cases where the Query arrivals are not Bernoulli-distributed random variable, and we assume that, additional information about the underlying query mechanism is available to the scheduler.

3.3.1 The Optimization Problem

The calculation of the average Query-Age of Information is given in (3.83). We define $Q_i(K)$ as the total number of queries arrived at Receiver i up to frame K.

$$J_Q(\pi) = \frac{1}{M}\left[\sum_{i=1}^{M}\frac{1}{Q_i(K)}\sum_{k=1}^{Q_i(K)}\Delta_{q_i}(k)\right] \tag{3.83}$$

The objective of the optimization problem is to minimize the expected value of average QAoI. We denote π as the Scheduling Policy among the set of scheduling policies Π.

$$\min_{\pi\in\Pi}\mathbb{E}\left[J_Q(\pi)\right], \text{ where } J_Q(\pi) = \frac{1}{M}\left[\sum_{i=1}^{M}\frac{1}{Q_i(K)}\sum_{k=1}^{Q_i(K)}\Delta_{q_i}(k)\right] \tag{3.84}$$

The calculation of $\Delta_{q_i}(k)$ under the Instantaneous Serving scenario was given in (3.5). In Equation (3.85), we argue that since the values of the non-queried frames are equal to zero, changing the upper limit of the summation from $Q_i(K)$ to K yields

the same result.

$$\sum_{k=1}^{Q_i(K)} \Delta_{q_i}(k) = \sum_{k=1}^{K} \Delta_{q_i}(k) \tag{3.85}$$

We can obtain an Average QAoI function similar to the average AoI and average QAoI functions we examined earlier by rearranging the Equation .

$$\begin{aligned} J_Q(\pi) =& \frac{1}{M}\left[\sum_{i=1}^{M}\frac{1}{Q_i(K)}\sum_{k=1}^{K} d_i(k)\Delta_i(k)\right] \\ =& \frac{1}{M}\left[\sum_{i=1}^{M}\frac{K}{Q_i(K)}\frac{1}{K}\sum_{k=1}^{K} d_i(k)\Delta_i(k)\right] \\ =& \frac{1}{M}\left[\sum_{i=1}^{M}\frac{1}{K}\sum_{k=1}^{K}\frac{Kd_i(k)}{Q_i(K)}\Delta_i(k)\right] \\ =& \frac{1}{KM}\left[\sum_{i=1}^{M}\sum_{k=1}^{K} w_i(k)\Delta_i(k)\right] \end{aligned} \tag{3.86}$$

The resulting Average QAoI function is very similar to the Average AoI function we used in Chapter 2. The average AoI with weights was extensively investigated in [13]. However, the main difference between AoI and QAoI is that the weight of the receivers $w_i(k)$ are not constant in the QAoI case and may vary among frames. The calculation of the $w_i(k)$ is given in (3.87).

$$w_i(k) = \frac{Kd_i(k)}{Q_i(K)} \tag{3.87}$$

Note that if we consider the query arrivals are i.i.d. Bernoulli-distributed random variables as we assumed earlier in the EAoI case, the $w_i(k)$ would be equal to the one at every frame. Therefore, the resulting problem would be identical to the AoI minimization problem. The evaluation of $w_i(k)$ under the i.i.d Bernoulli distributed

query arrival scenario is given in (3.88).

$$\begin{aligned}
&\lim_{K\to\infty} \mathbb{E}\left[\frac{Q_i(K)}{K}\right] = q_i \\
&\lim_{K\to\infty} \mathbb{E}\left[d_i(k)\right] = q_i \\
&\lim_{K\to\infty} \mathbb{E}\left[\frac{K}{Q_i(K)} q_i\right] = \frac{q_i}{q_i} = 1
\end{aligned} \tag{3.88}$$

We define the elapsed number of frames between $n-1$th and nth Query arrivals for receiver i as $I_{Q_i}[n]$. Similar to the age conservation law in [28], we define the "query conservation law". We argue that sum of all query inter-arrivals and the remainder term L_i will be equal to total number of frames K.

$$K = (\sum_{n=1}^{Q_i(K)} I_{Q_i}[n]) + L_i, \forall i \in \{1, 2, \cdots, M\} \tag{3.89}$$

We define the sample mean operator for the query inter-arrivals in (3.90).

$$\mathbb{M}\left[I_{Q_i}\right] = \frac{1}{Q_i(K)} \sum_{n=1}^{Q_i(K)} I_{Q_i}[n] \tag{3.90}$$

We take the limit of the $\frac{K}{Q_i(K)}$ as K goes to infinity. Throughout the chapter, we assume that query probabilities of the receivers are greater than zero. Under this assumption, we argue that $\frac{L_i}{Q_i(K)}$ term will become zero as K goes to infinity. Therefore we discard this term and calculate the mean time between query inter-arrivals $\mathbb{M}\left[I_{Q_i}\right]$ in Equation (3.91).

$$\begin{aligned}
&\lim_{K\to\infty} \frac{K}{Q_i(K)} = \frac{1}{Q_i(K)} \sum_{n=1}^{Q_i(K)} I_{Q_i}[n] + \frac{L_i}{Q_i(K)} \\
&\lim_{K\to\infty} \frac{K}{Q_i(K)} = \frac{1}{Q_i(K)} \sum_{n=1}^{Q_i(K)} I_{Q_i}[n] \\
&\lim_{K\to\infty} \frac{K}{Q_i(K)} = \mathbb{M}\left[I_{Q_i}\right]
\end{aligned} \tag{3.91}$$

3.3.2 Scheduling Policies

Throughout this section, we examine the variation of the QAoI metric in the case of a Markovian Query arrival process, assuming the behavior of the Query Source is

aligned with an Ergodic Markov Chain. In the Markov chain, we assume that there a class of states that represents the arrival of the query. We denote states in this class as the **Query State**. The query is generated only when the Query source is in the Query States and not generated when the Query Source is in other states. Throughout this section, we assume that the Scheduler knows the current state of the query source. Moreover, we assume that there is only one Query State in the Markov Chain.

