

Minimizing Age of Information for Semi-Periodic Arrivals of Multiple Packets

by

Mianlong Chen

B.Sc., China University of Geoscience (Wuhan), 2017

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science
University of Victoria
BC, Canada

© Mianlong Chen, 2019
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by
photocopying or other means, without the permission of the author.

Minimizing Age of Information for Semi-Periodic Arrivals of Multiple Packets

by

Mianlong Chen

B.Sc., China University of Geoscience (Wuhan), 2017

Supervisory Committee

Dr. Kui Wu, Supervisor
(Department of Computer Science)

Dr. Nishant Mehta, Departmental Member
(Department of Computer Science)

ABSTRACT

Age of information (AoI) captures the freshness of information and has been used broadly for scheduling data transmission in the Internet of Things (IoT). We consider a general scenario where a meaningful piece of information consists of multiple packets and the information would not be considered complete until all related packets have been correctly received. This general scenario, seemingly a trivial extension of exiting work where information update is in terms of single packet, is actually challenging in both scheduling algorithm design and theoretical analysis, because we need to track the history of received packets before a complete piece of information can be updated. We first analyse the necessary condition for optimal scheduling based on which we present an optimal scheduling method. The optimal solution, however, has high time complexity. To address the problem, we investigate the problem in the framework of restless multi-armed bandit (RMAB) and propose an index-based scheduling policy by applying Whittle index. We also propose a new transmission strategy based on erasure codes to improve the performance of scheduling policies in lossy networks. Performance evaluation results demonstrate that our solution outperforms other baseline policies such as greedy policy and naïve Whittle index policy in both lossless and lossy networks.

Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
Acknowledgements	ix
1 Introduction	1
1.1 Introduction of Age of Information	1
1.2 Motivations and Contributions	2
1.3 Thesis Organization	4
2 Related Work and Problem Formulation	6
2.1 Definition	6
2.1.1 Age of Information	6
2.1.2 Time Average AoI	8
2.2 Related Work	10
2.2.1 AoI Optimization based on Queuing Theory	10
2.2.2 AoI as a Tool	11
2.3 System Model and Problem Formulation	12
2.3.1 System Model and Assumption	12
2.3.2 Problem Formulation	17
3 A Necessary Condition for Optimal Scheduling and an Asymptotically Optimal Solution	19

3.1	Simple Example	19
3.2	Proof	23
4	Index-based Scheduling Policy Design and Analysis	28
4.1	Introduction of Multi-armed Bandit Problem	28
4.2	Approximating <i>AoI</i>	29
4.3	Algorithm Analysis and Design	30
4.3.1	Restless Multi-armed Bandit Problem (RMAB)	31
4.3.2	Decoupled Model and the Properties of Optimal Policy	31
4.3.3	Derivation of the Whittle Index	35
4.4	Scheduling Algorithm with the Whittle Index	37
4.5	Why Does Our Solution Work Well	39
5	Erasur Code-Aided AoI Minimization in Lossy Networks	40
5.1	Regular Transmission	41
5.2	Introduction of Erasure Code and Transmission Scheme	41
5.2.1	Erasure Code	41
5.2.2	Erasure Code-based Transmission	42
5.3	Analysis and Estimation	43
6	Simulation and Performance Evaluation	47
6.1	Comparison with the Optimal Solution	48
6.2	Impact of Number of Sources	49
6.3	Benefit of Coding in Lossy Networks	52
7	Conclusions and Future Work	55
7.1	Summary	55
7.2	Future Work	56
	Bibliography	57

List of Tables

Table 2.1 List of Main Notation	15
Table 6.1 The Hardware Environment	48
Table 6.2 Impact of Number of Sources on $E\text{AvgAoI}$	50

List of Figures

Figure 2.1	The Communication Framework: The base station (BS) makes transmission schedules to update N users d_1, d_2, \dots, d_N on information of sources s_1, s_2, \dots, s_N , respectively.	7
Figure 2.2	An Example of AoI Evolution of a Communication Link	8
Figure 2.3	An example of AoI update: Assume each time interval contains $T = 6$ timeslots. Information I_i^1 (of length 4 packets) and information I_i^2 (of length 6 packets) arrive in burst at the BS at the beginning of interval 2 and 4, respectively. The BS finishes the transmission of information I_i^1 at timeslot 10 and hence the destination updates A_i to $4 = (14 - 10)$; the BS finishes the transmission of information I_i^2 at timeslot 24 and hence the destination updates A_i to $6 = (24 - 18)$	16
Figure 2.4	The area under A_i for arbitrary timeslot $t_{k,j}$ within interval k	18
Figure 3.1	Consecutive Transmission: Each source transmits all packets and update the age, then the next one. Update order: $S_3 \rightarrow S_2 \rightarrow S_1$	20
Figure 3.2	Transmission with interruption: Let S_3 transmits 2 packets first, and then interrupted by S_2 . S_2 transmits all 2 packets, followed by S_1 . Finally, S_3 starts transmitting the last packet. Update order: $S_3(\text{interrupted}) \rightarrow S_2 \rightarrow S_1 \rightarrow S_3$	21
Figure 3.3	Illustration of AoI changes of two sources.	25
Figure 3.4	Adjustment of transmission schedule (s_1^1, s_1^2, \dots are packets in the information of s_1 , and s_2^1, s_2^2, \dots are packets in the information of s_2).	25
Figure 3.5	Scheduling with earliest deadline first (EDF).	26
Figure 3.6	Scheduling with Lemma 1 where BS transmits information of S_2 first.	26
Figure 4.1	The Approximation Age h_k under the policy π^* forms a DTMC.	33

Figure 4.2 The Approximation Age h_k under the threshold-type policy forms a DTMC.	35
Figure 5.1 A graphical illustration of the encoding and decoding process of erasure code.	42
Figure 5.2 Illustration of effect of BER on PDP.	43
Figure 5.3 Illustration of effect of BER on ETT, (8, 6) code is used for <i>EC Trans.</i>	45
Figure 6.1 Benchmark with the <i>Opt</i> algorithm.	49
Figure 6.2 Box-and-Whisker plot depicting the distribution of numerical results. Each box contains five important values of the dataset, the minimum, lower quartile, median (second quartile), upper quartile and maximum of <i>E AvgAoI</i> , from bottom to top. The white points above the maximum line are outliers.	51
Figure 6.3 Analysis of T size: $K = 3000, p_i = 1.0$	53
Figure 6.4 Analysis of information generation rate: $K = 3000, p_i \in \{0.7, 0.8, 0.9, 1.0\}$	54

ACKNOWLEDGEMENTS

I would like to thank my supervisor **Dr. Kui Wu**, who offered me a great opportunity to study at University of Victoria, a really beautiful university! **Dr. Kui Wu** steered me in the right direction during my graduate studies and research. I would like to thank him for his great help and advice on writing this thesis. I could not have finished this thesis without his patient instruction and expert guidance.

I want to thank **my parents**, for supporting me to study abroad. They give me the life and try their best to let me well educated.

I would also like to thank:

Dr. Nishant Mehta and Dr. Aaron Gulliver, for their willingness to be on the committee for my oral exam.

Dr. Fengyong Li, for his discussion and inspiration that helped me complete my thesis.

My landlord, Camillus, for her good care when I was in low moments which made me feel like at home.

My friends and office mates, for their kindness and encouragement during my life in Canada.

Thank you all!
Mianlong Chen

Chapter 1

Introduction

1.1 Introduction of Age of Information

The concept of *Age of Information*, simply age or AoI, was first proposed in [21] for qualifying the data freshness that represents the status regarding a remote system or node. More specifically, AoI depicts the time elapsed since the generation of status update that was most recently received at the destination. Different from traditional network performance metrics, such as delay and latency, AoI characterizes not only the queuing delay but also the inter-delivery time, due to the fact that AoI increases until the status observed at the destination is updated. Hence, AoI captures the data freshness property from the perspective of destination and is more precise to measure the timeliness of messages for systems that require timely information.

Keeping the status information fresh at the destination is of great importance for a wide range of applications. For example, a sensor of autonomous vehicle that measures the proximity to obstacles or other vehicles in the vicinity, usually samples new location information and communicates with the core monitor at a pretty high frequency. Information of relative location carried in a lagged update may be obsolete and result in collisions [11] [30]. Another example is the application of sensors that monitor real-time health status in medical service field [2]. For remote surgery, real-time data recording the heart beating rate, breathing rate, blood pressure, etc., needs to be updated very fast and frequently, since the data represents the status of the organs of the patients. What's more, during surgery, each operation of the surgery is based on a complete understanding of the real-time health status of the patients. Any outdated information that represents the organs status can lead to

wrong surgery operation or break the continuity of the operations, since the surgeon has a high requirement of data freshness regarding the organs. Both outdated and delayed information may raise the risk during surgery. In conclusion, the requirement of freshness of health related data may be extremely high. AoI can be used as a metric to evaluate the performance of these applications and systems in medical service field.

As a common metric, AoI can be used to measure the performance of a wide range of applications and systems. However, factors like channel quality and bandwidth will also affect the resource allocation and result in information transmission failure. Consequently, the status knowledge at the destination may not be updated as expected and hence the age becomes higher. In order to provide fresh information and minimize the age, algorithms and methods need to be developed and applied to optimize the allocation of limited resources, which is known as the AoI Minimization problem (AoIMP).

1.2 Motivations and Contributions

The AoI Minimization problem has been widely studied in the past [4, 13, 15, 16, 17, 18, 19, 35, 36]. The majority of existing AoI research adopts the following model: the time is slotted; only one source can be served at each timeslot, and one packet can be transmitted, if successful, in one timeslot. More importantly, they assume that each packet contains a status update and the AoI is updated upon the successful reception of a new packet. The hidden one-packet-one-information assumption, however, should not be the norm, since the one-packet-based AoI model cannot be applied in many real-world applications where a meaningful piece of information needs to be encoded in multiple packets.

Autonomous driving and smart manufacturing are two motivating examples that a status update should be performed on the basis of multiple packets. In autonomous driving, short videos captured by front cameras of a vehicle are critical for decision making. Useful information embedded in these videos can only be explored by performing intensive computation and analysis on the raw data. Therefore, useful information can only be derived and processed after the in-vehicle processor or road-side unit (RSU) receives a short video encoded in multiple packets [1]. In smart manufacturing, monitoring units that assemble multiple sensors and communication components are used to monitor the status of running machines, including information

such as temperature, speed, depth, and vibration [26]. An effective control decision can be made only after the collection of all needed information. In this context, the absence of any aspect will fail the update of control action. Thus, multiple packets need to transmit to form a valid update in terms of AoI. We are thus motivated to extend the one-packet-one-information model to a more general multiple-packet-one-information model.

Such an extension, while conceptually simple, poses technical challenges in both data transmission scheduling and theoretical analysis. From the viewpoint of transmission scheduling, a simple policy that achieves the minimum average AoI in the presence of random information size¹ is still unknown. In terms of theoretical performance analysis, it is extremely difficult, if not impossible, to follow the traditional queueing theory-based analysis [6, 7, 16, 23], because we need to track a random number of packets to determine the time for AoI update. To be more specific, the number of system status is non-deterministic would we build a Markov chain to track the history of successfully received packets before AoI update.

Motivated by this, we study the AoI Minimization problem with constraint that a complete status update information is consisted of multiple packets and the useful status update can not be explored only if all packets that belong to the information are successfully received at the destination. Hence, the *AoI* is updated at the information level, rather than the packet level. In general, the contributions of this thesis can be summarized as follows,

1. We study the problem of minimizing the average AoI in a multi-source system where a complete information consists of multiple packets and the information size is random. Tracking the AoI in such a system is difficult, since we need to record the time when a new information starts transmission and the time when all the packets have been received. To address this, we first derive a necessary condition for an optimal solution and design an algorithm that is asymptotically optimal.
2. To reduce the complexity of the optimal solution, we develop an index-based scheduling method. We avoid traditional queueing theory and cast our problem as a restless multi-armed bandit problem. For this, we propose a new way to track the AoI indirectly, by *approximating* the time elapsed since last AoI update. Such an approximation not only simplifies the analysis, but also allows

¹Information size refers to the number of packets that are needed to transmit the information.

us to design a Whittle index-based scheduling method that achieves a near-optimal solution.

3. For lossy networks, we further propose a packet transmission scheduling strategy that takes advantage of erasure codes for AoI update. Compared with the regular transmission (using retransmission when fails) strategy, erasure code transmission provides a higher successful packet transmission probability. As a result, the system performance measured in terms of AoI is improved.
4. Using simulation, we systematically evaluate the proposed AoI update strategy and compare it with two other baseline strategies, greedy policy and naïve Whittle index policy, in both lossless and lossy networks.

1.3 Thesis Organization

The rest of this thesis is organized as follows,

Chapter 2: The related work is introduced and the research problem is formulated.

The formal definition of AoI is given. This chapter also defines the time average age, a metric that can be used to estimate the performance of a system.

Chapter 3: We prove that consecutive transmission can achieve a lower average system age than transmission with interruption. This gives the necessary condition for optimal AoI scheduling in our application context, based on which an optimal scheduling algorithm is designed.

Chapter 4: We first introduce the basic definition of multi-armed bandit problem and its applications. To track and analyze the AoI in our system model, we approximate AoI using time since last update, measured in terms of time interval. Based on this approximation, we transform the AoI minimization problem into a restless multi-armed bandit problem. The problem is decoupled and analyzed in the form of a single bandit problem. By applying Whittle’s method, we derive the Whittle Index with specific representation. We finally propose a simpler scheduling policy.

Chapter 5: We assume unreliable channel transmissions and propose a new transmission strategy with erasure codes. With detailed analysis, we show that trans-

mission with erasure code provides higher single packet successful transmission probability and lower expected transmission time.

Chapter 6: With extensive simulation, we evaluate three different scheduling policies, greedy policy (GP), naïve Whittle index policy (NWIP), and Multi-packet Whittle Index Policy (MWIP). Numerical results show that the system average age increases as the number of sources increases and decreases as the information generation rate increases. In addition, the MWIP policy outperforms the other two methods.

Chapter 7: We conclude this thesis and propose future work.

Chapter 2

Related Work and Problem Formulation

In this chapter, we briefly summarize the research on *age of Information* in recent years and formally formulate our research problem. We first define two basic relevant notions, *age of information* and *Time Average AoI*. Followed by is the summary of related work, most of which has been referred in this thesis. Our goal is to design a scheduling policy to minimize the AoI for applications where a piece of information consists of multiple packets. Hence, we introduce the system model and explain the specific meaning of *age of information* in our context. We then formulate the research problem formally.

2.1 Definition

2.1.1 Age of Information

Consider a system with two sets of nodes, namely, the source set and the destination set. Each source is mapped with a corresponding destination. For simplicity, let S_i represent source i , and d_i denote the corresponding destination. Then, a pair of source and destination can be referred as a communication link (S_i, d_i) .

In Figure 2.1, the communication framework is illustrated with the source and destination sets. A stochastic process is observed at the sources S_i . The destination d_i has an interest in this stochastic process and needs the knowledge regarding the status at the source. Hence, the status information sampled at the sources needs to be transmitted to the destinations via a base station (BS), in the form of data

packets. Considering factors like uncertainty of channel and randomness of status information sampling, the base station needs to manage the resources and schedule the transmission opportunities to keep the destinations updated. Usually, there is a buffer at the BS storing packets waiting for transmission.

