

Design of Centralized Sampling and Scheduling Policies for Optimizing Information Freshness in Multi-Hop Wireless Networks with Interference

A thesis submitted
in partial fulfillment for the award of the degree of

Doctor of Philosophy

by

Nibin Raj



Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram, India

June 2026

Certificate

This is to certify that the thesis titled *Design of Centralized Sampling and Scheduling Policies for Optimizing Information Freshness in Multi-Hop Wireless Networks with Interference* submitted by **Nibin Raj**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfillment for the award of the degree of **Doctor of Philosophy** is a bona fide record of the original work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Vineeth B.S.
Associate Professor

Dr. Basudeb Ghosh
Professor and Head

Place: Thiruvananthapuram

Date: June 2026

Declaration

I declare that this thesis titled *Design of Centralized Sampling and Scheduling Policies for Optimizing Information Freshness in Multi-Hop Wireless Networks with Interference* submitted in partial fulfillment for the award of the degree of **Doctor of Philosophy** is a record of the original work carried out by me under the supervision of **Dr. Vineeth B.S.**, and has not formed the basis for the award of any degree, diploma, associateship, fellowship, or other titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

Place: Thiruvananthapuram

Date: June 2026

Nibin Raj
(SC19D038)

This thesis is dedicated to my family.

Acknowledgements

As I look back on the journey of completing this Ph.D. thesis, I am filled with deep gratitude for the people and circumstances that have shaped and supported me along the way.

I extend my sincere thanks to my supervisor, Dr. Vineeth B.S., Associate Professor, Department of Avionics, Indian Institute of Space Science and Technology (IIST), for his continuous guidance, encouragement, and unwavering support throughout this research. His generous advice, insightful feedback, and careful review of my work have been invaluable. I have benefited immensely, both professionally and personally, from being his student.

I would also like to express my sincere gratitude to my Doctoral Committee members, Dr. Basudeb Ghosh, Dr. N. Selvaganesan, Dr. B.S. Manoj, Dr. Deepak Mishra, Dr. Sooraj Ravindran, and Dr. Sumitra S., for their valuable feedback during each annual DC review. Their insightful comments and suggestions have greatly contributed to improving the quality and direction of this thesis. I am also deeply grateful to the following faculty members for their support and guidance: Dr. M. Vanidevi, Dr. Chris Prema S, Dr. J. Sheeba Rani, Dr. Anoop C. S, Dr. R. Lakshmi Narayanan, and Dr. Rajeevan P. P.

Additionally, I would like to express my heartfelt thanks to the IIST for providing an excellent research environment that played a pivotal role in the successful completion of this work. My sincere appreciation also extends to all the faculty and staff members of the Department of Avionics for their direct and indirect support throughout my doctoral journey.

It has been an absolute honour to work in the Communication Networks Lab (ComNet) at IIST, along with my labmate and colleague, Minha Mubarak, whose insightful technical discussions and constant companionship made this period both productive and memorable. I would also like to fondly acknowledge my friends at IIST: Sarath, Vishak, Sam, Arun, Amal, Francis, Sanjay, Abhijith, Geethu, Babitha, and Apoorva, whose friendship, support, and the many memorable moments we shared have made this journey more meaningful.

I am deeply thankful for the financial support received initially through the DST-SERB Early Career Research (ECR) Grant (ECR/2018/001649) and subsequently from the IIT Palakkad Technology IHub Foundation (IPTIF) in the form of the IPTIF Doctoral Fellowship (IPTIF/HRD/DF/022).

Most importantly, I extend my deepest gratitude to my family, including my wife, Dr. Chandana, my mother, my brother, and my in-laws, for their constant understanding, encouragement, and support throughout this entire Ph.D. journey. Their endless sacrifices, patience, and unwavering belief in me have been the pillars of my strength.

Finally, I humbly express my gratitude to God Almighty for His guidance and blessings, which have illuminated my path and provided me with strength throughout this journey of life.