For scheduling decisions, we use the Max-Weight Policy, which was derived in [13] and used for AoI minimization. We were also used Max-Weight for EAoI minimization in 3.2.4. In the AoI and EAoI cases, the weight was a constant value that does not vary through frames. In this section, we modify this variable according to (3.87), by assigning different weights to the states of the Markov Chain. Therefore, as a result of our modification the weights of the receivers may vary through frames.

In our case, the Max-Weight Policy will aim to minimize the average Lyapunov Drift by attempting the eliminate the receiver i with maximum $C_i(k)$ at every frame k.

$$C_i(k) = w_i(k)p_i\left(\Delta_i^2(k) + 2\Delta_i(k)\right) \tag{3.92}$$

Table 3.16: Table of Notation For Query-AoI

$Q_i(K)$	total number of queries arrived to receiver i up to frame K
$t_{\hat{s}}(s)$	Expected number of frames from state s to reach the Query State $\hat{s}$
$P(s)$	Steady State probability of state s
$P(\hat{s})$	Steady State probability of the query state
q_i	Query Probability of the receiver i

As we mentioned earlier, we assume that the queries are generated from an Ergodic Markovian Source. We calculate the weight parameter of Max-Weight for each state s of the Markov Chain. To calculate $w_i(k)$, we use the mean passage times and the steady-state probabilities. We assume that $\frac{Q_i(K)}{K}$ is aligned with the average query probability for the receiver i throughout K frames. In the Markov Chain domain, the average query arrival probability is equal to the steady state probability of the Query

State.

$$\lim_{K\to\infty} \mathbb{E}\left[\frac{Q_i(K)}{K}\right] = P(\hat{s}) = \frac{1}{\mathbb{M}\left[I_{Q_i}\right]} \tag{3.93}$$

We calculate $\mathbb{E}\left[d_i(k)\right]$ for each non-query state. We assume that $\mathbb{E}\left[d_i(k)\right]$ is related with the mean passage time of the current state s to the Query State $\hat{s}$. If the query source is in the Query State, we assume that $\mathbb{E}\left[d_i(k)\right]$ is related with the mean recurrence time.

$$\lim_{K\to\infty} \mathbb{E}\left[d_i(k)\right] = \frac{1}{t_{\hat{s}}(s)} \tag{3.94}$$

Then, we evaluate the $w_i(k)$ for state s in (3.95).

$$\lim_{K\to\infty} \mathbb{E}\left[\frac{K}{Q_i(K)} d_i(k)\right] = \frac{1}{P(\hat{s}) t_{\hat{s}}(s)} \tag{3.95}$$

In the literature, [4] used a system model that includes a deterministic and periodic query arrival process. We can analyze this case as a Markov Chain with state transition probabilities equal to one. Although the periodicity violates our ergodic chain assumption, we argue that we can determine the weights of the states in this process as we did in the stochastic arrival case. In this case, mean passage time $t_{\hat{s}}(s)$ is not stochastic anymore and perfectly known for every state s. Assuming that only one query arrives at every period, and the period of the query arrival process is equal to P, then the the expected query probability will be equal to $\frac{1}{P}$.

3.3.3 Experiment Results

We conducted experiments in a simulation environment to examine how the QAoI-Aware Max Weight policy performed within the scope of the QAoI metric. Throughout the experiments, we compared the QAoI-Aware Max-Weight and the traditional AoI-Aware Max-Weight from [13] with unity weight throughout frames. We assumed that self-transition probabilities of the Query States are zero, which means that consecutive query arrivals will not occur.

3.3.3.1 Experiments on Stochastic Query Arrival Process

We firstly investigate the Stochastic Query Arrival Case. We used the Markov Chain in 3.18 for the Query arrival process. In this Markov Chain, third state is assumed to

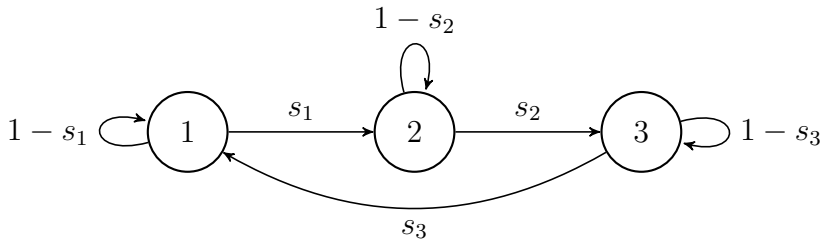

Figure 3.18: Markov Chain with one Query State (3) and two Non-Query States (1 and 2)

be the Query State.

In the first experiment, we simulated a network consisting of 20 receivers. The channel reliabilities of the receivers were uniformly distributed between 0.05 and 1. Throughout the experiment, we changed the transition probability of the first state s_1. The transition probability of the second state s_2 was constant and equal to 0.2, and the transition probability of the Query state was also constant equal to one, which indicates that consequent queries will not occur. The evaluation of AoI in the first experiment is given in 3.20, and the evaluation of QAoI in the first experiment is given in 3.19.

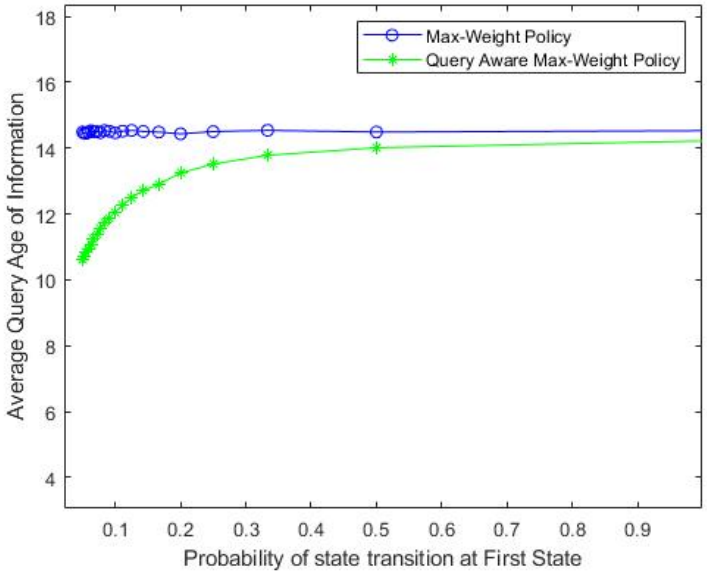

Figure 3.19: Evaluation of QAoI in the Experiment Set 1 (MATLAB Simulation)