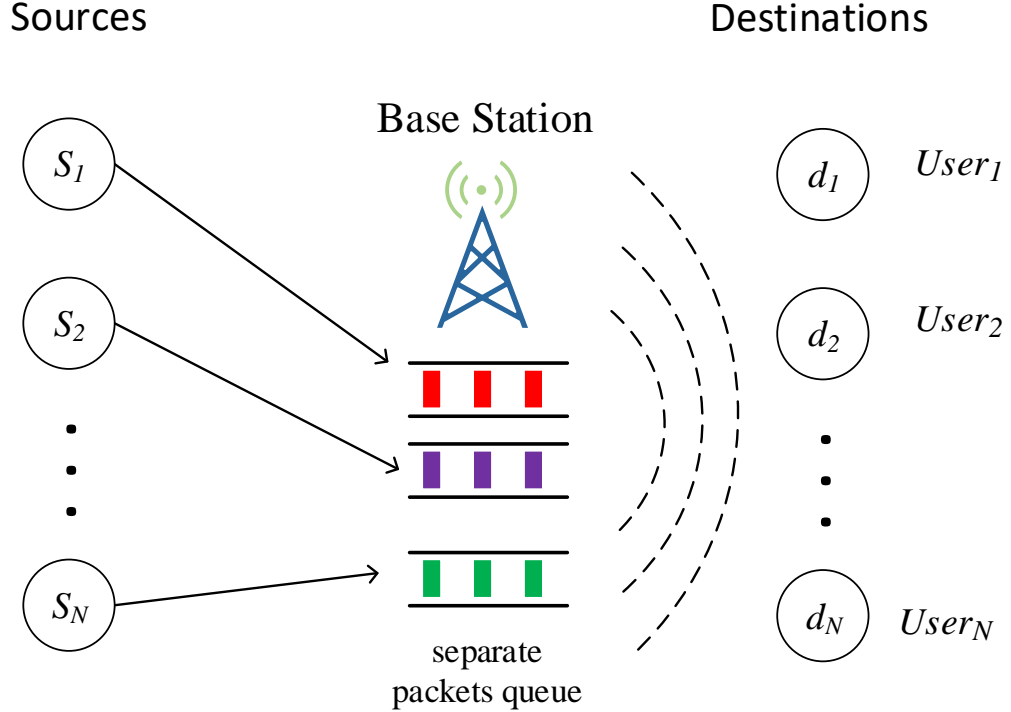


Figure 2.1: The Communication Framework: The base station (BS) makes transmission schedules to update N users d_1, d_2, \dots, d_N on information of sources s_1, s_2, \dots, s_N , respectively.

Assume that the first update is generated at time t_1 , followed by updates generated at t_2, t_3, \dots, t_k , and, let t_k' indicates the arrival time of the k -th update at the destination. Then, the AoI of an update is given by,

$$\Delta(k) = t_k' - t_k \quad (2.1)$$

Also, assume that the k -th update is the most recent one that has been delivered to destination d_i . Then at arbitrary time t ($\geq t_k'$), the AoI observed at destination d_i

is

$$AoI_i(t) = t - t_k \quad (2.2)$$

In the absence of a new update, the value of $AoI_i(t)$ increases linearly with time t , which means the knowledge regarding the source status gets older. When a new update arrives at destination d_i , the age is reset to a smaller value.

The evolution of AoI for a communication link is illustrated in Figure 2.2.

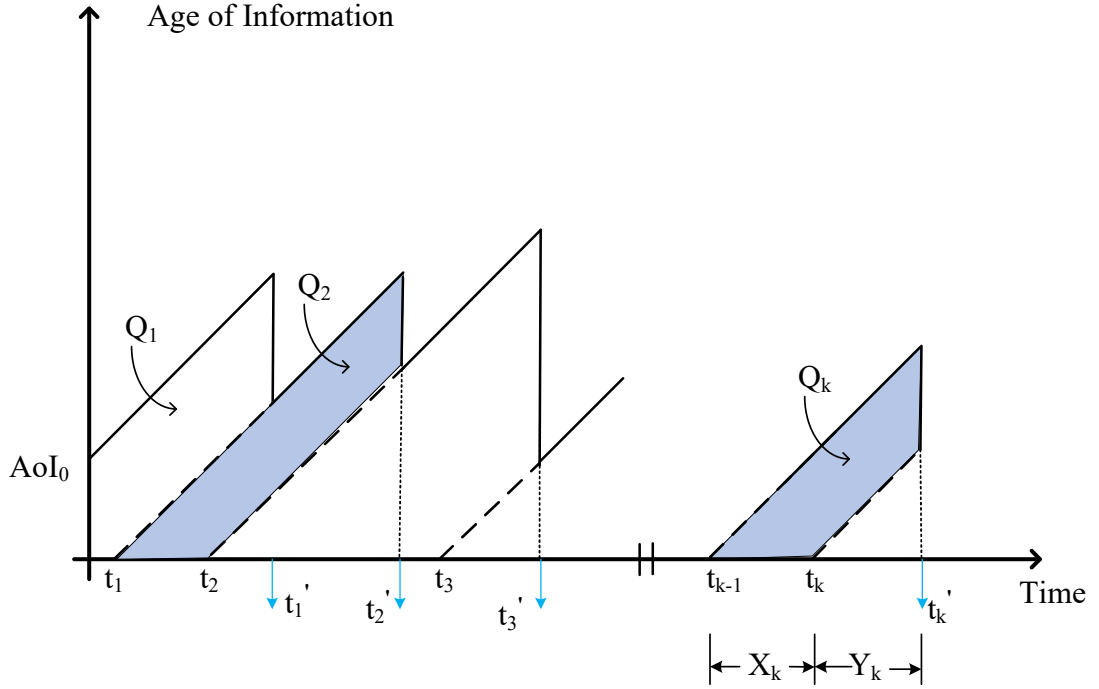


Figure 2.2: An Example of AoI Evolution of a Communication Link

2.1.2 Time Average AoI

In accordance with the representation and evolution of AoI, it's easy to catch the accurate AoI value at any time t . In many cases, to estimate the performance of a system or application, the *time average age* is needed. The *time average age* of the received status updates is the area under the sawtooth in Figure 2.2, normalized by the time interval of observation. Over an arbitrary time interval (t_{start}, t_{end}) , the

average AoI is defined as follows:

$$\overline{AoI} = \frac{1}{t_{end} - t_{start}} \int_{t_{start}}^{t_{end}} AoI(t) dt \quad (2.3)$$

For simplicity, we set $t_{start} = 0$ and $t_{end} = T = t'_k$, then expression (2.3) can be written as:

$$\overline{AoI} = \frac{1}{T} \int_0^T AoI(t) dt, \quad (2.4)$$

where the selected time interval of observation is $(0, T)$. We decompose the area under the sawtooth into a sum of disjoint parts. Due to the uncertainty of the initial age AoI_0 and the generation time t_1 , the size of the first polygon area Q_1 is not fixed. Following this part are the trapezoids Q_k for $k \geq 2$ (Q_2 and Q_k are highlighted in the figure), and the final triangular area with width Y_k over the sub-time interval (t_k, t'_k) . By concatenating these parts, we have the time average age over $(0, T)$:

$$\overline{AoI} = \frac{Q_1 + \sum_{k=2}^{N(T)} Q_k + Y_k^2/2}{T} \quad (2.5)$$

where $N(T) = \max\{k | t_k \leq T\}$ denotes the number of status updates received by time T .

Moreover, from Figure 2.2, we can regard each trapezoid Q_i as the difference of a bigger isosceles triangle and a smaller triangle. Defining

$$X_k = t_k - t_{k-1} \quad k \geq 2 \quad (2.6)$$

to be the inter-delivery time of two consecutive status updates, it follows that

$$\begin{aligned} Q_k &= \frac{(X_k + Y_k)^2}{2} - \frac{Y_k^2}{2} \\ &= X_k \cdot Y_k + X_k^2/2 \end{aligned} \quad (2.7)$$

Replacing Q_k in (2.5) by (2.7), gives

$$\overline{AoI} = \frac{Q}{T} + \frac{1}{T} \sum_{k=2}^{N(T)} \left[X_k \cdot Y_k + \frac{X_k^2}{2} \right] \quad (2.8)$$

where $Q = Q_1 + Y_k^2/2$. It can be observed that the age contribution Q represents a boundary effect that is finite, thus the value of Q/T will vanish as T grows. When the value of T is large enough, the first part of (2.8) can be ignored.

2.2 Related Work

The importance of providing timely information has been recognized and needed in different domains, including, for example, environment protection, health monitoring, and intelligent traffic. We summarize the related research work in this section.

Queuing theory is the most used theory to analyze the scheduling process. In multiple source systems, first-come-first-serve (FCFS) policy may not be applicable when sources need to be scheduled with different priorities. Moreover, the sources with high information sampling rate may be over-served, consuming too many transmission opportunities. Consequently, the AoI related to other sources may become high. Hence, the scheduling decision needs to be optimized as to minimize the system AoI. In addition, AoI has been utilized as a control tool in some areas.

2.2.1 AoI Optimization based on Queuing Theory

Queuing theory has been widely recognized as a core methodological framework for analyzing traditional network performance metric, such as delay and throughput. Inspired by the work studying traditional network metrics, the communication model was considered as a simple queuing system in the early AoI work [20] [21] [22] [23].

In [21], the authors focused on minimizing the age of status update sent by vehicles over a carrier-sense multiple access (CSMA) network. The minimum age may be approached with gradient descent. Unfortunately, it is unknown if this method works for general cases. In [22], the authors show that a smaller age can be achieved by allowing nodes to piggyback other node's status updates.

In order to meet the user demands, the authors of [20] try to minimize the freshness of data in warehouses. At the staging age where status updates wait before they are committed to the database, the queue length and delay are estimated for system performance optimization. Nevertheless, the authors didn't consider optimal update rates in [20].

In [23], the authors focused on providing timely information under the constraint of limited network resources. They derived general age minimization methods and

applied them to a queueing-based abstract system consisting of a source, a service facility, and monitors. In particular, three simple models M/D/1, M/M/1 and D/M/1 were studied under the First-Come-First-Serve (FCFS) policy.

2.2.2 AoI as a Tool

Although the notion of AoI was only proposed in recent years, there has already been some work that uses it as a control tool.

In cellular networks, each base station (BS) needs to estimate the channel responses from the user equipment (UE) that are active in the current block. These responses are utilized to process the uplink and downlink signals by the BS. The knowledge of the current channel response is the channel state information (CSI). In non-reciprocal wireless links, the knowledge of current channel state observed at the transmitter is explored from the CSI feedback sent from the receiver. In this case, the available information at the destination has aged over time, affecting the efficiency of the communication. Channel information aging is caused by multiple factors, including but not limited to measuring times, transmission delay, processing time of decoding and so on. Aiming at studying the effect of outdated CSI on the performance of feedback links and protocols, in [9] utility functions are used as a general performance metric that accounts for various scenarios and the cost of feedback. Also, in [8] the channel is modeled as a Finite State Markov Channel (FSMC) with two states representing the fading conditions, good and bad. Both works yield tractable analytic results, which are useful for designing efficient adaptation functions and feedback protocols.

Consider a system with energy harvesting sources. The time-varying availability of energy and battery capability constraints can limit the sampling rate at the sources. Thus, it is interesting to investigate how a stochastic energy harvesting system affects the AoI at the destination. The minimization regarding AoI under such conditions is studied in [3]. An offline solution that minimizes the time average AoI for an arbitrary energy replenishment profile is derived, using discrete time dynamic programming formulation. An effective online heuristic that achieves performance close to an offline policy is also proposed. Simulation results indicate that significant improvement over a greedy approach can be obtained. Energy harvesting constraints are also considered in [35], while the randomness is on the service times, not on the energy arrival process. The authors show that the optimal policy is lazy, i.e., following a service completion,

the service facility is frequently left idle even though the server may possibly have sufficient energy to submit a new update.

2.3 System Model and Problem Formulation

We consider the case that complete information consists of multiple packets and the AoI can only be updated upon the arrival of all packets without error. Hence, the original definition of AoI for single packet update cannot be applied in our research problem.

In this section, we redefine the *Age of Information* in the above context based on the concept introduced in Sec. 2.1.1. Then, we formally formulate our research problem and give the objective function in accordance with Sec. 2.1.2.

2.3.1 System Model and Assumption

In this thesis, we consider a typical IoT network, where a set of sensing devices (e.g., surveillance video cameras) send information to a 5G base station (BS). The BS forwards the information to the corresponding destinations (e.g., network users), which process the information and update the AoI accordingly. The system architecture is shown in Figure 2.1. Moreover, we make the following assumptions:

- A complete piece of information (e.g., a frame of image) I_i is measured in terms of r_i ($r_i \geq 1$) packets, where r_i is a random variable following some distribution. We use positively truncated normal distribution¹, i.e., $r_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, as an example in our performance evaluation, but note that the algorithms developed in this thesis can be applied for any distribution.
- Time is slotted, and one packet can be successfully transmitted within one timeslot with probability p . To ease explanation, we first assume that $p = 1$ and then relax this constraint for lossy networks in Chapter 5. In addition, we assume a shared wireless channel between the BS and the destinations such that only one destination can communicate with the BS in one timeslot.
- We assume an information interval (\mathcal{I}) at the sources. At the beginning of each \mathcal{I} , a source s_i generates a piece of information with probability p_i or remains

¹Since values of r_i must be positive integers, we take samples of positively truncated normal distribution and then round them up to be integer.

idle with probability $1 - p_i$. The *bursty* packets in the information arrive at the BS, which maintains a separate queue for each source, as shown in Fig. 2.1. This assumption is applicable to many practical IoT systems where the sensors normally send information at some interval times. We assume that an \mathcal{I} includes multiple timeslots to align with the assumption that a piece of information consists of multiple packets. This semi-periodic information generation model is a good approximation of many real-world applications (e.g., surveillance video cameras or edge-aided industrial systems), where traffic may be neither strictly periodic nor purely random (Poisson).

- We assume an *information-buffer-free* network, that is, if a source generates new information but the BS has not finished transmitting its previous information, the BS stops the transmission of current information and starts to transmit the new information. This assumption is needed to make our later analysis valid. Considering information-buffer is more challenging and is left for future research.

We explain the concept of *status update* in the context of multiple packet-based information. At the BS, there is a buffer for each source which stores the samples in the form of packets, each containing the timestamp when the corresponding sample was extracted. Each information consists of a (random) number of packets. The BS sends the information to intended destinations; when a destination receives all the packets in the information, the destination is said to have a status update regarding the corresponding source node.

The freshness of the knowledge the destination has about the status of the source node is captured by the concept of the AoI. Like most previous work [14, 24], *this thesis only focuses on the transmission scheduling between the BS and the destinations (i.e., the last hop of information delivery)*, since AoI of a source is viewed from the point of the corresponding destination. Under this context, when people say “source s_i generates information” or “information I_i is generated”, it means the BS has the information I_i from s_i in the buffer.