Nibin Raj

Abstract

Emerging applications such as remote estimation and control in industrial automation, cyber-physical systems (CPS), and Internet of Things (IoT) are enabled by next generation wireless networks. In such networks, spatially distributed source nodes observe and sample physical processes which contain the desired information for such applications. The samples are packetized and transmitted to destination nodes. In many scenarios, due to transmission energy or deployment constraints, the sources are not able to transmit the packets directly to destination nodes. Rather the updates are forwarded through intermediate relay nodes before reaching a gateway or a destination resulting in multi-hop transmission. Such applications require freshness of source information at destination nodes rather than high throughput or low latency. Although, information freshness and its optimization has been extensively studied in single-hop settings, there has been limited study of information freshness for multi-hop wireless networks with interference and packet losses. Therefore, designing source sampling and multi-hop transmission policies to ensure information freshness in the presence of interference, packet losses, and multi-hop routing constraints is a fundamental network design challenge which we address in this thesis.

Information freshness is quantified using metrics such as Age of Information (AoI) and Age of Incorrect Information (AoII). AoI measures the timeliness of information by quantifying the time elapsed since the generation of the most recently received update, whereas AoII additionally captures the correctness of the receiver's estimate of the underlying process state. AoII is particularly relevant for remote monitoring and control applications, where incorrect information can lead to degraded system performance or erroneous decisions. The network model considered for formulating the policy design problem is motivated by practical industrial communication standards such as IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH). A time slotted centrally scheduled multi-hop wireless network model is considered for AoI or AoII optimization. Interference in link transmissions is modelled using K -hop or general graph based interference models. Packet losses are modelled as being random across different links. This thesis develops analytically motivated scheduling algorithms for minimizing the average AoI (AAoI) and average AoII (AAoII) in such networks. The developed scheduling algorithms use limited information about the network available from packets received at the central scheduler node.

The thesis first addresses AAoI minimization in multi-hop networks with interference and unreliable links for centralized schedulers. A centralized scheduler is required to determine source sampling and link activation times using limited information about the current state of the network which is available at the scheduler. A direct Markov decision process (MDP) formulation for obtaining scheduling policies is shown to be computationally intractable due to the large state space. To overcome this challenge, the problem is reformulated as a Restless Multi-Armed Bandit (RMAB) under a restriction on the policy structure used for deriving priority indices. This novel reformulation enables the derivation of analytically characterized priority indices, leading to low-complexity index-based sampling and link scheduling policies. Simulation results demonstrate substantial performance improvements over existing AoI-aware policies, particularly in large networks and under unreliable communication links. The focus is then extended to AoII minimization in multi-hop networks, where again the centralized scheduler has access to limited information. The resulting problem is modeled as a partially observable Markov decision process (POMDP). By developing tractable RMAB approximations and establishing indexability properties, novel Whittle index-based policies are derived for AoII minimization. In simulations, these policies are observed to achieve significant AAoII reductions across a wide range of network topologies and interference scenarios.

Motivated by the deterministic communication requirements of industrial wireless networks, the thesis further investigates periodic deterministic sampling and scheduling policies for information freshness optimization in multi-hop networks. A periodic policy does not require global information about the network and thus is amenable for practical implementation. For any periodic policy with a given period, an explicit analytical characterization of AAoI is obtained as a function of the period and source-to-sink delivery delays. We then formulate an integer programming problem to obtain the optimal period and delivery delays for minimizing the AAoI objective, subject to constraints, e.g. interference constraints on link activations. The integer program is relaxed to a linear program to obtain numerical lower bounds. In addition, analytical lower bounds on AAoI are derived for tree networks with K -hop interference constraints. The resulting periodic policies are shown to achieve near-optimal performance by comparison with lower bound as well as the solution of the integer programming problem for small networks. We then consider AoII, for which an analytical characterization of AAoII under periodic deterministic policies is developed in terms of the policy period and packet delivery delays. This characterization enables the derivation of fundamental lower bounds on AAoII for any periodic deterministic policy, together with analytical upper bounds based on deterministic scheduling strategies. Numerical

cal results demonstrate the near-optimality of the proposed policies and show that periodic deterministic scheduling can outperform non-periodic policies under complete interference and K -hop interference model.

Overall, this thesis develops sampling and scheduling algorithms for optimizing information freshness in multi-hop wireless systems. This work advances the design of multi-hop networks for reliable and timely remote estimation, thereby ensuring information freshness in next-generation industrial and cyber-physical systems.