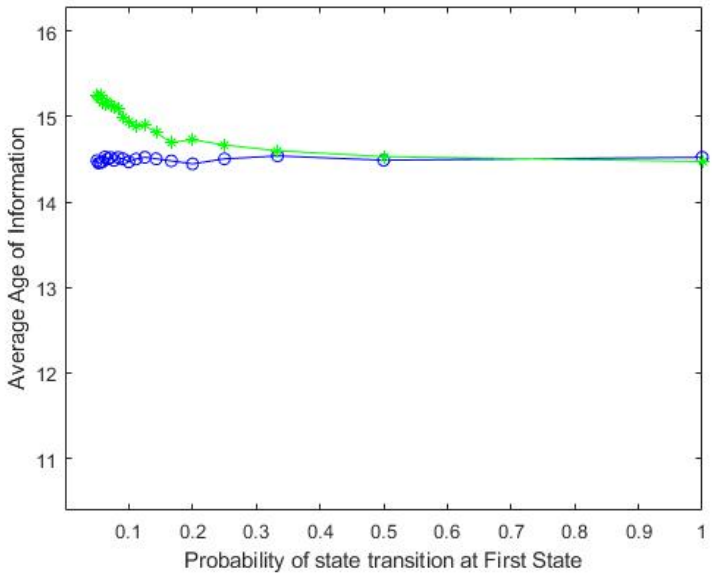

Figure 3.20: Evaluation of AoI in the Experiment Set 1 (MATLAB Simulation)

The results show that, by utilizing the weights, we were able to reduce QAoI. Decreasing the transition probability of the first state in our model increases the QAoI performance. We interpret this result as the Scheduling Policy performs better when the total number of queries per receiver is low and query arrivals are more infrequent.

In the second experiment, we altered the number of receivers in the network. The channel reliabilities of the receivers were uniformly distributed. The transition probability of the first state s_1 was constant and equal to 0.1. The transition probability of the second state s_2 was also constant and equal to 0.2. The transition probability of the Query state was equal to one. The evaluation of AoI in the second experiment is given in 3.22, and the evaluation of QAoI in the second experiment is given in 3.21.

The gap between the Max-Weight and Query-Aware Max-Weight widens by increasing the number of receivers in the network. As the average AoI increases in the network with increasing number of receivers, the difference between AoI and QAoI becomes more visible.

In both experiments, the AoI performance of Query Aware Max Weight is worse than the traditional Max-Weight. This is because Query Aware Max Weight Policy tries to improve QAoI performance by sacrificing AoI performance.

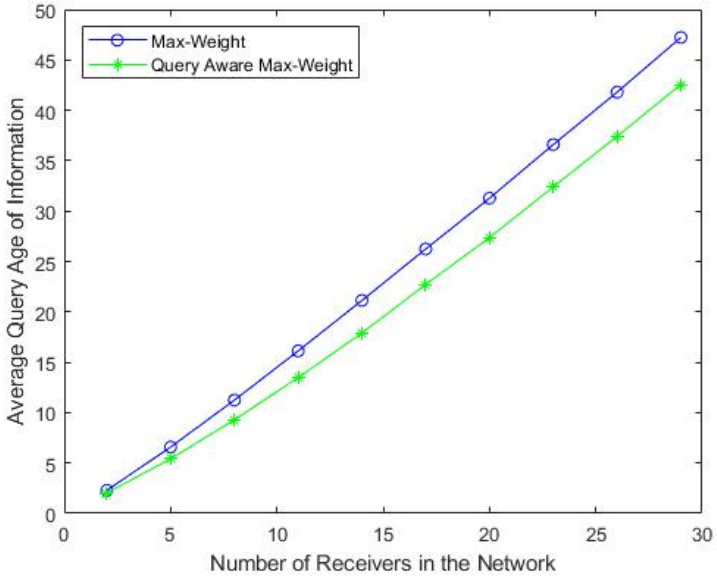

Figure 3.21: Evaluation of QAoI in the Experiment Set 2 (MATLAB Simulation)

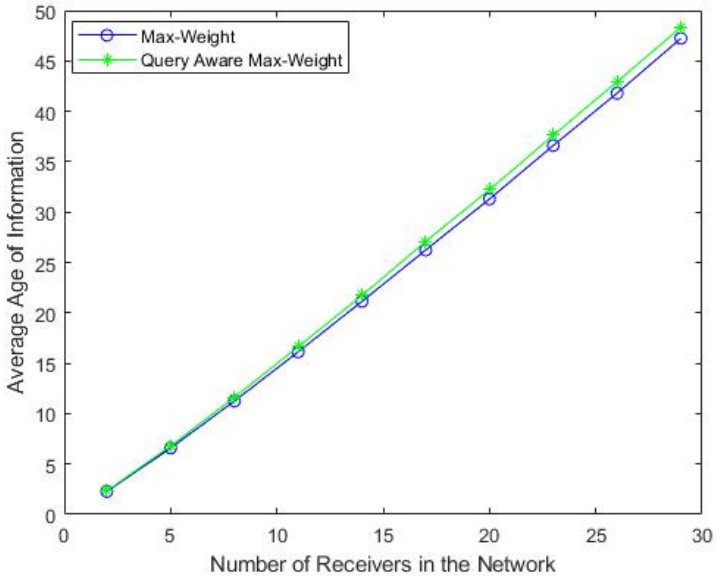

Figure 3.22: Evaluation of AoI in the Experiment Set 2 (MATLAB Simulation)

3.3.3.2 Experiments on Periodic Query Arrival Process

The query process is deterministic in the Periodic Arrival case. In this section, we shared the Average AoI and Average QAoI results of the third experiment.