Following the above assumptions, we can formally define AoI in our context .

Definition 1. (AoI:) *The AoI of source i at time t , is defined as*

$$A_i(t) = t - \mu_i \tag{2.9}$$

where μ_i denotes the timestamp in the first packet of the information for the most recent status update regarding source i . By default, $\mu_i = 0$ if the destination does not have any status update yet. Since the basic scheduling unit is in terms of timeslot, we use the post-action age when calculating AoI value, i.e., A_i is checked only at the end of a timeslot.

It is important to notice that the AoI definition above is different from that in existing work [23, 18, 17]. In our problem, we do not assume an explicit status update packet from the source. Instead, the status update happens only after all packets belonging to the information have been received.

For reference, the main notation in this thesis are listed in Table 2.1.

Remark 1. For schedulability purpose, we need the following constraint

$$T \geq \sum_{i=1}^N p_i E[r_i], \quad (2.10)$$

where N is the total number of sources. The above constraint implies that the average information generation rate (measured in terms of average packets/per \mathcal{I}) in the system must not be higher than the system throughput. Otherwise the system would not be stable in the sense that the AoI of some sources will go to infinity in the long run. For instance, a source with a very fast information rate may never get its information updated at the destination.

Table 2.1: List of Main Notation

Symbol	Meaning
s_i	the index of source nodes, $i \in \{1, \dots, N\}$;
I_i	a piece of information from source s_i ;
r_i	the number of packets included in information I_i ;
\mathcal{I}	information interval;
$A_i(t)$	age of information of s_i at the end of timeslot t ;
k	index of \mathcal{I} , $k \in \{1, \dots, K\}$;
K	the total number of \mathcal{I} 's;
T	the number of slots in each \mathcal{I} ;
$h_{i,k}$	the number of \mathcal{I} 's since last information I_i has been received;
$t_{k,j}$	the j -th timeslot within k -th \mathcal{I} , $j \in \{1, \dots, T\}$;
p_i	the probability that s_i generates information in each \mathcal{I} ;
$\Lambda_i(k)$	indicator variable whether I_i is generated at the beginning of the k -th \mathcal{I} ;
$u_i(t_{k,j})$	indicator variable whether s_i is selected for update at timeslot $t_{k,j}$;

With the notation introduced, we now use an example to show how AoI should be updated at the destination. Clearly, BS needs at least r_i timeslots to complete the transmission of I_i . A_i will increase until all r_i packets are successfully transmitted to the destination. An example that illustrates the evolution of A_i is shown in Fig 2.3.

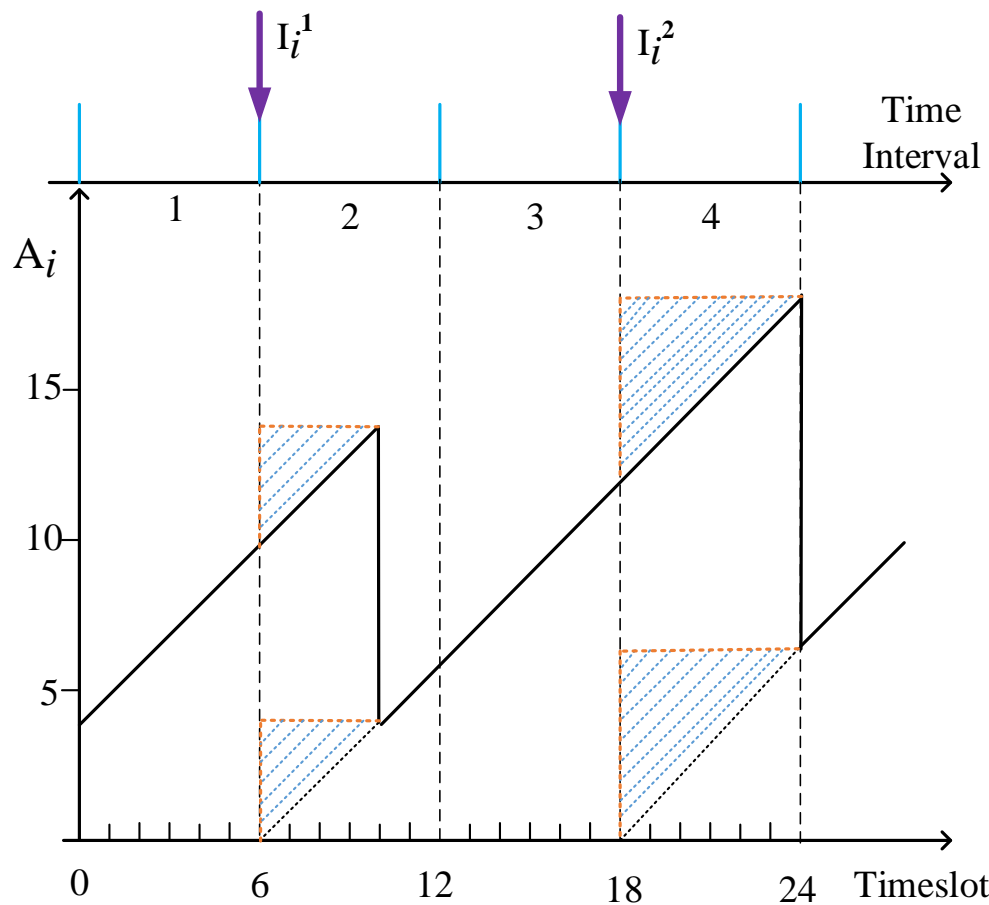


Figure 2.3: An example of AoI update: Assume each time interval contains $T = 6$ timeslots. Information I_i^1 (of length 4 packets) and information I_i^2 (of length 6 packets) arrive in burst at the BS at the beginning of interval 2 and 4, respectively. The BS finishes the transmission of information I_i^1 at timeslot 10 and hence the destination updates A_i to $4 = (14 - 10)$; the BS finishes the transmission of information I_i^2 at timeslot 24 and hence the destination updates A_i to $6 = (24 - 18)$.

2.3.2 Problem Formulation

As seen in Fig 2.4, the area under the AoI line is the *cumulative* AoI calculated within an arbitrary timeslot. As mentioned above, we use the *post-action* age to represent the A_i at the end of each timeslot and the A_i is calculated in terms of timeslot. Thus we can calculate the long-term *expected average AoI* for source s_i under scheduling policy π by

$$\begin{aligned} E[A_i^\pi] &= \frac{1}{KT} E \left[\sum_{k=1}^K \sum_{j=1}^T (A_i(t_{k,j}) - \frac{1}{2}) \bar{A}_i(0) \right] \\ &= \frac{1}{KT} E \left[\sum_{k=1}^K \sum_{j=1}^T A_i(t_{k,j}) \bar{A}_i(0) \right] - \frac{1}{2} \end{aligned} \quad (2.11)$$

where the expectation is with respect to the randomness in information generation and the scheduling policy, and $\bar{A}_i(0)$ denotes the initial AoI value of source i . Without loss of generality², we set $\bar{A}_i(0) = 1$ for $\forall i$ and omit $\bar{A}_i(0)$ henceforth. Note that the value of $\frac{1}{2}$ is the size of the triangle shown in Fig. 2.4, which should be deducted since we use post-action AoI. Therefore, for the system we can define the long-term *expected average AoI* ($E\text{AvgAoI}$) as

$$\begin{aligned} E[A^\pi] &= \frac{1}{N} \sum_{i=1}^N E[A_i^\pi] \\ &= \lim_{K \rightarrow \infty} \frac{1}{KNT} E \left[\sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^T A_i(t_{k,j}) \bar{A}_i(0) \right] - \frac{1}{2}. \end{aligned} \quad (2.12)$$

Note that we focus on the average age, but our work can be easily extended to the case with a general weight parameter that calculates the weighted average of ages. A scheduling policy is defined as a scheduling vector $\mathbf{U} \triangleq \{u_1(t_{k,j}), \dots, u_N(t_{k,j})\}$, where $u_i(t_{k,j}) \in \{0, 1\}$ indicates whether or not the BS decides to transmit s_i 's information at the beginning of timeslot $t_{k,j}$.

Our goal is to design an optimal scheduling policy π^* *at the BS* that minimizes $E\text{AvgAoI}$ defined by (2.12).

²This is because the given initial value will not impact the long-term scheduling decision.

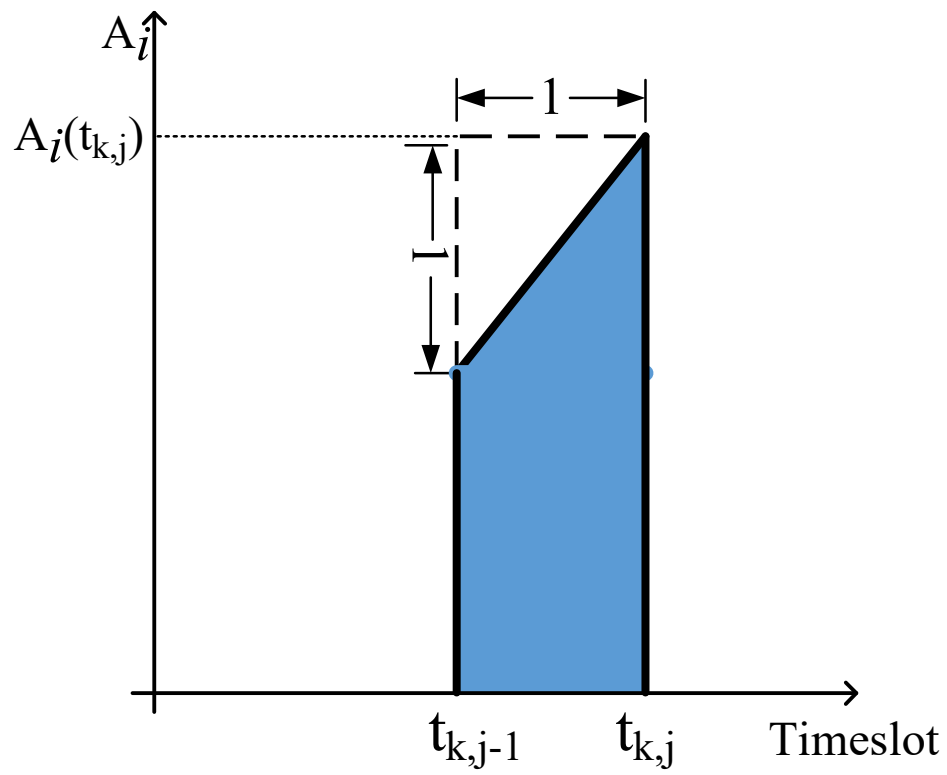


Figure 2.4: The area under A_i for arbitrary timeslot $t_{k,j}$ within interval k .

Chapter 3

A Necessary Condition for Optimal Scheduling and an Asymptotically Optimal Solution

Due to the fact that complete information consists of multiple packets and the AoI at the destination will not be updated until all packets have been successfully received without error, there are mainly two ways, *consecutive transmission* and *transmission with interruption*, to transmit the packets containing status update information under the transmission constraint (one packet each timeslot). The former is to transmit all packets of a complete piece of information whenever the source is scheduled for transmission. The latter allows another source to interrupt the transmission of the current source. We will prove that *consecutive transmission* achieves lower average age than that of *transmission with interruption*.

3.1 Simple Example

We consider a simple network with 3 sources, denoted as S_1 , S_2 , and S_3 . Also, assume the information size of each source is $r_1 = 1$, $r_2 = 2$ and $r_3 = 3$, respectively. Assume that the initial age of these 3 sources are 1, 2 and 3, respectively. We can simulate the AoI evolution for this network. In order to compare the performance of *consecutive transmission* and that of *transmission with interruption*, we need to calculate the time average age within the interval. For simplicity, we ignore the normalization of time, just calculate the total age, which is the area under the AoI evolution line.

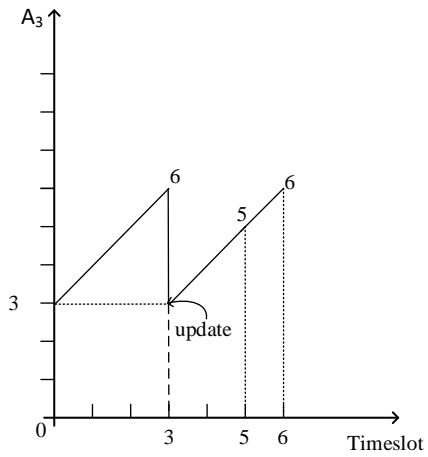
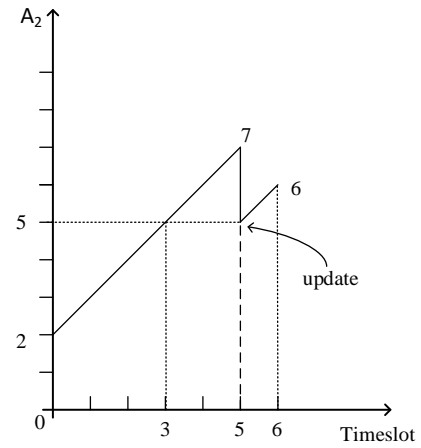
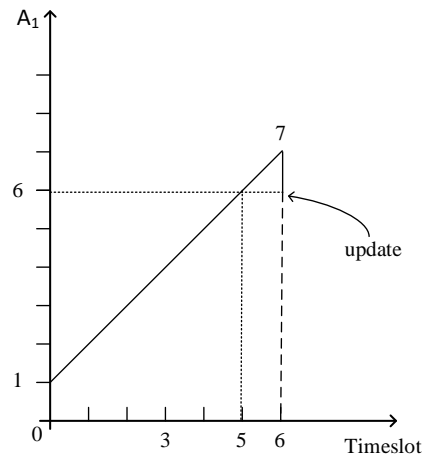
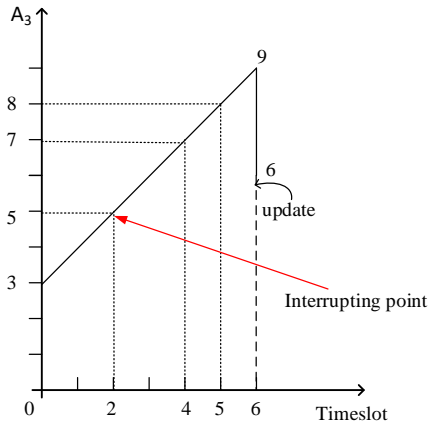
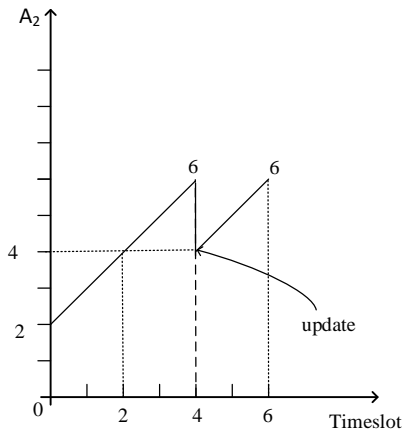
(a) S_3 : start transmission at $t = 0$ (b) S_2 : start transmission at $t = 3$ (c) S_1 : start transmission at $t = 5$

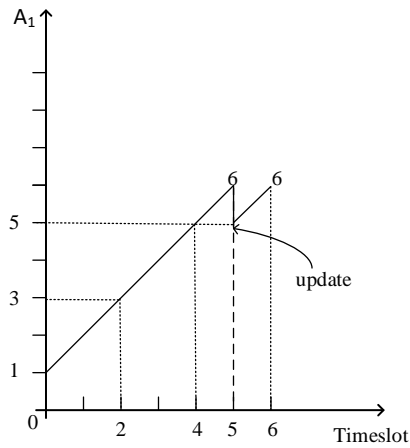
Figure 3.1: Consecutive Transmission: Each source transmits all packets and update the age, then the next one. Update order: $S_3 \rightarrow S_2 \rightarrow S_1$



(a) S_3 : start transmission at $t=0$, interrupted at $t=2$, re-start at $t=5$



(b) S_2 : start transmission at $t=2$



(c) S_1 : start transmission at $t=4$

Figure 3.2: Transmission with interruption: Let S_3 transmits 2 packets first, and then interrupted by S_2 . S_2 transmits all 2 packets, followed by S_1 . Finally, S_3 starts transmitting the last packet. Update order: $S_3(\text{interrupted}) \rightarrow S_2 \rightarrow S_1 \rightarrow S_3$

We plot the process of AoI evolution using *consecutive transmission* and *transmission with interruption* in Figure 3.1 and 3.2, respectively. We calculate the total age as follows.