Contents

List of Tables	xvii
Abbreviations	xix
1 Introduction	1
1.1 Background	3
1.2 Literature Survey	7
1.3 Thesis Objectives, Outline, and Contributions	11
2 Priority-Index Based Sampling and Scheduling Policies for Minimization of Average Age of Information in Time-Slotted Multi-hop Networks	17
2.1 Introduction and Prior Work	17
2.2 System Model and Problem Statement	24
2.3 RMAB Formulation and a Priority-Index Policy	28
2.4 Extension to K-hop Interference Model	32
2.5 Simulations and Performance Analysis	33
2.6 Other extensions	36
2.7 Conclusions	44
Appendix	46
2.A Proof of Proposition 2.1	46
3 Priority-Index Based Sampling and Scheduling Policies for Minimization of Average Age of Incorrect Information in Time-Slotted Multi-hop Networks	49
3.1 Introduction and Prior Work	49
3.2 System Model and Problem Statement	57
3.3 RMAB Formulations and Whittle Index based Genie Policies	60
3.4 Implementable Centralized Sampling and Scheduling Policies	69

3.5	Implementable Policies for Networks with K-Hop and General Interference Models	71
3.6	Simulations and Performance Analysis	72
3.7	Other extensions	81
3.8	Conclusions	83
	Appendix	85
3.A	Discussion: Justification for Threshold Policies	85
3.B	Proof of Proposition 3.1	87
3.C	RMAB Formulation for Π_{C-G-1P}	90
3.D	Proof of Proposition 3.4	92
3.E	Alternative Implementation of π_H	94
4	Periodic Deterministic Policies for Minimizing Average Age of Information in Time-Slotted Multi-Hop Networks	95
4.1	Introduction and Prior Work	95
4.2	System Model and Problem Statement	100
4.3	Minimization of Average AoI and Lower Bounds for Periodic Deterministic Policies	103
4.4	Periodic Deterministic Sampling and Scheduling Policies for Multihop Networks	114
4.5	Simulations and Performance Analysis	125
4.6	Extensions	131
4.7	Conclusions	139
	Appendix	141
4.A	Computational Difficulties Associated with ILP	141
4.B	Proof of Proposition 4.6	142
4.C	Proof of Proposition 4.8	143
4.D	Proof of Proposition 4.9	144
5	Periodic Deterministic Policies for Minimizing Average Age of Incorrect Information in Time-Slotted Multi-Hop Networks	145
5.1	Introduction and Prior Work	145
5.2	System Model and Problem Statement	149
5.3	Analytical Characterization and Bounds	152
5.4	Simulations and Performance Comparisons	156
5.5	Discussion: Extension to the case of lossy links	160

5.6	Conclusions	163
	Appendix	164
5.A	Proof of Proposition 5.1	164
5.B	Proof of Proposition 5.2	165
5.C	Proof of Proposition 5.3	167
6	Summary and Scope for Future Work	173
	Bibliography	176
	List of Publications	191

List of Tables

1.1	Summary and comparison of the information-freshness studies presented in this thesis.	10
2.1	Overview of sampling and scheduling policies reported in prior work. . . .	19
2.2	List of symbols and their descriptions used in Chapter 2	23
2.3	Evolution of $S^s(t)$, $A^s(t)$, and $H^s(t)$ from t to $t + 1$ as a function of $D^s(t)$ for source s (note: w.p. is for “with probability”).	29
3.1	Sampling and scheduling policies in this chapter and comparisons with prior work. The last three policies are implementable at the central node. . .	53
3.2	List of symbols and their descriptions used in Chapter 3	55
3.3	A summary of the main assumptions used in this chapter	58
3.4	Evolution of $(A^s(t), S^s(t), H^s(t))$ from t to $t + 1$ for any source s based on $D^s(t)$ (in columns). Different cases based on the current values of $A^s(t)$, $S^s(t)$, and $H^s(t)$ are shown.	62
3.5	Evolution of the embedded AoII A_{m+1} from A_m when scheduling decision $D_m \in \{0, 1\}$ for SMDP. The single stage cost $c(A_m, D_m)$ of the SMDP is the sum of the single stage cumulative AoII cost (2nd column) and λD_m . . .	85
3.6	Evolution of $A^s(t)$, $S^s(t)$, and $H^s(t)$ from time t to $t + 1$ for policies in Π_{C-G-1P} under Case I: $X^s(t) \neq X^s(t - 1)$ and Case II: $X^s(t) = X^s(t - 1)$. . .	90
4.1	List of symbols and their descriptions used in Chapter 4	98
5.1	List of symbols and their descriptions used in Chapter 5	148
5.2	Evolution of the embedded AoII A_n^s to A_{n+1}^s	154
5.3	Evolution of the embedded AoII A_n^s to A_{n+1}^s for a network with link losses .	162