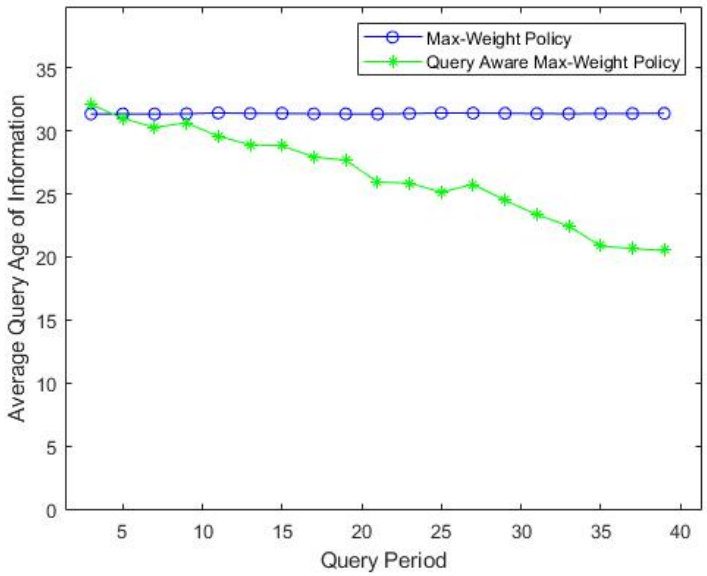

Figure 3.23: Evaluation of QAoI in the Experiment 3 (MATLAB Simulation)

The results of the third experiment indicate that decreasing the query period reduces the performance of QAoI-MW Policy. If we interpret the Period as a Markov chain containing as many states as the period width, an increase in the period width can be interpreted as an increase in the number of states. As the number of states increases, the weights of the MW are more quantized. The more precise distribution of weights could explain the performance increase of the AoI-Aware Max-Weight Policy. Moreover, as we observed in the first experiment, increasing the query inter-arrival times may positively effect the performance of QAoI-MW.

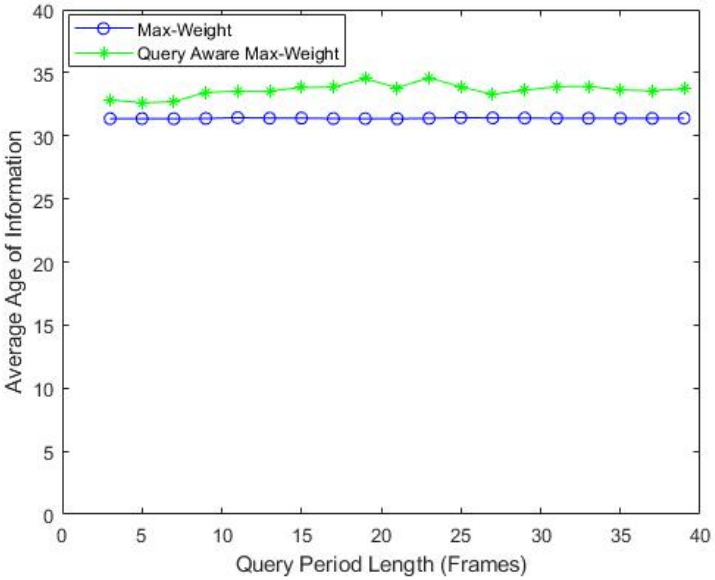

Figure 3.24: Evaluation of AoI in the Experiment 3 (MATLAB Simulation)

CHAPTER 4

IMPLEMENTATION

4.1 Background Information

Research on theoretically designed system models may not yield the same results in real communication systems. The implementation studies intend to discover the details that the real system could observe but the theoretical model miss.

4.1.1 Software Defined Radios

Software-based radios, which are known as "Software radios" in pioneering studies [19], are radios whose some of the physical layer features such as center frequency, bandwidth, coding of the communication system can be changed by modifying the software [5]. Since these devices enable us to intervene in the communication system starting from the Physical Layer, one can alter various parameters and protocols from the physical layer to the application layer. These radios play an essential role in developing today's technologies that require communication across multiple frequency bands and standards.

Allowing changes to be made in the Physical Layer and Data Link Layer makes Software-Dased Radios particularly important. Without the Software Defined Radios, it would be necessary to modify the hardware to make such a change, which creates a significant burden in terms of time and cost. Thanks to Software-Defined Radios, one can change the Physical Layer or Data Link layer by simply changing the device's software. This changeable process enables rapid prototyping of new designs [6].

We used Ettus USRP N210, NI USRP 2920, and NI USRP 2930 SDR devices in

the implementation. The overview of a NI USRP 2920 taken from [22] is given in 4.1. The USRP 2920 has one transmit module and one receive module, which are independent of each other. For this reason, the USRP2920 can be used as both a transmitter and a receiver at the same time. However, it is not possible to execute two receive operations or two transmit operations simultaneously. The device has two RF ports. The signal from RF ports can be switched inside the USRP. This switch structure allows both ports to be defined as receivers but does not allow both ports to be defined as transmitters. The connection of the device with the host computer is provided via a 1 GB Ethernet link. In-Phase and quadrature components of the signal are carried over this link. In a Transmission scenario, Ethernet packets coming from the host containing In-Phase and Quadrature signal information are converted to Analog signal in USRP. The resulting signal is filtered, then, In-Phase and Quadrature signals are mixed and sent to the Transmit Amplifier. The output gain of this amplifier can be controlled by software. The signal at the amplifier output is given to the antenna from the TX1 port. In the Receive scenario, the signal from the antenna is transmitted to the Receive module over the RF switch. Within this module, the signal first passes through the LNA. It then passes through an Amplifier whose gain can be controlled by software. In the next step, the signal is split into In-Phase and Quadrature components, filtered and digitized, then sent to the Host Computer within the ethernet frame [22].