1. For consecutive transmission:

$$\begin{aligned}
Total &= Total_{S_3} + Total_{S_2} + Total_{S_1} \\
&= \left(\frac{(3+6) \times 3}{2} + \frac{(3+6) \times 3}{2} \right) \\
&\quad + \left(\frac{(2+7) \times 5}{2} + \frac{(5+6) \times 1}{2} \right) \\
&\quad + \frac{(1+7) \times 6}{2} \\
&= 27 + 28 + 24 \\
&= 79
\end{aligned} \tag{3.1}$$

2. For transmission with interruption:

$$\begin{aligned}
Total &= Total_{S_3} + Total_{S_2} + Total_{S_1} \\
&= \frac{(3+9) \times 6}{2} \\
&\quad + \left(\frac{(2+6) \times 4}{2} + \frac{(4+6) \times 2}{2} \right) \\
&\quad + \left(\frac{(1+6) \times 5}{2} + \frac{(5+6) \times 1}{2} \right) \\
&= 36 + 26 + 23 \\
&= 85
\end{aligned} \tag{3.2}$$

Obviously, the total age achieved with *consecutive transmission* is smaller. Consequently, the average AoI is lower over the time interval.

In addition, we repeat the simulation of *transmission with interruption* in Figure 3.2 by letting only 1 packet of S_3 transmitted when it's interrupted or letting S_1 interrupt the transmission of S_2 . Finally, the system total age obtained is 82 and 85, both larger than that obtained in *consecutive transmission*.

The above example illustrates that in our context, *consecutive transmission* can achieve lower average age than *transmission with interruption*.

3.2 Proof

We first consider a simple but basic case where all the information generated at the beginning of an interval is transmitted in the interval.

Lemma 1. *Assume that all the information generated at the beginning of an interval is transmitted in the interval. An optimal solution for minimizing the EAvgAoI defined in (2.12) has the following property: once the BS transmits the information of a source, if any, the BS should continuously transmit all the packets in the information without switching to transmit information of another source.*

Proof. Take an arbitrary interval and assume that the BS has the information from two sources s_1 and s_2 at the beginning of this interval, with information size of l_1 and l_2 , respectively. Assume that the initial age of s_1 and s_2 at the beginning of the interval is $A_1(0)$ and $A_2(0)$, respectively¹. Without loss of generality, assume that the BS finishes the information of s_1 before the information of s_2 . We can calculate the total AoI of the system in this interval. In particular, the total AoI of s_1

$$\begin{aligned} \sum_{j=1}^T A_1(j) &= \frac{(A_1(0) + (A_1(0) + l_1 + x))(l_1 + x)}{2} \\ &\quad + \frac{(l_1 + x + T)(T - l_1 - x)}{2} \\ &= A_1(0)(l_1 + x) + \frac{T^2}{2}, \end{aligned} \tag{3.3}$$

where $x < l_2$ denotes the number of packets of source s_2 delivered before finishing s_1 's information. The above equation calculates the size of the shaded area in Fig. 3.3(a). The total AoI of s_2 , irrelevant to the scheduling policy (since its AoI is updated only after both have been delivered), is:

$$\begin{aligned} \sum_{j=1}^T A_2(j) &= \frac{(A_2(0) + (A_2(0) + l_1 + l_2))(l_1 + l_2)}{2} \\ &\quad + \frac{(l_1 + l_2 + T)(T - l_1 - l_2)}{2}, \end{aligned} \tag{3.4}$$

which is the size of the shaded area in Fig. 3.3(b).

Hence, the total AoI of the system in this interval is: $\mathcal{S}(x) = \sum_{j=1}^T A_1(j) + A_2(j)$.

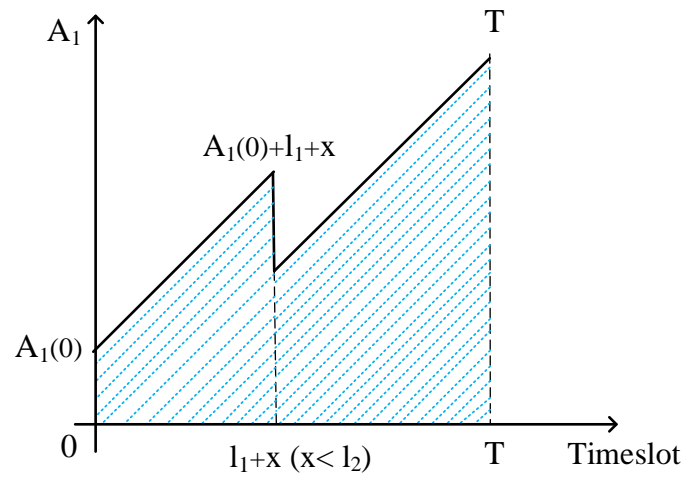
¹Since AoI is viewed at the destinations, the destinations should piggyback the age information to the BS in acknowledgements.

Minimizing $\mathcal{S}(x)$, we get $x = 0$. This means that continuous transmission of s_1 's information leads to a total AoI no larger than that from non-continuous transmission.

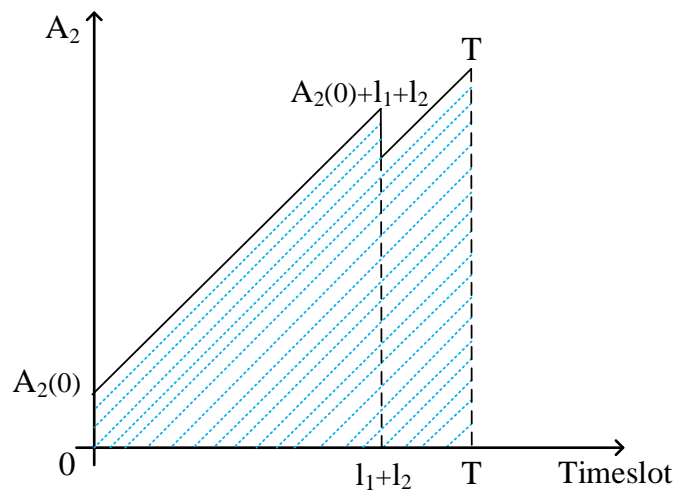
Now assume that the BS has information of $m(> 2)$ sources at the beginning of the interval. For any two sources among them, s_1 and s_2 , assume that the BS finishes s_1 's information before s_2 's information without loss of generality. If s_1 's and s_2 's information transmission times do not overlap each other, we do not need to do any adjustment. Otherwise, we can adjust their transmission order such that s_1 's information is transmitted continuously before s_2 's information being transmitted. To be more specific, if the k -th packet s_2^k from s_2 is transmitted during the transmission of s_1 's information, the adjustment simply switches the order by moving s_2^k right before s_2^{k+1} , as shown in Fig. 3.4. Based on the two-source case analysis, the above adjustment lead to a total AoI no larger than that before the adjustment. The lemma thus holds. \square

Remark 2. *Lemma 1 does not imply that our problem is equivalent to earliest deadline first (EDF) or largest/smallest job first in a system with single-packets of various length. It is easy to give counterexamples. For instance, consider a simple network with two sources S_1 and S_2 . The information size of source S_1 is 3 and the information will be outdated in 6 timeslots; the information size of S_2 is 2 and the information will be outdated 10 timeslots. Using the EDF scheduling policy, source S_1 has a higher priority. Figure 3.5 and Figure 3.6 plot the scheduling results with EDF and with Lemma 1, respectively. It can be seen that EDF leads to a higher total age ($33 > 32$).*

While Lemma 1 is based on the assumption that all the information generated at the beginning of an interval is transmitted in the interval, we can use this lemma to design a scheduling algorithm by relaxing this assumption with information carryover: (a) we find the transmission orders for the current interval by enumerating the results for all possible transmission orders; (b) if some information cannot be transmitted in the current interval with the optimal order, we carry the remaining packets and the age of their corresponding sources, i.e., the pairs (A_i, l_i) where A_i denotes the age of source s_i and l_i denotes number of packets remaining at the end of the current interval, into the next interval. The carryover information is outdated and replaced if new information from the same source arrives in the next interval. Repeat steps (a) and (b) as time goes. The optimal schedule is the one that leads to the smallest $EAvgAoI$.

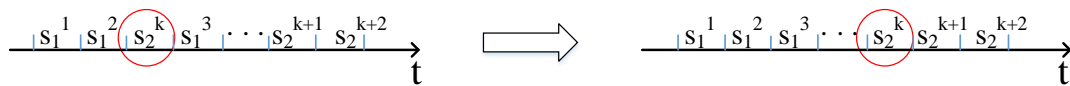


(a)



(b)

Figure 3.3: Illustration of AoI changes of two sources.

Figure 3.4: Adjustment of transmission schedule (s_1^1, s_1^2, \dots are packets in the information of s_1 , and s_2^1, s_2^2, \dots are packets in the information of s_2).

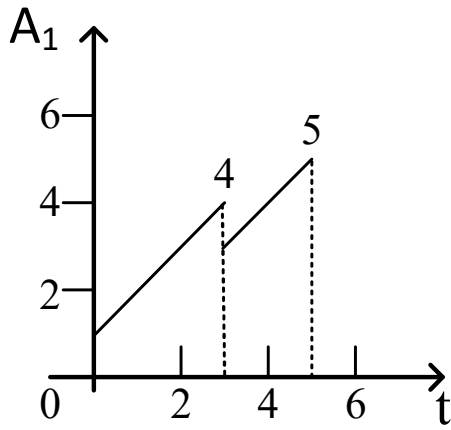
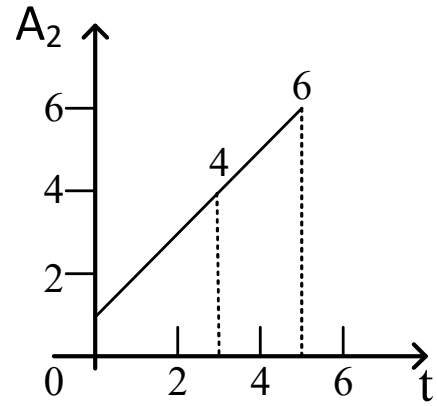
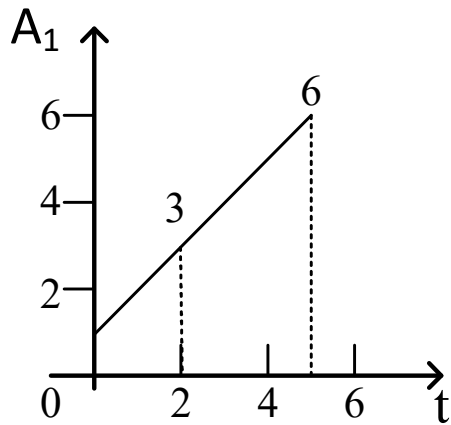
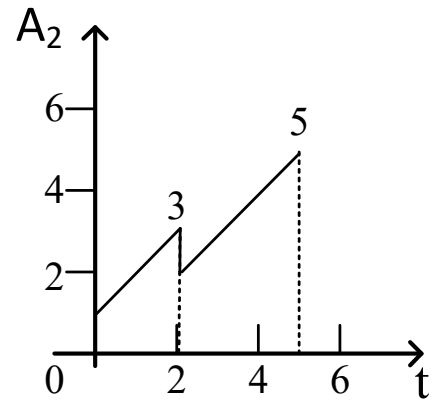
(a) S_1 : start transmission at $t = 0$ (b) S_2 : start transmission at $t = 3$

Figure 3.5: Scheduling with earliest deadline first (EDF).

(a) S_1 : start transmission at $t = 2$ (b) S_2 : start transmission at $t = 0$ Figure 3.6: Scheduling with Lemma 1 where BS transmits information of S_2 first.

We call this algorithm *Opt*. It is optimal, since it is essentially a brute-force search for all possible transmission orders. The worst-case complexity of *Opt* is $O((N!)^K)$. We would like to raise the awareness that the optimal transmission orders in each interval together do not necessarily lead to the global optimal, because an optimal transmission order in the current interval may lead to initial AoI values which are not optimal for the next interval. Due to this reason, brute-force search is necessary for guaranteeing global optimality.

Chapter 4

Index-based Scheduling Policy Design and Analysis

As analyzed in Sec. 3.2, the *Opt* solution has a pretty huge search space that includes all possible scheduling order combinations. Searching the whole space to find the optimal order is complicated and time consuming. Our goal in this thesis is to design a low-complexity policy that can optimize the scheduling process and hence minimize the Expected Average AoI. We want to analyze the scheduling decision and then design a close to optimal scheduling policy based on the framework of *restless bandits* [33].

In this section, we first introduce the basic knowledge of multi-armed bandit problem. Considering the hardness of tracking AoI measured in terms of timeslot, we propose an indirect way, using the *time since last update*, to approximate the AoI, which is also effective to minimize the expected average AoI. By using the *time since last update*, we can easily transform our problem into a *restless multi-armed bandit* problem (RMBP) and develop a scheduling policy with Whittle Index methodology. In addition, we explicitly derive the Whittle index, which can be used to schedule the transmissions in our system.

4.1 Introduction of Multi-armed Bandit Problem

Consider a sequential decision problem, where the agent must select an action from a set of n available actions at each time, knowing the “state” of each action. The action selected reveals some information about the action and the agent will receive

a corresponding payoff by performing the action. The states of actions may change over time and the information received by the agent may help to decide the action selection in the future. The goal of the agent is to maximize the total payoffs that would be received by choosing the right sequence of actions. This problem is known as the “bandit” problem in the literatures [5] [12] [32].

The multi-armed bandit problem (MAB) was first introduced by Robbins in [29], where a gambler has to decide which arm of K different slot machines to play in a sequence of trials so as to maximize the reward. This classical problem has received much attention, because the simple model provides the trade off between *exploration* (trying out each arm and find the one with best reward) and *exploitation* (playing the arm believed to return the best payoff). Each selected arm will result in an immediate random payoff (possibly zero or negative), while the process determining these payoffs evolves during the play of the bandit. The distinguishing feature of bandit problems is that the distribution of returns from one arm only changes when that arm is selected for playing. Hence, the rewards from an arm are independent from those of the other arms.