Abbreviations

CPS	Cyber-Physical System
AoI	Age of Information
AAoI	Average Age of Information
AoII	Age of Incorrect Information
AAoII	Average Age of Incorrect Information
RMAB	Restless Multi-armed Bandit
MDP	Markov Decision Process
SMDP	Semi-Markov Decision Process
POMDP	Partially Observable Markov Decision Process
MC	Markov Chain
L-MDP	Lagrange-MDP
SS-MDP	SingleSource-MDP
RRT	Renewal Reward Theorem
MRRT	Markov Renewal Reward Theorem
TSCH	Time-Slotted Channel Hopping
PeDe	Periodic Delivery condition
ILP	Integer (binary) Linear Programming
SS-Line	Sampling and Scheduling algorithm for Line networks
SS-Tree	Sampling and Scheduling algorithm for Tree networks
CS	Continue-Service
IID	Independent and Identically Distributed

Chapter 1

Introduction

Smart cities, automated agriculture, industrial process and control, cyber-physical systems, and Internet of Things rely on wireless communication networks to obtain fresh status updates for accurate remote estimation, monitoring, and control [1]. In these networks, spatially distributed sensors observe physical processes and generate packetized status updates. Due to transmission power limitations, coverage requirements, and the large geographical areas encountered in industrial deployments, these updates often cannot be delivered directly to a sink node. Instead, they are forwarded through intermediate relay nodes, resulting in a multi-hop wireless network. For the example applications discussed, it is not sufficient merely to receive data packets; the received information must also be sufficiently fresh to accurately represent the current state of the monitored processes. Information freshness characterizes the timeliness and accuracy of the information available at the receiver and depends on update delivery delays and rate of delivery of successful packet receptions. Traditional communication metrics such as throughput and delay are inadequate for quantifying the freshness of information available at the sink. Designing communication networks for ensuring information freshness is therefore a fundamental challenge, particularly in the presence of wireless interference and packet losses. Although information freshness has been studied extensively in single-hop scenarios, its study for multi-hop network scenarios remains limited. This thesis aims to fill this gap by considering the design of sampling and scheduling policies for multi-hop networks in the presence of interference and noise.

This thesis focuses on two information freshness metrics: AoI [2] and AoII [3]. AoI measures the time elapsed since the generation of the most recently received update and provides a direct measure of information timeliness. However, AoI does not account for whether the received information accurately reflects the current state of the underlying process. AoII extends AoI by incorporating estimation correctness and quantifies the mismatch

between the actual process state and its estimate at the receiver. This metric is particularly relevant for remote monitoring and control applications, where incorrect information may lead to degraded system performance or erroneous control actions.

The network models that we consider are motivated by industrial wireless communication standards such as IEEE 802.15.4e TSCH [4] and WirelessHART [5], which employ time-slotted communication and centralized network management. In these systems, a centralized controller determines the sampling times of source nodes and the activation schedules of wireless links using available network information. Furthermore, the network is subject to unreliable wireless links and interference constraints. This thesis investigates the design of centralized sampling and scheduling policies for optimizing information freshness in time-slotted multi-hop wireless network models with a tree topology. Interference is modelled using graph-based interference models, including complete interference and K -hop interference models. Link reliability is modelled as being random. The objective is to develop analytically motivated centralized sampling and scheduling policies for such models. The policies are also designed to work with limited information about the network which is available at the scheduler through received packets, rather than relying on global information about network and packets within the network.