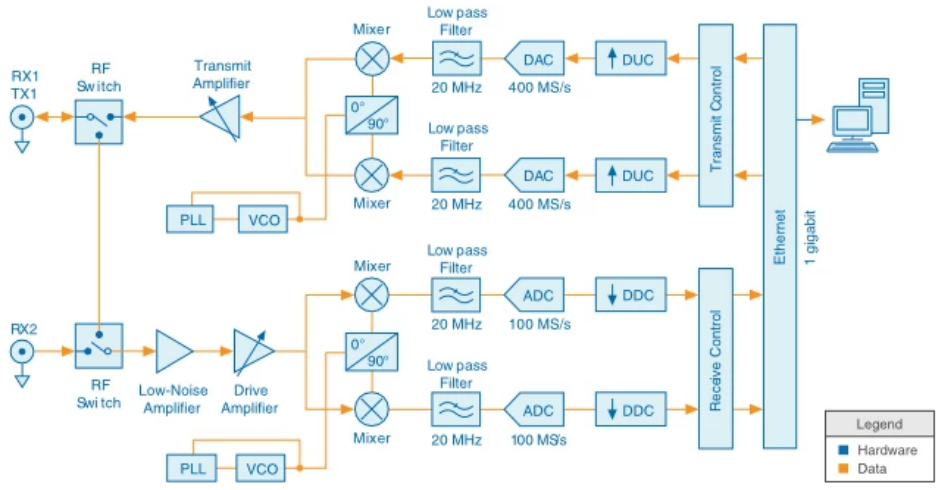

Figure 4.1: Overview of the NI USRP2920

4.2 Implementation of the network

4.2.1 Overview

To implement the network and the scheduling methods, we used four USRP Software-Defined Radios. One of the USRPs was the base station, and the other three were the receiver modules. The radios were communicating with the host computer via Ethernet connection. Synchronization problems are thus avoided as all USRPs are managed from a single host computer. The system is controlled on a Labview session interfacing with the USRP devices. The base station and receiver modules work on different threads in the same Labview session. Time is discretized in 50ms-length frames, and Labview keeps track of the frame number, i.e., the total number of frames passed since the system started. The frame number is the reference clock of the system. At each frame, a message with a timestamp is prepared in Labview and sent to the Transmitter USRP, which then sends it to the air interface. In the meantime, receiver USRPs collect data from the air interface. This data is sent to the Labview session and analyzed. The succession of the Transmission is tracked in Labview. The overview of the parameters in our implementation is given in 4.1.

An overview of the SDR network is given in Figure 4.2. The packet content created in the Labview environment on the host computer is sent to the base station via an Ethernet connection after being modulated with QPSK modulation. The base station broadcasts the constructed signal on the air interface. Receivers listen to the air interface and send the signal they obtain to the Host computer via Ethernet connection. Demodulation of the signal and error detection with CRC is made via the Labview program at the Host Computer. Messages without errors are used in the calculation of AoI. The USRP receivers are located at different distances from the base station for obtaining asymmetric channel reliabilities due to significantly different path losses. Experiments are thus aimed to be conducted under diverse channel reliabilities.

We used QPSK modulation at the air interface. In QPSK, each symbol carries two bits of information. Maximum operating frequency of USRP-2920 is 2.2GHz [20]. To cause higher path loss, we aimed to choose the center frequency of the carrier signal as large as possible. Therefore we choose Center Frequency as 1.9GHz, and

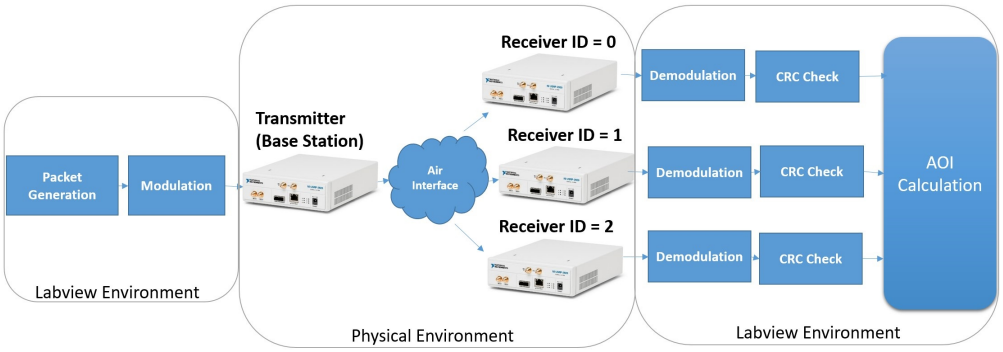

Figure 4.2: Overview of the Implementation

Table 4.1: Overview of Parameters

Modulation:	Quadrature Phase-Shift Keying
Center Frequency:	1.9GHz
I/Q Rate:	500k Samples/sec
Sample Rate:	400k Samples/sec
Bandwidth:	200kHz
Bits Per Symbol:	2
Samples Per Symbol:	8
Duration of one Frame:	50ms

this value is identical for all receivers.

We can configure the sampling rate of the USRP via LabVIEW. This configuration is presented in the USRP documentation under the name I/Q rate [20]. NI states that I/Q Rate must be multiplied by 0.8 to convert to the sampling rate [21]. In our experiments, we used 500k Samples/sec as the I/Q rate. This corresponds to a sampling rate of 400k Samples/sec, hence a bandwidth of 200kHz. This bandwidth is more than enough for our application. If higher I/Q rates are selected, an increase in bandwidth can be achieved, but an increase in the number of samples processed per second can both cause slowdowns in the internal order of the USRP and cause more load on the Ethernet line between the Host Computer and USRP. Since the time-sensitivity in AoI calculations is susceptible, we kept the I/Q Rate low to obtain a more deterministic working system and not to overload the USRP.

4.2.2 Packet Interface

Each packet sent by the base station consists of the message, cyclic redundancy check (CRC), Guard and Synchronization fields. The structure of a packet is illustrated in Figure 4.3.

6 guard bits are placed at the beginning of the packet. The Pulse Shaping Filter, which puts digital signals in a transmittable form, discards these 6 bits. These bits that do not carry any information are placed to prevent the pulse shaping filter from destroying the bits that carry information.

Implementation includes a 30-bit synchronization sequence known to both the sender and receiver modules. The Receiver, which continuously samples the air interface, can detect the beginning of the packet with the help of this sequence. This 30-bit series is formed with a function provided by Labview that generates pseudorandom data in the Galois domain.