4.2 Approximating AoI

In Figure 2.3, we plotted the age evolution in our system model. The information I_i (if any) is randomly generated at the beginning of an interval, and can start delivering at arbitrary timeslot within that interval. Unlike previous work such as [15] and [17], the information I_i is considered complete (from the perspective of recipient) if and only if all packets have been received. Also, due to the fact that the information size is non-uniform, the transmission time for a whole piece of information from different sources may be different. It is thus challenging to track the exact value of AoI_i directly, since we need to not only monitor the starting time of transmission of information I_i , but also the ending time, as well as the information occurrence distribution of other sources.

To address this difficulty, we introduce a variable $h_{i,k}$ to quantify the time that has passed since the last information update of s_i in the k -th interval. The value of $h_{i,k}$ is measured in terms of interval and checked for update at the end of k -th

interval. In particular, we have

$$h_{i,k+1} = \begin{cases} 1, & \text{if } \sum_{j=1}^T u_i(t_{k,j}) \geq 0 \\ h_{i,k} + 1, & \text{otherwise} \end{cases} \quad (4.1)$$

Note that $\sum_{j=1}^T u_i(t_{k,j}) > 0$ means that in the k -th interval, the BS transmits s_i 's information and thus it must have information to deliver at the beginning of this interval. Due to the schedulability assumption (Remark 1), we assume¹ s_i 's information can be updated in the k -th interval and thus $h_{i,k+1}$ is reset to 1. Otherwise (i.e., $\sum_{j=1}^T u_i(t_{k,j}) = 0$), the BS has no information of s_i at the beginning of the k -th interval and thus $h_{i,k+1} = h_{i,k} + 1$.

Therefore, we then use $h_{i,k}$ to estimate the value of AoI_i . To be more specific

$$AoI_i(t_{k,j}) \sim h_{i,k} \cdot T, \forall j \in \{1, \dots, T\} \quad (4.2)$$

Since the value of $h_{i,k}$ is updated only at the end of an interval, the above approximation allows an easy calculation of AoI_i and tractable performance analysis in the following section. In addition, the maximum error of the above approximation is upper bounded by the length of time interval, T .

Remark 3. *The approximation of AoI with $h_{i,k}$ is to ease analysis, based on which we can design effective algorithm (i.e., index policy) to solve the scheduling problem raised in Section 2.3. In the actual performance evaluation in Chapter 6, however, we calculate the exact AoI value whenever all packets in an information have been received.*

Next, we develop a low-complexity scheduling algorithm whose $E Avg AoI$ is close to the minimum by leveraging the Whittle's methodology.

4.3 Algorithm Analysis and Design

In accordance to the approximation of AoI, we can simplify the representation of age when we try to analyze and design the optimal scheduling policy. In this section, we

¹This assumption is to simplify analysis and allows us to develop an effective index-based policy. It is reasonable since $1 \geq \sum_{i=1}^N p_i \frac{E[r_i]}{T}$ due to Remark 1. Our later simulation-based evaluation, however, does not depend on this assumption.

first introduce the *restless multi-armed bandit* (RMAB) framework and show how our problem is mapped to RMAB. Then, we propose an optimal scheduling policy with Whittle index [33].

4.3.1 Restless Multi-armed Bandit Problem (RMAB)

RMAB is a generalization of the classical *multi-armed bandit problem* (MAB) [31]. In MAB, a player, with full knowledge of the current state of each arm, chooses one out of N arms to activate at each time and receives a reward determined by the state of the activated arm. Only the activated arm may change state and the states of passive arms remain unchanged. Whittle generalized MAB to RMAB by allowing M ($1 \leq M \leq N$) arms to be played simultaneously and allowing passive arms to change states even if they are not played. In general, RMAB has been shown to be PSPACE-hard by Papadimitriou and Tsitsiklis in [27]. Hence, Whittle proposed an optimal index policy for the RMAB problem under a relaxed constraint: the number of activated arms can vary over time but its average over the infinite horizon equals M . With this relaxation, Whittle then applied the Lagrangian approach to decouple the MAB problem into multiple sub-problems, namely the decoupled model.

It is easy to see that our problem belongs to the relaxed RMAB, since (1) we can regard each source as an arm and all arms are restless, and (2) an action for each arm is made in *each interval*, and the value of $\sum_{i=1}^N p_i$ means the average number of active arms in the long term. Refer to the next subsection for the bandit model and the actions.

Hence N sub-problems can be decoupled by applying Whittle's approach. Each sub-problem corresponds to the BS's transmission for the information of a single source s_i and adheres to the network model in Sec. 2.3.1. To obtain the optimal packet transmission policy, our goal becomes to find a way of calculating the Whittle index that matches our context. To be more specific, the resulting Whittle index should help us determine BS should transmit the information of which source at the beginning of *each timeslot*.

4.3.2 Decoupled Model and the Properties of Optimal Policy

Using the AoI approximation in Sec. 4.2, we can analyze the dynamics of variable $h_{i,k}$ to develop an index policy. Since each sub-problem corresponds to only one source, in the following analysis we *omit the source index i* for simplicity. We first transform

the sub-problem into a Markov Decision process (MDP) [28], with basic components (states, actions, transitions, and objective) defined as follows

- **States:** To characterize the state $s(k)$ of the MDP, we define $s(k) = (\Lambda(k), h_k)$, where $\Lambda(k)$ indicates whether or not information is generated at the beginning of the k -th interval. Theoretically, the number of states could be infinite because the age is possibly unbounded.
- **Actions:** We use $a(k) \in \{0, 1\}$ to denote the action taken in the k -th interval. To be specific, $a(k) = 1$ if the source's status is updated and $a(k) = 0$ if not.
- **Transition:** Given the action $a(k) = a$, the state transition probability from current state $s = (\Lambda, h)$ to next state s' is as follows.

1. if action $a = 1$:

$$\begin{aligned} P[s' = (1, h + 1) \leftarrow s = (0, h)] &= p; \\ P[s' = (0, h + 1) \leftarrow s = (0, h)] &= 1 - p; \\ P[s' = (1, 1) \leftarrow s = (1, h)] &= p; \\ P[s' = (0, 1) \leftarrow s = (1, h)] &= 1 - p; \end{aligned}$$

2. if action $a = 0$:

$$\begin{aligned} P[s' = (1, h + 1) \leftarrow s = (\Lambda, h)] &= p; \\ P[s' = (0, h + 1) \leftarrow s = (\Lambda, h)] &= 1 - p; \end{aligned}$$

- **Objective:** There are two actions that the edge server can take, active ($a = 1$) or passive ($a = 0$). Action $a = 1$ will incur a service cost for transmitting information; and action $a = 0$ will incur a cost of information aging for not transmitting the information. Combining these two types of costs, we define the total cost of executing arbitrary action a given state $s = (\Lambda, h)$ as follows

$$\Delta(s, a) \triangleq ((1 - \Lambda \cdot a) \cdot h + 1) \cdot \omega + C \cdot r \cdot a \cdot (1 - \omega) \quad (4.3)$$

where $(1 - \Lambda \cdot a) \cdot h + 1$ represents the age evolution of h if the action $a(k) = a$ is taken, C represents the cost for transmitting one packet, and $\omega \in (0, 1)$ is a weight parameter.

Based on the definition of the related components, we can define the objective function under policy π as

$$\Upsilon^\pi = \frac{1}{K} E \left[\sum_{t=1}^K \Delta(s(k), a(k)) \right] \quad (4.4)$$

A policy is called cost-optimal if it minimizes the average cost defined by (4.4). It is proved in [14] that a cost-optimal policy is stationary and deterministic. In particular, a policy is stationary if for $\forall k_1, k_2 \in \{1, \dots, K\}$, we have $a(k_1) = a(k_2)$ when $s(k_1) = s(k_2)$. Taking the variable h that represents the time since the last update as an example, let π^* be the deterministic stationary policy which always performs the action $a(k) = 1$ for each time interval k if information is generated at the beginning of that interval. The evolution of h under the policy π^* forms a discrete-time Markov chain (DTMC) as shown in Figure 4.1.

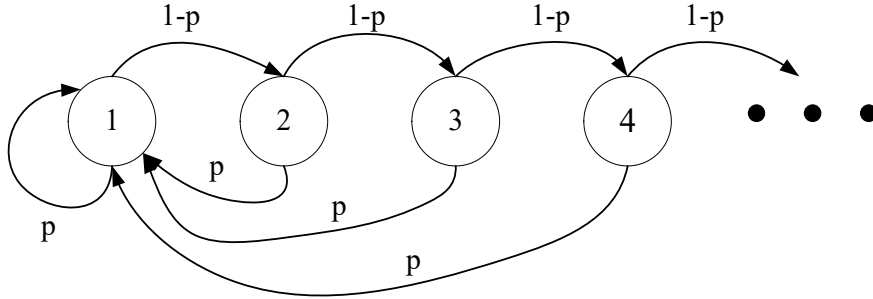


Figure 4.1: The Approximation Age h_k under the policy π^* forms a DTMC.

Also, recall that we always update the value of h at the end of the k -th interval. Hence during an arbitrary interval, the optimal policy either idles in every timeslot, or transmits until the information is delivered. Hence, the policy is still stationary, when we consider it at the level of a timeslot.

For analyzing the system in steady-state, we extend K to an infinite horizon. Then, the Bellman equations can be formulated as

$$\theta(s, a) + \beta = \min_{a \in \{0,1\}} \{U_a(s)\} \quad (4.5)$$

where $\theta(s, a)$ is the cost-to-go function and β is the optimal average cost. More specifically

$$\begin{aligned} U_0(s) &= p \cdot \theta((1, h+1), a) + (1-p) \cdot \theta((0, h+1), a) + h+1; \\ U_1(s) &= p \cdot [\theta((1, h+1), a) + \theta((1, 1), a)] \\ &\quad + (1-p) \cdot [\theta((0, h+1), a) + \theta((0, 1), a)] + r \cdot C + 1 \end{aligned} \quad (4.6)$$

Further, we show that the deterministic stationary scheduling policy is threshold-type. Obviously, for state $s(0, h)$, the optimal action is to be idle (since $C \geq 0$). For state $s(1, h)$, if we assume the optimal action is to update, i.e.

$$U_1(s(1, h), 1) - U_0(s(1, h), 0) \leq 0 \quad (4.7)$$

Then, for state $s(1, h+1)$, we have

$$\begin{aligned} &U_1(s(1, h+1), 1) - U_0(s(1, h+1), 0) \\ &= (rC + 1 + E[J(s')]) - (h+2 + E[J(s')]) \\ &\stackrel{(nd)}{\leq} (rC + 1 + E[J(s')]) - (h+1 + E[J(s')]) \\ &= U_1(s(1, h), 1) - U_0(s(1, h), 0) \\ &\leq 0 \end{aligned} \quad (4.8)$$

where $E[J(s')]$ is the expectation taken over all next state s' that is possibly reachable from state s , and (nd) results from the non-decreasing property of $E[J(s')]$ [14]. Hence, we can claim that the cost-optimal policy is also threshold-type.

By applying the threshold-type scheduling policy, the BS server is idle when $1 \leq h < \bar{H}$ and transmits if $h \geq \bar{H}$, where $\bar{H} \in \{1, 2, \dots\}$ is the threshold. This is illustrated in Figure 4.2. Next, we explicitly investigate the relation between the average cost and the threshold, based on which we derive the Whittle index.

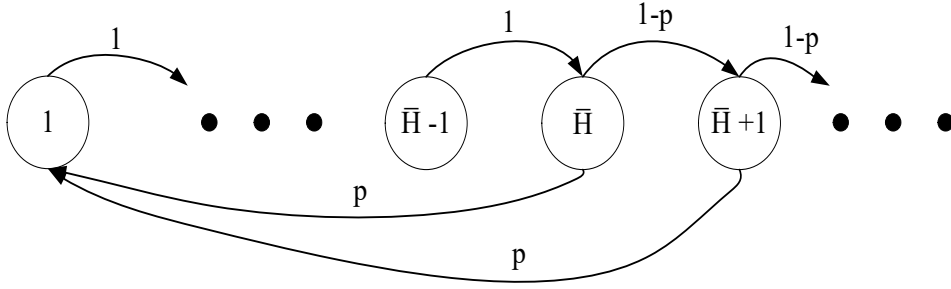


Figure 4.2: The Approximation Age h_k under the threshold-type policy forms a DTMC.

Remark 4. *The Markov Decision Process introduced as above does not directly indicate transmission scheduling. Instead, it can help design an index policy (Sec. 4.3.3), based on which the transmission schedule is made for each timeslot (Sec. 4.4).*

4.3.3 Derivation of the Whittle Index

Prior to derive the Whittle index, we first form a discrete-time Markov Chain (DTMC) that depicts the state transition of the variable h . Within each interval, an action will be taken and incur different costs. For example, the DTMC incurs the cost of $rC + 1$ when h is reduced to 1 at the end of the interval, which means a transmission opportunity is allocated and hence the AoI will be updated. Otherwise, the service cost is not considered, while h increases by 1. Hence, the state space of the DTMC is $\{1, 2, \dots, \bar{H}, \dots\}$.

Lemma 2. *Denote the steady-state distribution of the above DTMC as $Q = \{q_1, q_2, \dots\}$.*

We have

$$q_i = \begin{cases} \varphi(\bar{H}), & \text{if } i \leq \bar{H} \\ \varphi(\bar{H})(1-p)^{i-\bar{H}} & \text{otherwise} \end{cases} \quad (4.9)$$

where $\varphi(x) = \frac{p}{1+p(x-1)}$ and \bar{H} is the threshold since the transition probability is deterministic 1 while $h \leq \bar{H}$.