In this thesis, both periodic and non-periodic scheduling policy design is considered to overcome the challenges associated with optimization information freshness. First, the average AoI minimization problem is investigated under interference constraints and unreliable wireless links. A direct formulation of the policy design problem for AoI minimization leads to a Markov decision process (MDP). To overcome the computational complexity of solving this MDP, the problem is reformulated as a RMAB, leading to low-complexity index-based scheduling policies. Subsequently, the focus is extended to AoII, where the centralized controller has only partial knowledge of the network state. This problem is modeled using a POMDP, and tractable RMAB approximations are developed to derive practical scheduling policies. Motivated by the deterministic communication requirements of industrial wireless networks, the thesis further investigates periodic deterministic sampling and scheduling policies for both AoI and AoII optimization. Analytical performance characterizations, lower bounds, and near-optimal scheduling algorithms are developed for these settings. Together, these contributions provide a comprehensive framework for optimizing information freshness in multi-hop wireless networks.

1.1 Background

1.1.1 Information Freshness

Age of Information (AoI): The exchange of fresh information over multi-hop wireless networks is becoming increasingly important with the rise of ad hoc wireless systems such as the IoT [6], vehicular networks [7], and networks of unmanned aerial vehicles [8]. Traditional network performance metrics, such as delay or throughput, quantify the efficiency of data transfer but do not directly capture the freshness or correctness of information at the receiver [9]. For the scenarios discussed above, it is often more important to know how up-to-date the information is. AoI is a metric used to measure how up-to-date the information at a receiver is [10], [11]. AoI increases linearly when no update is received. When a new update is received, the AoI reduces to the time that has passed since that update was generated, and then increases linearly until the next update arrives. An illustration of this evolution is shown in Figure 1.1. Suppose t denotes time (we consider discrete time systems in this thesis, so that $t \in \mathbb{Z}_+ = \{0, 1, 2, \dots\}$). At time $t \geq t_g$, an update packet with timestamp t_g (the packet generation time) has age $t - t_g$. Let $t_g(t)$ denote the timestamp of the most recent update received by the destination at time t ; then the age of information is defined as the stochastic process

$$\Delta(t) = t - t_g(t). \quad (1.1)$$

Age of Incorrect Information (AoII): Although AoI captures timeliness, it does not directly measure whether the receiver's estimate reflects the true state of the source. In particular, minimizing AoI does not necessarily minimize estimation error for Markovian sources [12]. To address this limitation, alternative metrics such as Age of Synchronization (AoS) [13] and effective age [14] have been proposed. However, these approaches fail to capture the information structure of the source and the current estimate at the receiver, which is essential for remote estimation applications. To address this, a new metric AoII was introduced in [3], which measures the freshness of information while accounting for the content acquired from the transmitter. AoII is a semantic metric that reflects the correctness of the transmitted information beyond traditional performance measures such as delay and throughput [15].

For defining AoII, the source process of a source s is modeled as a stochastic process $X^s(t)$, and the transmitter sends status updates to the receiver to keep it informed of the

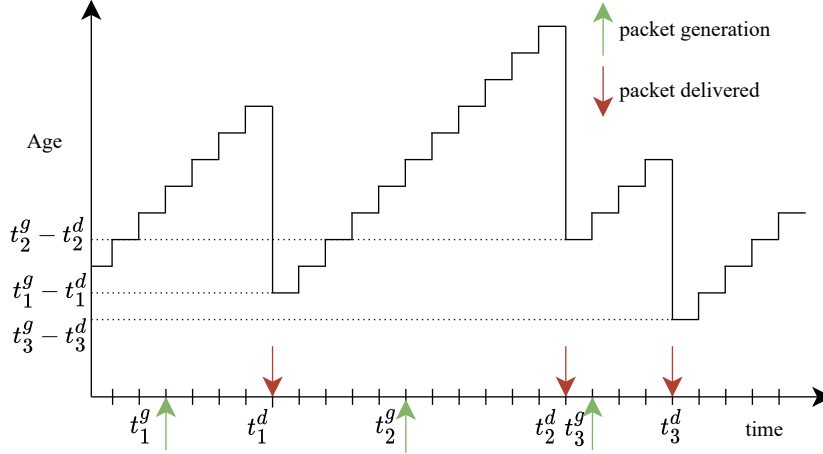


Figure 1.1: Illustration of AoI evolution. Packet generation times are indicated by green upward arrows, and packet deliveries to the destination are shown by red downward arrows. When the first packet is received, the AoI drops to $t_1^d - t_1^g$, where t_1^g and t_1^d denote the generation and delivery times of packet 1, respectively. Similar transitions are shown for packets 2 and 3, with AoI drops of $t_2^d - t_2^g$ and $t_3^d - t_3^g$.