The semantic information in the packet is carried in the message area. The message area consists of two parts: Receiver ID (RX ID) and Packet ID. In our implementation, since all receivers listen to the same frequency, the packet sent by the Transmitter to the air interface reaches all receivers, provided that the channel conditions are

appropriate. However, in the system model we examined, each packet has only one Receiver. For this purpose, there is a Receiver ID field in the packet, which contains the Receiver's index that the Base Station sent the packet. RX ID is a 4-bit sequence used as a unique identifier for the receivers (note that the number of receivers in our experiment is only 3, which is below 16). Upon receiving a packet, each Receiver compares this field with its own RX ID and decides whether the packet belongs to it. If the RX IDs of the packet and Receiver do not match, the Receiver discards the packet, and the $\Delta_i(k)$ for this Receiver is increased in the next frame.

As mentioned in 4.2.1, the frame number is used as a reference clock of the system. It starts with one at the beginning of each experiment and increases by one at every frame. We used the frame number as a packet timestamp. We used Frame Number as packet timestamp and AoI calculation tool since we considered it as the common time reference of the whole system. With the Packet ID field, we carry the generation time of the packet. The Generation time of the packet indicates the Frame number of the system when the packet is formed. Upon the reception, the Receiver gains information about the generation time of the packet. As the Receiver also knows the actual frame number, the instantaneous Age of Information $\Delta_i(k)$ can be computed upon the reception of a packet as the difference between the actual frame number and the Packet ID.

Due to noise, there may be distortions in the packets received by the Receiver. We must discard the corrupted package. Otherwise, these distortions may cause us to be unable to measure the AoI of the network or to mismeasure it. 16-bits Cyclic Redundancy Check (CRC) is used in the implementation to detect corrupted packets. The message field of the packet is given as an input to the 16-bits CRC. When the Receiver obtains the packet, it passes the message part of the packet through the CRC and compares the results with the CRC field in the packet. If they match, the message is considered received without error, and the number of successful CRC checks for that Receiver increases by one. In the system, this value is kept for all receivers and updated at each frame. The total number of correct CRC checks for Receiver i is divided by the actual frame number of the system to calculate the channel reliability p_i of the Receiver i. If the system's configuration is not changed, i.e., the transmitter output signal gain, operating frequency, or the locations of the USRP's

are not changed, the channel reliabilities are assumed to converge. Since MW and WIP policies take channel reliability as input, it is essential to calculate this value precisely. In the implementation, we calculated the channel reliabilities dynamically throughout the experiment. To obtain a value of channel reliability at the start of the experiment, we pre-run the experiment setup for 100 frames. During these frames, only the p_i values are initialized without making AoI calculations. In this way, we take a precaution against the incorrect estimate of channel reliability at the start of the experiment.

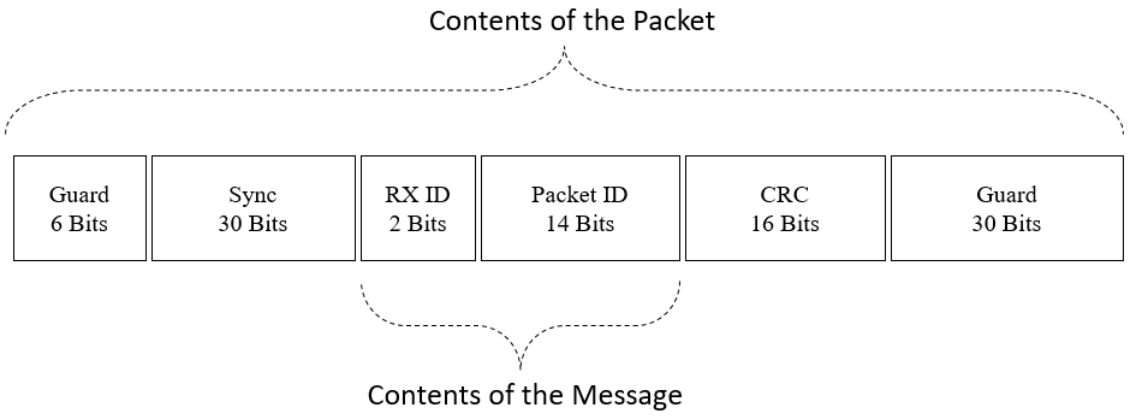

Figure 4.3: Packet content in the air interface

4.2.3 Runtime of the SDR Network

Labview environment contains useful built-in functions for communication system implementation. We used them frequently in our study. We also benefited from the examples regarding the PSK-Modulated communication system and packet-based digital link examples provided by the Labview and the Labview Community [24].

The working mechanism of the system is given in the figure 4.4. Since the Labview program allows multi-threading, we were able to execute processes independently in different threads. Receiver modules, the Transmitter module, and the Logging Module are implemented as separate threads in the program. In this way, we were able to carry out these operations simultaneously. Labview program has the feature of providing synchronization between threads. By putting this feature into use, it was

possible to organize processes running in different threads and should be consecutive to each other.

With the update of the frame number, we consider that the new frame has begun. Receivers, each running in a separate thread, perform the data collection process from Receiver USRPs at the start of the frame. Meanwhile, the Transmission thread calculates the instantaneous AoI's of the receivers with the data obtained in the previous frame. AoI information and channel statistics are the inputs of the scheduling policy. The policy reports the Receiver selected in the new frame as output. This information is written in the "Receiver ID" field in the package created for the Transmission. The current frame number is also written in the "Packet ID" field. After completing the content, the packet is modulated using Labview's Modulation function and prepared for Transmission. The transmission thread transmits this data over Ethernet to Transmitter USRP. The transmitter USRP converts this information to an RF signal and broadcasts it on the air interface.