Proof. In accordance with the *threshold-type* policy, we know $q_1 = q_2 \dots = q_i$ when $i \leq \bar{H}$. Once h is larger than \bar{H} , it continues to increase with probability $1-p$ and reduces to 1 with probability p . Hence, $q_i = q \times (1-p)^{i-\bar{H}}$ when $i > \bar{H}$. Here,

$q = q_1 = \dots = q_i$ for $\forall i \leq \bar{H}$. Since the sum of the corresponding probability in Q must equal 1, we have

$$\lim_{n \rightarrow \infty} \bar{H} \times q + q \times (1-p) + \dots + q \times (1-p)^{n-\bar{H}} = 1 \quad (4.10)$$

Solving (4.10), we obtain that $q = \frac{p}{1+p(\bar{H}-1)}$. The lemma thus holds. \square

Therefore, the average cost can be calculated as the expectation over all states

$$\begin{aligned} \Phi(\bar{H}) &= \sum_{i=2}^{\bar{H}} i \cdot \varphi(\bar{H}) \cdot \omega + (1 \cdot \omega + C \cdot r \cdot (1-\omega)) \cdot \varphi(\bar{H}) \\ &+ \sum_{i=\bar{H}+1}^{\infty} i \cdot \varphi(\bar{H}) \cdot (1-p)^{i-\bar{H}} \cdot \omega \\ &= \frac{(\frac{\bar{H}^2}{2} + (\frac{1}{p} - \frac{1}{2})\bar{H} + \frac{1}{p^2} - \frac{1}{p})\omega + rC(1-\omega)}{\bar{H} + \frac{1-p}{p}} \end{aligned} \quad (4.11)$$

We can regard the threshold as a variable of h , and let $f(h) = \Phi(h)$. Note that $f(h)$ is strictly convex in the domain. Let h^* be the value observed at the minimization point of $f(h)$. Then, the value of an optimal threshold for minimizing the average cost $\Phi(\bar{H})$ is either $\lfloor h^* \rfloor$ or $\lceil h^* \rceil$

$$\bar{H}^* = \begin{cases} \lfloor h^* \rfloor, & \text{if } \Phi(\lfloor h^* \rfloor) \leq \Phi(\lceil h^* \rceil) \\ \lceil h^* \rceil & \text{if } \Phi(\lceil h^* \rceil) \leq \Phi(\lfloor h^* \rfloor) \end{cases} \quad (4.12)$$

Thus, both actions for state $s = (1, h)$ are equivalent if $\Phi(h) = \Phi(h+1)$. By solving this equation, we can explicitly derive the Whittle index, denoted by $I(s = (\Lambda, h))$ as

$$I(\Lambda, h) = \left(\frac{h^2}{2r} - \frac{h}{2r} + \frac{h}{pr} \right) \cdot \frac{\omega}{1-\omega} \quad (4.13)$$

Let $\mathcal{P}(C)$ be the set of system states where the optimal action is to idle when the service cost is C , for $\forall s \in \mathcal{P}(C)$. Next, we give the definition of indexability.

Definition 2. (Indexability:) *If $\mathcal{P}(C)$ increases monotonically from \emptyset to the entire state space as C increases from 0 to $+\infty$, we say that the decoupled problem*

corresponding to the information of source s_i is indexable. The age of information minimization problem is called indexable if the decoupled problem is indexable for every source.

From (4.11) and (4.13), we know that the cost C is a function of the threshold of h . In addition, by calculating the derivative of (4.13), we have

$$\frac{\omega}{(1-\omega)^r} \cdot \left(h - \frac{1}{2} + \frac{1}{p}\right) > 0, \quad (4.14)$$

since h is non-negative integer, $\omega \in (0, 1)$ and $p \in (0, 1]$. Hence, the index function monotonically increases. Moreover, substituting $h = 0$ yields $C = 0$, implying that $\mathcal{P}(C) = \emptyset$, and $h \rightarrow +\infty$ results in $C \rightarrow +\infty$, consequently, $\mathcal{P}(C) = \mathbb{N}$. Thus, the decoupled problem associated with source s_i is indexable. Considering that source s_i is arbitrary, the original AoI minimization problem is hence indexable.

4.4 Scheduling Algorithm with the Whittle Index

We next design the scheduling policy based on the Whittle index. Taking $I(\Lambda, h_k)$ as the Whittle index for each source s_i , the index policy works as follow: At the beginning of each timeslot, we select the source with the maximum Whittle index for transmission (randomly select one if there is a tie), and if the source with the maximum Whittle index has finished transmission of all packets for the current information, select the source with the next largest Whittle index to transmit. Repeat the above process until information for a candidate source is found or the BS has no information to send. An important advantage of the index policy is the low complexity in scheduling packet transmissions, especially when the information is composed of multiple packets. Intuitively, we can think of the index $I(\Lambda, h_k)$ as the cost to update for a source, and the scheduling algorithm is to select the most valuable packet to transmit. The pseudo code of the Whittle index-based scheduling is shown in Algorithm 1. Note that in line 4, the condition that “if source i has information” includes both scenarios:

- A new piece of information arrives at the beginning of the time interval;
- No new information arrives at the beginning of the time interval and the most recent information transmission is incomplete. Hence, the information is the one inherited.

Algorithm 1 Whittle Index-based Scheduling Policy

Input: a series of sources

```

1: initialization  $k \leftarrow 0$ ;
2: calculate Whittle index;
3: for interval index  $k \leq K$  do
4:   if source  $i$  has information then
5:     set the flag as True
6:   else
7:     set the flag as False
8:   initialization  $j \leftarrow 0$ 
9:   for timeslot index  $j \leq T$  do
10:    select the source with maximum index
11:    if the flag of the selected source is True then
12:      transmit a packet
13:      if the transmitted packet is the last one of  $I_i$  then
14:        set the flag as False
15:       $j \leftarrow j + 1$ 
16:    for each source do
17:      if source  $i$  get updated in interval  $k$  then
18:        set the variable  $h_{i,k+1} \leftarrow 1$ 
19:      else
20:        set the variable  $h_{i,k+1} \leftarrow h_{i,k} + 1$ 
21:    calculate Whittle index;
22:     $k \leftarrow k + 1$ 

```

4.5 Why Does Our Solution Work Well

Since we do not update the Whittle index until the end of the interval, within each interval, once the BS decides to transmit information for a source, it continuously transmits all the packets in the information. This matches the necessary condition of an optimal solution stated in Lemma 1. In contrast, other existing baseline methods, such as the Greedy Policy (GP) [18] and the Naïve Whittle Index Policy (NWIP) as shown in the later evaluation (Sec. 6), do not meet this necessary condition. This partially explains why our method outperforms other existing methods.

Chapter 5

Erasure Code-Aided AoI Minimization in Lossy Networks

In this chapter, we relax the assumption of guaranteed packet transmission in each timeslot. Instead, we assume a lossy network where there is channel noise. Thus, packets may be dropped during the transmission process or arrive at the destination with bit error. In this case, it will still be dropped since packet containing incorrect message loses its value. Only those packets received without error are accepted at the destinations. By introducing *erasure code* method, we can improve the performance of Whittle Index-based scheduling policy in lossy networks.

To quantify the channel quality, let q ($0 < q < 1$) be the *bit error rate* (BER) associated with the wireless channel. Assume that a sample information M needs to be transmitted to the recipient and the size of this information, measured in terms of *bits*, is m . Then the probability that the information is successfully delivered to the recipient without error is

$$\lambda = (1 - q)^m \tag{5.1}$$

Notice that, $q = 0$ means the channel is noiseless and the sample information is guaranteed to be delivered to the corresponding recipient with no errors, which is the case studied in Chapter 4. Next, we introduce two transmission schemes that can be used in lossy networks.

5.1 Regular Transmission

Recall that in our system model, each piece of information consists of r packets. The recipient cannot update the AoI until all packets are received. Considering the unreliability of communication channel, packet may fail the transmission attempt and the sender has to re-transmit it. As a result, the transmission of next packet will not start until the current packet has been successfully received by the recipient without error, which is indicated by an *acknowledge message* sent from the receiver and confirmed by the sender. We name this transmission scheme as *Regular Transmission* (Reg Trans).

Let X and Y be the number of bits contained in each packet and acknowledge message, respectively. According to (5.1), the *packet delivery probability* (PDP)¹ of *Reg Trans* scheme is

$$\lambda_1 = (1 - q)^{X+\alpha Y} \quad (5.2)$$

where α is the total amount of *acknowledge message* used to confirm that a packet has been successfully received by the recipient without bit error. Since one packet is confirmed by one *acknowledge message*, we set $\alpha = 1$ in our system, and omit it thereafter.

5.2 Introduction of Erasure Code and Transmission Scheme

5.2.1 Erasure Code

Erasure code (EC) is a technology of data protection and recovery in which data is broken into smaller fragments, which are encoded with redundant codes. It has been widely used in fields such as data storage system [10, 25]. The key idea behind erasure code is that k blocks of data are expanded into n ($n > k$) blocks of encoded data, such that *any* subset of k encoded blocks suffices to reconstruct the original data. Such a code is named as (n, k) code and allows the recipient to recover from up to $n - k$ losses in a group of n encoded blocks of data. A graphical illustration of the encoding and decoding process is shown in Figure 5.1.

¹The probability that a packet is successfully received by the receiver without bit error.

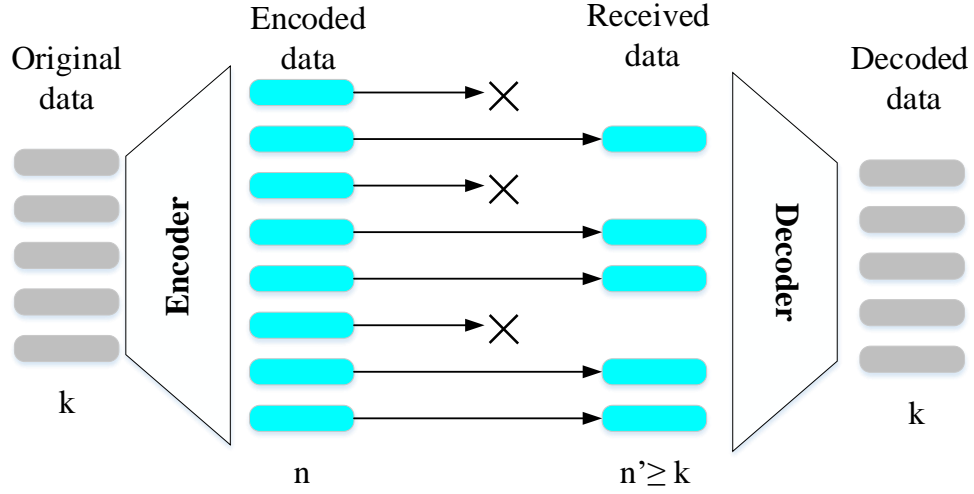


Figure 5.1: A graphical illustration of the encoding and decoding process of erasure code.

One such code is Reed-Solomon code, whose details can be found in [34]. We omit the encoding/decoding details since they are out of the focus of the thesis.

5.2.2 Erasure Code-based Transmission

Letting $k = r$, we can regard each packet as a block of data. After encoding, the r original packets are expanded and encoded to n ($n > r$) new packets. To deliver the information, we transmit the n encoded packets without retransmission if a transmission fails. As such, the source node does not require the destination node to send back *acknowledgement* to confirm the correct reception of every packet. Instead, as soon as any r (out of the n) packets are successfully received by the destination, the destination will send an *acknowledgement* to terminate the transmission session and then start reconstructing the source information and updating AoI. As a result, at most n timeslots are needed to complete the transmission process. We name this transmission scheme as *Erasure Code-based Transmission* (EC Trans). Since packets (except the last one in an information piece) are transmitted without requirement of *acknowledgement*, the PDP of *EC Trans* scheme is:

$$\lambda_2 = (1 - q)^X. \quad (5.3)$$

Since for each information (instead of each packet) the receiver needs to send back an *acknowledgement* to the source node, we denote

$$\lambda_3 = (1 - q)^Y. \quad (5.4)$$

5.3 Analysis and Estimation

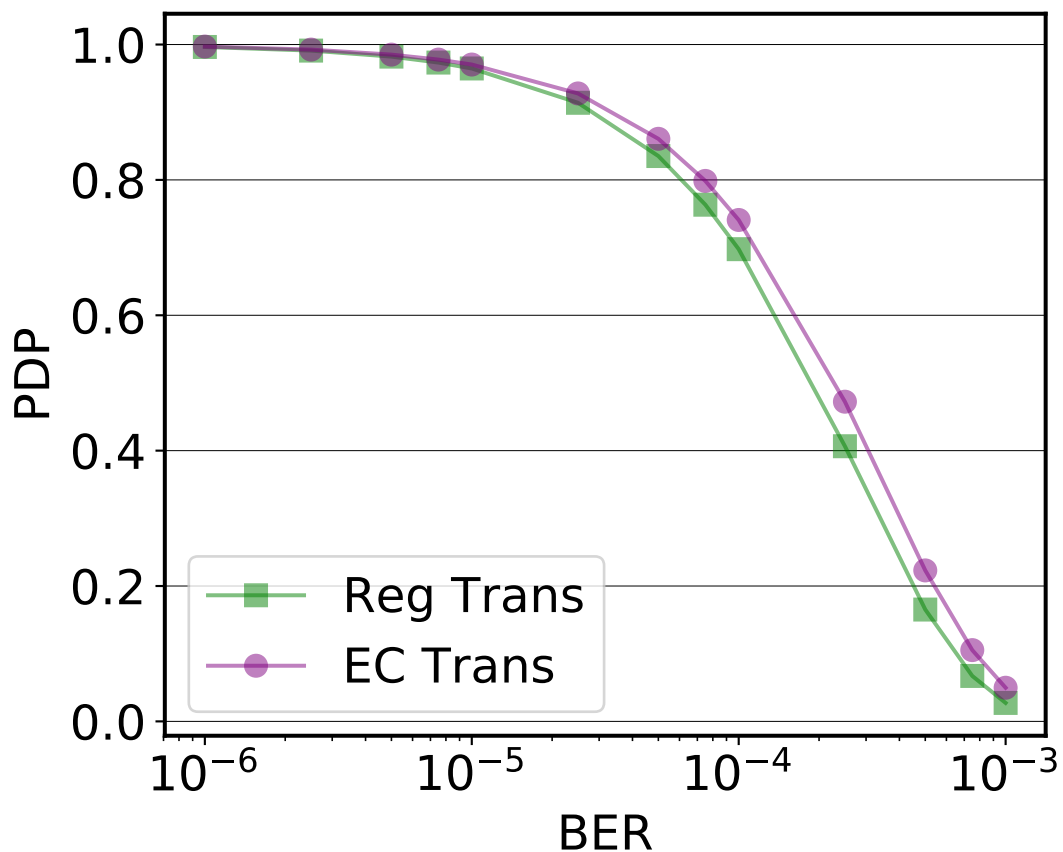


Figure 5.2: Illustration of effect of BER on PDP.

Due to the fact that $X, Y > 1$, we have $\lambda_2 > \lambda_1$ for $\forall q > 0$. By setting $X = 4000$ and $Y = 400$, we plot Figure 5.2 to demonstrate the effect of BER on PDP.

Obviously, *EC Trans* achieves a higher PDP, but when the bit error rate is extremely low or extremely high, the difference can be ignored. Especially, when the

BER is between 10^{-4} and 10^{-3} , a big difference can be obtained. However, this difference can change if the values of X and Y are changed.

We next calculate the expected time for successfully delivering information, namely *Expected Transmission Time* (ETT). To avoid unfairly penalizing *Reg Trans*, we assume that the data/acknowledgement two-way handshake can be finished within one timeslot.

- For *Reg Trans* scheme, the expectation of successful single packet transmission time is $1/\lambda_1$. The transmission of each packet is independent, hence the total *expected transmission time* of a complete piece of information is

$$E^{Reg}[T] = \frac{r}{\lambda_1} \quad (5.5)$$

- For *EC Trans* scheme, as long as any k of the n encoded packets are successfully received without bit error, the transmission process is terminated. Hence, the expected transmission time can be calculated as follows

$$\begin{aligned} E^{EC}[T] &= \lambda_3 \{ k \lambda_2^k \\ &+ (k+1) \binom{k}{k-1} \lambda_2^{k-1} (1-\lambda_2) \lambda_2 \\ &+ (k+2) \binom{k+1}{k-1} \lambda_2^{k-1} (1-\lambda_2)^2 \lambda_2 + \dots \\ &+ n \binom{n-1}{k-1} \lambda_2^{k-1} (1-\lambda_2)^{(n-1)-(k-1)} \lambda_2 \} \\ &+ (1-\lambda_3)n + \Gamma n \\ &= \lambda_3 \sum_{i=0}^{n-k} (k+i) \binom{k+i-1}{k-1} \lambda_2^k (1-\lambda_2)^i \\ &+ (1-\lambda_3 + \Gamma)n, \end{aligned} \quad (5.6)$$

where i represents the number of lost packets during the transmission process, Γ is the probability that no k (out of n) packets have been correctly received, i.e.

$$\Gamma = 1 - \sum_{i=0}^{n-k} \binom{k+i-1}{k-1} \lambda_2^k (1-\lambda_2)^i, \quad (5.7)$$

and λ_3 represents the probability that the BS correctly receives the *acknowledgement*. Note that the BS might continue sending packets (up to n) if the acknowledgement is lost.

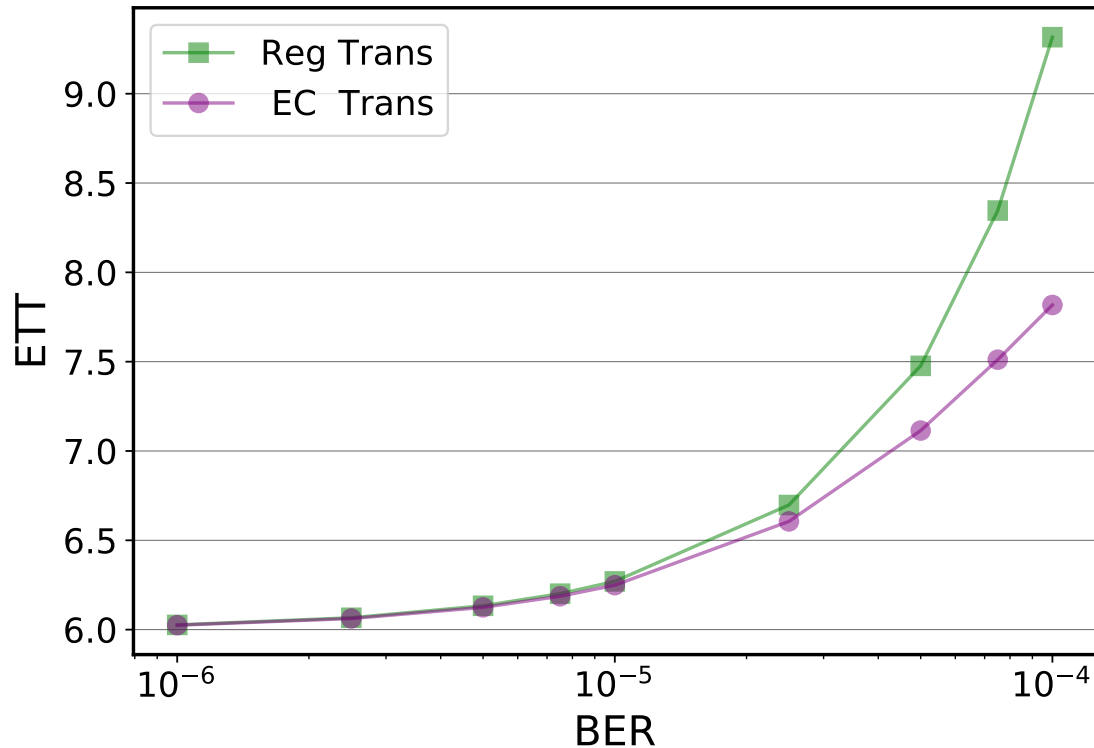


Figure 5.3: Illustration of effect of BER on ETT, (8,6) code is used for *EC Trans*.

Remark 5. Note that there is a subtle difference between the two schemes in handling transmission failures. *Reg Trans* keeps trying until all packets are successfully transmitted, while *EC Trans* stops trying after transmitting n packets. In this sense, the above comparison can only give us a rough idea on the transmission time of *Reg Trans* and *EC Trans*. It may be unfair to compare the expected transmission time, since in theory *Reg Trans* has infinitely long transmission time. Nevertheless, if information is stored in the BS for too long, it will be replaced by new information, meaning that *Reg Trans* would not result in infinite retransmission in practice.

As an example, we illustrate the ETT values obtained when $X = 4000$ bits, $Y = 400$ bits and BER varies from 10^{-6} to 10^{-4} in Figure 5.3. Obviously, *EC Trans* outperforms *Reg Trans* when the BER is high (e.g., higher than 10^{-5}). This makes

sense since high BER means multiple re-transmissions may be needed to successfully transmit a packet, resulting in a high ETT for *Reg Trans*.

Chapter 6

Simulation and Performance Evaluation

In this section, we evaluate the performance of our method, denoted as Multiple-packet Whittle Index Policy (MWIP), with simulation. We compare MWIP with the following two baseline scheduling policies:

- Greedy Policy (GP) [18]: A scheduling policy always selects the source with maximum current AoI for transmission.
- Naïve Whittle Index Policy (NWIP): In [14], the Whittle index has been used for AoI minimization when each packet includes complete information and thus causes a status update. This Whittle index policy cannot be applied directly to our case, and thus it is impossible to fairly compare the method in [14] with ours. Nevertheless, we can modify the method in [14] to be applicable to our case by disabling the update of status and the Whittle index if the received packet is not the last packet of the information. We call this modified Whittle index policy as naïve Whittle index policy (NWIP).

We are interested in the expected average AoI, which is defined by (2.12). We evaluate the performance with different numbers of sources and different rates of information generation. In addition, we evaluate the benefit of coding in the presence of packet losses.

6.1 Comparison with the Optimal Solution

First, we implement the optimal solution Opt and use it as the benchmark for the above three methods. Nevertheless, due to the high complexity of Opt , such comparison is only possible for a small network over a small time period. Hence, we simulate the following simple networking scenario

- There are 2 sources and 2 corresponding destinations in the network.
- The information generation rate is 0.9 for each source. Actually, this value could be random, and the sources can have different rate from each other.
- For an arbitrary source, $r_i \in \{2, 3\}$. To diversify the traffic, we set $r_i \neq r_j$ if $i \neq j$.
- The total time length is 20 intervals, which means $K = 20$. Moreover, we set the number of timeslots of each interval as $T = 5$, in accordance with (2.10).

The simulation result is plotted in Fig. 6.1. This shows that our scheduling policy MWIP achieves a smaller $E AvgAoI$, compared with that of GP and NWIP and it is very close to the minimum $E AvgAoI$ achieved by Opt . In addition, the $E AvgAoI$ of GP and NWIP is the nearly identical. Recall the Whittle index in [14], when the information generation rate is the same for all sources, is only effected by AoI, which is the same as GP. What's more, in [14] the index calculation formula is monotonic, which means a higher AoI will always result in a bigger Whittle index. In other words, the source with a higher AoI will be scheduled for transmission first if NWIP is applied. This matches the key idea of GP. Hence, the GP and NWIP policies will make the consistent decision sequence with a high chance.

Table 6.1: The Hardware Environment

OS	Linux 64-bit
Processor	Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz
Memory	8 GB

Regarding the running speed, we run the simulation on a commodity Linux machine with parameters given in Table 6.1. NWIP returns the result in 0.3 ms; GP returns the result in 0.25 ms; MWIP returns the result in 0.3 ms. Opt , however, has

a much higher running time, which is 314572.8 ms. This is because the number of scheduling order combinations that meet the necessary condition (Chapter 3) is up to $(2!)^{20}$, and *opt* search as for the optimal value in the entire space.

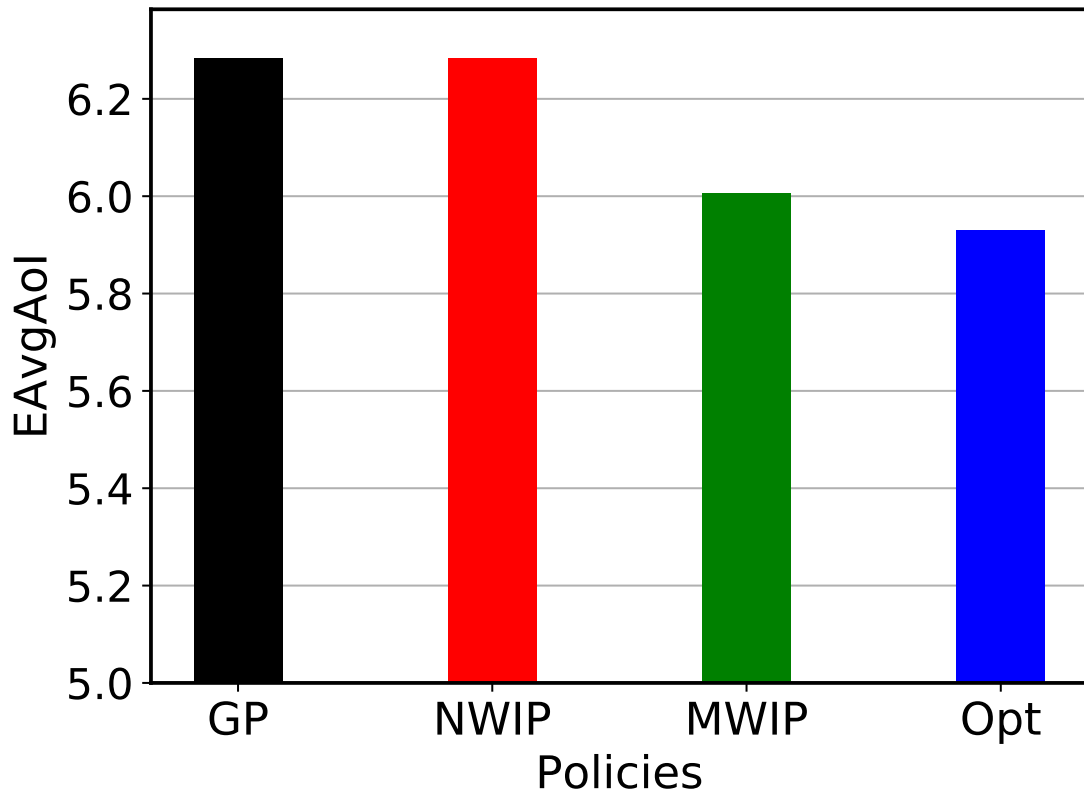


Figure 6.1: Benchmark with the *Opt* algorithm.

6.2 Impact of Number of Sources

Based on the system model described in Sec. 2.3.1, we simulate a network with multiple sources and longer observed time interval in this section. In particular, we assume that an arbitrary packet can be successfully delivered to the intended destination with a packet transmission probability high enough whenever it's scheduled for transmission. We set the parameters $K = 1000$ and $\omega = 0.5$.

For each source i , we assume that each information includes a random number of packets r_i , which is randomly selected from $\{1, 2, 3, 4, 5, 6\}$ (with replacement). The information generation rate of each source is set to 0.8. We use the *minimum* T that

satisfies (2.10) in our simulation, that is $T = \left\lceil 0.8 \times \sum_{i=1}^N r_i \right\rceil$. Note that this value of T changes with the number of sources. To obtain the average value of $EAvgAoI$, we run the simulation 100 times for all policies.

Table 6.2: Impact of Number of Sources on $EAvgAoI$

Policy	Source Amount									
	5	10	15	20	25	30	35	40	45	50
GP	27.28878	39.71936	67.68997	90.82541	111.09172	133.69518	163.09424	190.16638	203.67403	230.79568
NWIP	27.28878	39.79056	67.68603	90.66856	110.97294	133.62292	163.07634	190.00626	203.79690	230.87717
MWIP	27.28878	38.45890	60.89890	81.46848	101.56829	121.01118	147.64967	172.59384	187.68068	211.94173

The average simulation results of policies GP, NWIP, and MWIP are listed in Table 6.2. Obviously, MWIP outperforms the other two when the number of sources is big and observed time interval is long. NWIP and GP achieve very similar $EAvgAoI$, as explained in Sec. 6.1. By analyzing the result obtained under the same policy, we observe that a larger number of sources causes a higher $EAvgAoI$. This is because when the information generation rate of each source is fixed, a larger number of sources means a higher system workload, resulting in higher AoI.

We also collected all the numerical data obtained in simulation and draw a Box-and-Whisker plot in Fig. 6.2 to show the data distribution through their quartiles. Obviously, compared across the policies, MWIP has an advantage regarding the minimum, lower quartile, median, upper quartile and maximum of dataset, which is formed by the result obtained in each round of simulation. The maximum value from MWIP is even lower than the minimum values of GP and NWIP. This also explains why we get much smaller $EAvgAoI$ in Table 6.2 for MWIP.

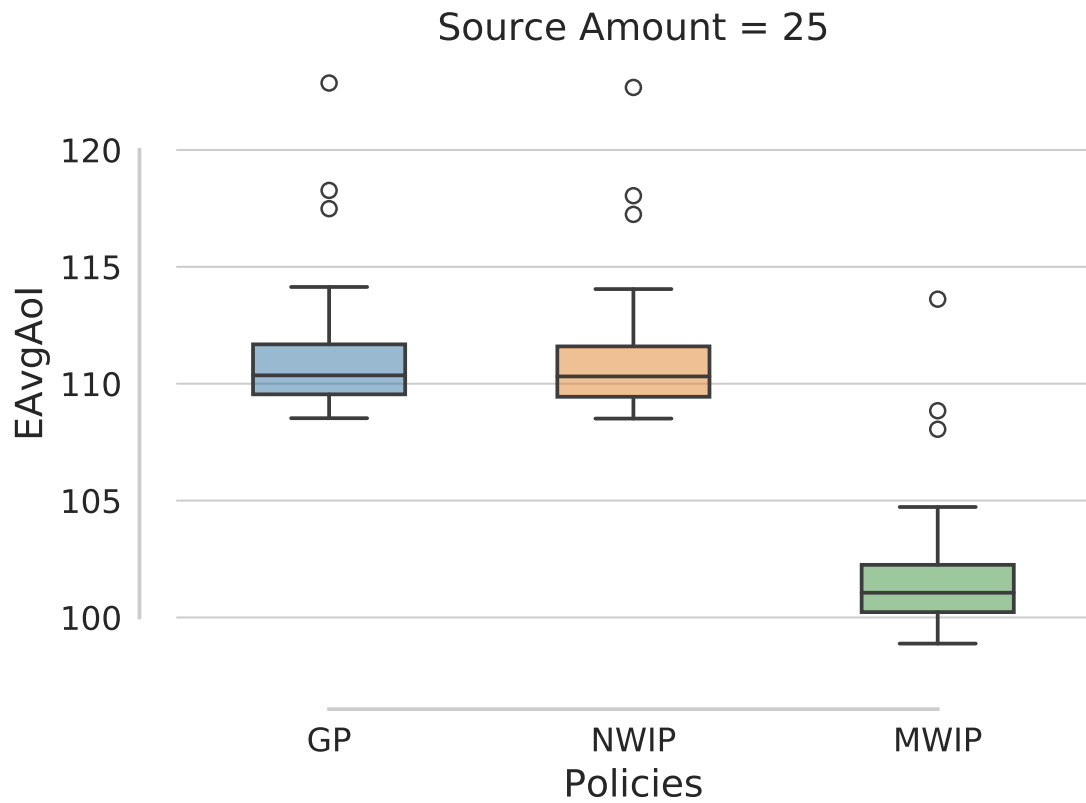


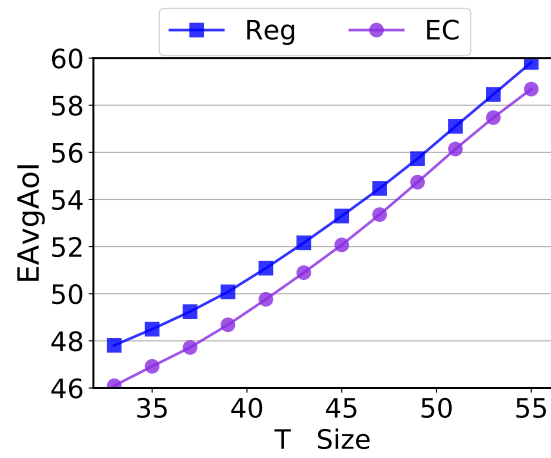
Figure 6.2: Box-and-Whisker plot depicting the distribution of numerical results. Each box contains five important values of the dataset, the minimum, lower quartile, median (second quartile), upper quartile and maximum of $EAvgAoI$, from bottom to top. The white points above the maximum line are outliers.