We kept the receiver thread's acquisition time long enough compared to the transmitter thread's time to create a packet and send it to the air interface. We used the synchronization feature of the Labview program. The Transmission and the Receiver threads are synchronized. With synchronization, it is intended that the Receiver can capture the signal sent by the Transmitter. After the acquisition process, the Receiver thread demodulates the signal with Labview's demodulation function. This function also takes synchronization bits as input. We can obtain the packet if the demodulation process succeeds. In the received data, first of all, CRC control is performed. The Receiver knows that the CRC result lies between the 17th and 32nd bits of the packet content. Then, the Receiver thread passes the first 16 bits of data through the CRC. Then, it compares this result with the CRC field in the packet. If the values are identical, then the CRC result of the packet is considered as successful. Otherwise, the packet is discarded. A packet that passes this stage successfully enters the Receiver ID control stage secondly. If the Receiver ID in the package content is different from the Receiver's own ID, the Receiver discards the packet again. Still, it increases the number of packets that it can receive its CRC correctly. This statistic is then used in the channel reliability calculation. The packet that passes these controls is saved into a variable. The result of the overall receive process is also saved to the memory as a

boolean variable. These variables are used in AoI and EAoI calculations. Receiver threads that have finished their operations are set to be idle for a while. The receiver module starts to listen to the air interface again before the new frame starts to ensure the acquisition process begins before the transmission process.

Results obtained in a frame are passed to another thread. The main task of this thread is to calculate the average AoI and EAoI based on the results of the experiments. In addition, the calculation of channel reliability, general statistics about the experiments are also calculated in this thread. After the experiment is completed, information about the overall test results is noted in a text file.

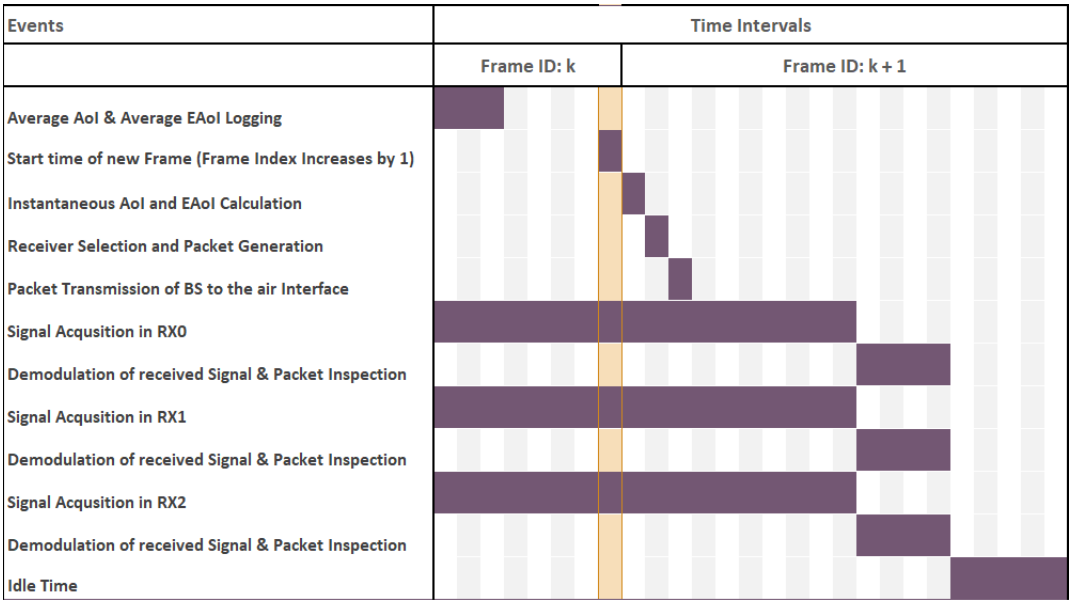

Figure 4.4: Runtime of the SDR Network

CHAPTER 5

CONCLUSION AND FUTURE WORK

Within the scope of the thesis, we examine Age of Information, Query Age of Information, and Effective Age of Information, which are included in the set of semantic metrics. First of all, we study the Age of Information metric. By implementing AoI-aware policies on SDRs, we examine how these policies perform in real environment conditions.

In the AoI domain, we studied the performance of different wireless link scheduling policies (Round Robin, Greedy, Whittle's Index Policy, and Max-Weight) in terms of the AoI. Emulation results reveal that the Whittle's Index and Max-Weight policies are superior to Round Robin and Greedy Policies, as they take the link reliabilities into account. Experiment results are close to the theoretical ones when the link reliabilities are higher. As the link reliabilities decreased, the Normalized performances of the Max-Weight Policy and Whittle's Index Policy gradually drifted away from the theoretical lower bound. We interpret that this performance drop is caused by the looseness of the lower bound at lower channel reliabilities and the varying noise profile of the real environment.

Next, we discuss the Effective AoI and Query AoI metrics used to examine pull-based networks. Using the methods defined for AoI in the literature, we derive a Lower bound for Effective AoI in the Proactive Serving scenario. We use the Max-Weight Policy in the literature for our studies on the SDR network by adapting it to the Effective AoI model. We obtain Throughput-adjustable Whittle's Index Policy by adding a small change to Whittle's Index Policy. We implemented and tested the EAoI-Aware Max Weight and EAoI-Aware Whittle's Index Policies on the SDR Network.

In the EAoI domain, experiments show that utilizing the statistical information about the query process increases the EAoI performance of Whittle's Index and Max Weight. However, the EAoI performance gap between the Query-aware policies and AoI-aware policies is dependent on the alignment of channel reliabilities and the query probabilities. We observed that Q-MW Policy often provides higher throughput than the Q-WIP Policy.

We study the Query-AoI metric, similar to the EAoI metric but does not include the non-queried frames in the average Age calculation. In contrast, the EAoI includes the non-queried frames by assuming their value is equal to zero. By adapting the Max-Weight Policy to the QAoI case, we obtain a scheduling policy that can operate in multi-user networks. For the adaptation of the Max-Weight to the QAoI domain, we propose a weight function. We tested the resulting QAoI-Aware Max Weight in a simulation environment. The simulation results reveal that utilizing the QAoI-Aware Max-Weight policy in the QAoI domain gives notable results. Although the assumptions that we use regarding the system model was very restrictive at this early work, our results show that the method we use can perform well.

In future studies, our primary aim is to examine different semantic metrics beyond AoI. To this end, we want to deepen our preliminary work on another semantic metric, QAoI. With the help of the QAoI perspective, we want to study pull-based scenarios in more detail. Moreover, in further studies, we aim to modify our SDR testbed to examine the QAoI metric in real wireless networks. As far as we know, the studies conducted so far on the QAoI concept do not cover multiuser scenarios and do not include realization studies. We think that the studies we plan to do for the QAoI metric would be innovative and literature-enhancing.

REFERENCES

[1] O. Ayan, H. Özkan, and W. Kellerer. An experimental framework for age of information and networked control via software-defined radios. In *IEEE International Conference on Communications (ICC)*, pages 1–6, June 2021.

[2] H. B. Beytur, S. Baghaee, and E. Uysal. Towards AoI-aware smart IoT systems. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 353–357, 2020.

[3] E. T. Ceran, D. Gündüz, and A. György. A reinforcement learning approach to age of information in multi-user networks. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1967–1971, 2018.

[4] F. Chiariotti, J. Holm, A. Kalør, B. Soret, S. Jensen, T. Pedersen, and P. Popovski. Query age of information: Freshness in pull-based communication, 05 2021.

[5] P. Cruz, N. B. Carvalho, and K. A. Remley. Designing and testing software-defined radios. *IEEE Microwave Magazine*, 11(4):83–94, 2010.

[6] B. A. Fette. Chapter 1 - history and background of cognitive radio technology. In B. A. Fette, editor, *Cognitive Radio Technology (Second Edition)*, pages 1–26. Academic Press, Oxford, second edition edition, 2009.

[7] Z. Han, J. Liang, Y. Gu, and H. Chen. Software-defined radio implementation of age-of-information-oriented random access. In *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pages 4374–4379, 2020.

[8] Q. He, D. Yuan, and A. Ephremides. Optimal link scheduling for age minimization in wireless systems. *IEEE Transactions on Information Theory*, 64(7):5381–5394, 2018.

[9] J. Holm, A. E. Kalør, F. Chiariotti, B. Soret, S. K. Jensen, T. B. Pedersen, and P. Popovski. Freshness on demand: Optimizing age of information for the query process. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021.

[10] I. Kadota and E. Modiano. Minimizing the age of information in wireless networks with stochastic arrivals. Mobihoc '19, page 221–230, New York, NY, USA, 2019. Association for Computing Machinery.

[11] I. Kadota, M. S. Rahman, and E. Modiano. WiFresh: Age-of-information from theory to implementation, 2020.

[12] I. Kadota, A. Sinha, and E. Modiano. Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints. *IEEE/ACM Transactions on Networking*, 27(4):1359–1372, 2019.

[13] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano. Scheduling policies for minimizing age of information in broadcast wireless networks. *IEEE/ACM Transactions on Networking*, 26(6):2637–2650, 2018.

[14] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano. Minimizing age of information in broadcast wireless networks. In *Invited paper, 54th Annual Allerton Conf.On on Communication, Control, and Computing, Monticello, Il, USA.*, Sept 2016.

[15] S. Kaul, M. Gruteser, V. Rai, and J. Kenney. Minimizing age of information in vehicular networks. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 350–358, Dec. 2011.

[16] F. Li, Y. Sang, Z. Liu, B. Li, H. Wu, and B. Ji. Waiting but not aging: Optimizing information freshness under the pull model. *IEEE/ACM Transactions on Networking*, 29(1):465–478, 2021.

[17] A. Maatouk, M. Assaad, and A. Ephremides. The age of incorrect information: an enabler of semantics-empowered communication, 12 2020.

[18] A. Mahajan and D. Teneketzis. *Multi-Armed Bandit Problems*, pages 121–151. 10 2007.

[19] J. Mitola. The software radio architecture. *IEEE Communications Magazine*, 33(5):26–38, 1995.

[20] NI. USRP-2920 Software Defined Radio Device Specifications. [online] Available: http://www.ni.com/manuals/.

[21] NI White Papers. Acquiring an Analog Signal: Bandwidth, Nyquist Sampling Theorem, and Aliasing. Technical report, National Instruments.

[22] NI White Papers. What Is NI USRP Hardware? Technical report, National Instruments.

[23] T. K. Oğuz, E. Uysal, and T. Girici. Implementation and evaluation of age-aware downlink scheduling policies. In *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–3, 2021.

[24] Packet-Based Digital Link. National Instruments Community. [online] Available: https://forums.ni.com/.

[25] E. Sert, C. Sönmez, S. Baghaee, and E. Uysal-Biyikoglu. Optimizing age of information on real-life TCP/IP connections through reinforcement learning. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, May 2018.

[26] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff. Update or wait: How to keep your data fresh. *IEEE Transactions on Information Theory*, 63(11):7492–7508, Nov 2017.

[27] C. Sönmez, S. Baghaee, A. Ergişi, and E. Uysal-Biyikoglu. Age-of-Information in practice: Status age measured over TCP/IP connections through WiFi, Ethernet and LTE. In *2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–5, June 2018.

[28] R. Talak, S. Karaman, and E. Modiano. Improving age of information in wireless networks with perfect channel state information. *IEEE/ACM Transactions on Networking*, 28(4):1765–1778, 2020.

[29] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, T. Soleymani, B. Soret, and K. Johansson. Semantic communications in networked systems, 03 2021.

[30] J. Vermorel and M. Mohri. Multi-armed bandit algorithms and empirical evaluation. In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, and L. Torgo, editors, *Machine Learning: ECML 2005*, pages 437–448, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[31] P. Whittle. Restless bandits: activity allocation in a changing world. *Journal of Applied Probability*, 25(A):287–298, 1988.

[32] R. D. Yates. Lazy is timely: Status updates by an energy harvesting source. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 3008–3012, June 2015.

[33] B. Yin, S. Zhang, Y. Cheng, L. X. Cai, Z. Jiang, S. Zhou, and Z. Niu. Only those requested count: Proactive scheduling policies for minimizing effective age-of-information. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 109–117, 2019.

[34] L. Zhou. A survey on contextual multi-armed bandits. *ArXiv*, abs/1508.03326, 2015.