6.3 Benefit of Coding in Lossy Networks

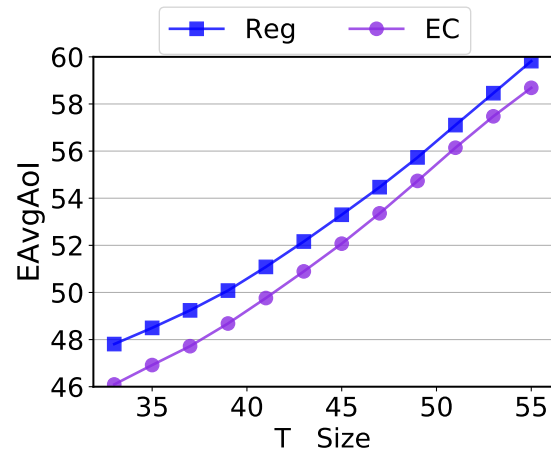
We simulate the scheduling and transmission process in a lossy network, where packets are transmitted without delivery guarantee. We want to explore the benefits of *EC Trans* strategy in lossy networks. By setting the bit error rate ($BER = 4 \times 10^{-4}$) and $X = 4000$, $Y = 400$, it's easy to calculate that the *packet delivery probability* for the two transmission strategies, *Reg Trans* and *EC Trans*, are $\lambda_1 = 0.64$ and $\lambda_2 = 0.67$, respectively. We consider a network with 10 sources.

Considering the uncertainty of channel status, we use (r_i, T) code for *EC Trans* transmission strategy, where T is the interval size obtained according to (2.10). We first investigate the effect of T on the *EAvgAoI*, by increasing T from *minimum T* to double this value. Note that the *minimum T* is obtained using the approach in Sec. 6.2. We also run the simulation for 100 times and calculate the average. Fig. 6.3 presents the simulation result by increasing the value of T gradually. *EC Trans* strategy achieves lower *EAvgAoI* for all policies, compared with that of *Reg Trans* strategy.

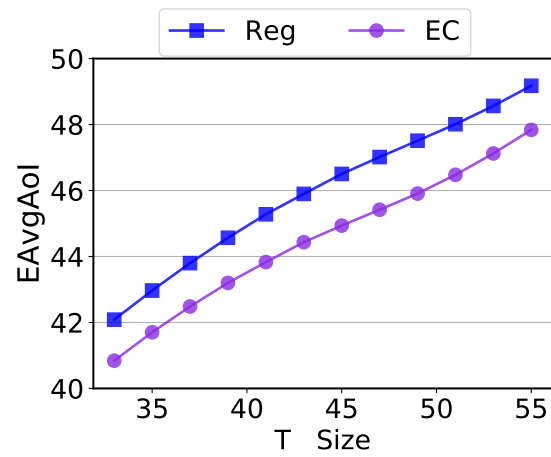
We also investigated the impact of information generation rate by changing it from 0.7 to 1.0. Following the information generation rate, the *minimum T* varies. By using the *minimum T* for each rate, we plot the average result in Fig. 6.4. We observe that low *EAvgAoI* comes with high information generation rate for all scheduling policies, since information is transmitted more frequently to update the knowledge at the destinations, resulting in a low AoI. Similarly, *EC Trans* strategy achieves a lower *EAvgAoI*. This illustrates the advantage of erasure code method in lossy networks. Overall, compared to GP and NWIP, MWIP achieves a lower *EAvgAoI* in all the tested scenarios.



(a) GP

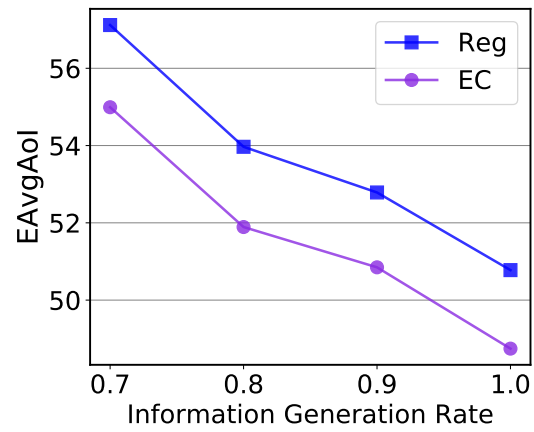


(b) NWIP

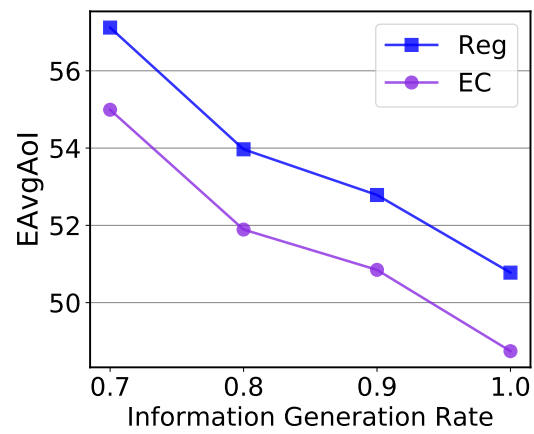


(c) MWIP

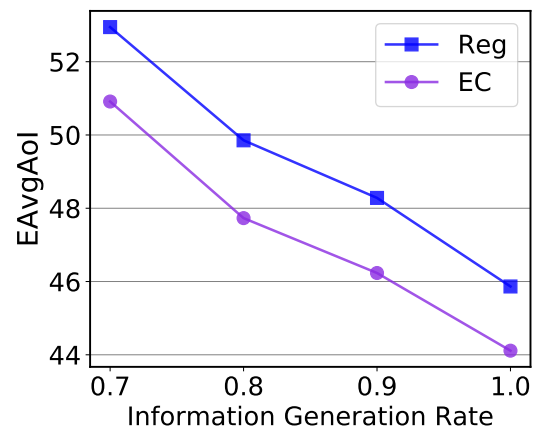
Figure 6.3: Analysis of T size: $K = 3000$, $p_i = 1.0$.



(a) GP



(b) NWIP



(c) MWIP

Figure 6.4: Analysis of information generation rate: $K = 3000$, $p_i \in \{0.7, 0.8, 0.9, 1.0\}$.

Chapter 7

Conclusions and Future Work

7.1 Summary

Age of information (AoI), new metric to capture the freshness of information, has attracted great attention in the last couple of years. In many real-world systems such as autonomous driving, timely information update is critical since outdated information not only wastes network bandwidth but also may lead to disastrous consequences. More often than not, a piece of information that is meaningful to the applications may be embedded in multiple packets, and as such we need to extend existing models from the one-packet-one-information case to a more general multiple-packet-one-information scenario. This thesis addressed this challenge. We first analyzed the necessary condition for optimal AoI scheduling, based on which we proposed an optimal solution. To address the high complexity of the optimal solution, we proposed a new way to track the age with approximation. The approximation not only offers an easy way to calculate AoI update, but also allows us to develop a Whittle index-based solution, called Multiple-packet Whittle Index Policy (MWIP), that runs much faster and achieves close to optimal results. For networks with unreliable transmission, we proposed a transmission strategy using erasure codes. With extensive simulation, we evaluated the performance of MWIP and compared it with two baseline methods, namely Greedy Policy (GP) and Naïve Whittle Index Policy (NWIP). In conclusion, MWIP reduces the average age in the general multiple-packet-one-information model in both reliable and lossy networks.

7.2 Future Work

It is interesting to study how to minimize the average AoI when the new status update information can be sampled at arbitrary time, rather than at the beginning of a time interval. In this case, it's possible that transmitting the newly generated information first by interrupting an on-going one may achieve a lower average AoI. Hence, it's necessary to take the information sampling time into consideration.

Other interesting directions for future work include: investigating the information arrival patterns, understanding the effect of channel status on successful packet transmission probability and scheduling transmission with priority.

Bibliography

- [1] Noor Hafizah Amer, Hairi Zamzuri, Khisbullah Hudha, and Zulkifli Abdul Kadir. Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges. *Journal of Intelligent & Robotic Systems*, 86(2):225–254, 2017.
- [2] Jumpei Arata, Hiroki Takahashi, Phongsaen Pitakwatchara, Shin’ichi Warisawa, Kazuo Tanoue, Kozo Konishi, Satoshi Ieiri, Shuji Shimizu, Naoki Nakashima, Koji Okamura, et al. A remote surgery experiment between japan and thailand over internet using a low latency codec system. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 953–959. IEEE, 2007.
- [3] Baran Tan Bacinoglu, Elif Tugce Ceran, and Elif Uysal-Biyikoglu. Age of information under energy replenishment constraints. In *2015 Information Theory and Applications Workshop (ITA)*, pages 25–31. IEEE, 2015.
- [4] Baran Tan Bacinoglu and Elif Uysal-Biyikoglu. Scheduling status updates to minimize age of information with an energy harvesting sensor. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1122–1126. IEEE, 2017.
- [5] Donald A Berry. A bernoulli two-armed bandit. *The Annals of Mathematical Statistics*, pages 871–897, 1972.
- [6] Maice Costa, Marian Codreanu, and Anthony Ephremides. Age of information with packet management. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 1583–1587. IEEE, 2014.

- [7] Maice Costa, Marian Codreanu, and Anthony Ephremides. On the age of information in status update systems with packet management. *IEEE Transactions on Information Theory*, 62(4):1897–1910, 2016.
- [8] Maice Costa, Stefan Valentin, and Anthony Ephremides. On the age of channel information for a finite-state markov model. In *2015 IEEE International Conference on Communications (ICC)*, pages 4101–4106. IEEE, 2015.
- [9] Maice Costa, Stefan Valentin, and Anthony Ephremides. On the age of channel state information for non-reciprocal wireless links. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 2356–2360. IEEE, 2015.
- [10] Alexandros G Dimakis, Vinod Prabhakaran, and Kannan Ramchandran. Decentralized erasure codes for distributed networked storage. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2809–2816, 2006.
- [11] Nils Gageik, Paul Benz, and Sergio Montenegro. Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. *IEEE Access*, 3:599–609, 2015.
- [12] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.
- [13] Qing He, Di Yuan, and Anthony Ephremides. Optimal link scheduling for age minimization in wireless systems. *IEEE Transactions on Information Theory*, 2017.
- [14] Yu-Pin Hsu. Age of information: Whittle index for scheduling stochastic arrivals. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2634–2638. IEEE, 2018.
- [15] Yu-Pin Hsu, Eytan Modiano, and Lingjie Duan. Age of information: Design and analysis of optimal scheduling algorithms. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 561–565. IEEE, 2017.
- [16] Longbo Huang and Eytan Modiano. Optimizing age-of-information in a multi-class queueing system. *arXiv preprint arXiv:1504.05103*, 2015.
- [17] Igor Kadota, Abhishek Sinha, and Eytan Modiano. Optimizing age of information in wireless networks with throughput constraints. In *IEEE INFOCOM*

- 2018-IEEE Conference on Computer Communications*, pages 1844–1852. IEEE, 2018.
- [18] Igor Kadota, Elif Uysal-Biyikoglu, Rahul Singh, and Eytan Modiano. Minimizing the age of information in broadcast wireless networks. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 844–851. IEEE, 2016.
- [19] Clement Kam, Sastry Kompella, Gam D Nguyen, Jeffrey E Wieselthier, and Anthony Ephremides. On the age of information with packet deadlines. *IEEE Transactions on Information Theory*, 2018.
- [20] Alexandros Karakasidis, Panos Vassiliadis, and Evaggelia Pitoura. Etl queues for active data warehousing. In *Proceedings of the 2nd international workshop on Information quality in information systems*, pages 28–39. ACM, 2005.
- [21] Sanjit Kaul, Marco Gruteser, Vinuth Rai, and John Kenney. Minimizing age of information in vehicular networks. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 350–358. IEEE, 2011.
- [22] Sanjit Kaul, Roy Yates, and Marco Gruteser. On piggybacking in vehicular networks. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pages 1–5. IEEE, 2011.
- [23] Sanjit Kaul, Roy Yates, and Marco Gruteser. Real-time status: How often should one update? In *INFOCOM, 2012 Proceedings IEEE*, pages 2731–2735. IEEE, 2012.
- [24] Antzela Kosta, Nikolaos Pappas, Vangelis Angelakis, et al. Age of information: A new concept, metric, and tool. *Foundations and Trends® in Networking*, 12(3):162–259, 2017.
- [25] Hsiao-Ying Lin and Wen-Guey Tzeng. A secure erasure code-based cloud storage system with secure data forwarding. *IEEE transactions on parallel and distributed systems*, 23(6):995–1003, 2011.
- [26] Sameer Mittal, Muztoba Ahmad Khan, David Romero, and Thorsten Wuest. Smart manufacturing: characteristics, technologies and enabling factors. *Pro-*

- ceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 233(5):1342–1361, 2019.
- [27] Christos H Papadimitriou and John N Tsitsiklis. The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999.
- [28] Martin L Puterman. Markov decision processes: Discrete stochastic dynamic programming. *Journal of the Operational Research Society*, 46(6):792–792, 1995.
- [29] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.
- [30] Saber Taghvaeeyan and Rajesh Rajamani. Portable roadside sensors for vehicle counting, classification, and speed measurement. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):73–83, 2014.
- [31] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.
- [32] Peter Whittle. Multi-armed bandits and the gittins index. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):143–149, 1980.
- [33] Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A):287–298, 1988.
- [34] Stephen B Wicker and Vijay K Bhargava. *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.
- [35] Roy D Yates. Lazy is timely: Status updates by an energy harvesting source. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 3008–3012. IEEE, 2015.
- [36] Bo Zhou and Walid Saad. Minimizing age of information in the internet of things with non-uniform status packet sizes. In *submitted to Proc. of IEEE International Conference on Communications (ICC)*, 2019.