current state of source s is $X^s(t)$. Due to transmission rate constraints, the transmitter may remain idle at times, i.e., it does not transmit updates continuously. While idle, the monitor retains the last received state, which may differ from the current state of the process $X^s(t)$. This retained value is denoted as the estimated state $\hat{X}^s(t)$ at the monitor. When $\hat{X}^s(t) \neq X^s(t)$, the monitor is in an undesirable state, which incurs a penalty. AoII metric [3] quantifies this penalty.

For instance, we model each source as a piecewise constant process [16], where the state of the monitored process at time t for source s is denoted by $X^s(t) \in \mathbb{R}$. The evolution of $X^s(t)$ is defined as follows. At the beginning of time slot t , with probability p_c^s , the process transitions to a new state sampled independently from a probability density function $f_X^s(x)$ (e.g., a uniform distribution $U(-m_s, m_s)$, with $m_s > 0$); otherwise, with probability $q_c^s = 1 - p_c^s$, the process remains in its current state. Mathematically, the process evolution is given by:

$$X^s(t+1) = \begin{cases} X^s(t), & \text{w.p. } q_c^s, \\ \sim f_X^s(x), & \text{w.p. } p_c^s. \end{cases}$$

At time slot t , if the scheduler selects source s , the source node samples the process $X^s(t)$ and generates a packet containing the value $X^s(t)$. This packet is then forwarded

where information from multiple sources is aggregated and delivered to a common destination. For instance, in intelligent transportation systems [22, 23], vehicles share information about traffic congestion and road conditions to avoid collisions and reduce congestion. Multi-hop networks provide several important advantages:

1. **Energy efficiency:** Nodes transmit over shorter distances, reducing transmission power requirements.
2. **Scalability:** The network can support a larger number of nodes distributed over wide geographic areas.
3. **Coverage:** Nodes beyond the direct communication range of the sink can participate through intermediate relays.

Despite these advantages, multi-hop networks introduce additional challenges. Packets must traverse multiple hops, which increases end-to-end delay. Scheduling and interference management also become more complex. Consequently, designing scheduling policies and analyzing multi-hop networks with the goal of optimizing freshness metrics such as AoI and AoII is a challenging but important task. The literature on AoI and AoII optimization for scheduling in multi-hop networks has received relatively limited attention, which is the focus of this thesis.

1.1.3 Industrial IoT Networks: Modelling Assumptions

Industrial Internet of Things (IIoT) is expected to significantly increase productivity and enable new applications. Wireless connectivity is a crucial component of IIoT networks. Although there exist wireless sensor networks, they are typically designed for requirements different from those of industrial systems. This has generated significant research interest in developing wireless communication systems that can match the performance of industrial wired networks, particularly in terms of reliability, deterministic latency, and IP interoperability. Efforts to address these challenges have been undertaken by standardization bodies such as the Internet Engineering Task Force (IETF). In particular, the Deterministic Networking (DetNet) working group was formed with the primary goal of supporting high reliability in packet delivery and providing deterministic worst-case bounds on latency [24], [25]. The IEEE 802.15 working group, established in 2003, developed technologies such as ZigBee [26] and WirelessHART [5]. IEEE 802.15.4 defined the physical and data link layers of the protocol stack, supporting low-rate, low-energy communications. However, it lacked the determinism and reliability required for industrial applications, which

demand high reliability, robustness, low latency, and scalability. To address this, IEEE 802.15.4e was introduced in 2012 as an amendment to enhance the MAC layer and meet industrial requirements [27], [28]. A key feature of the IEEE 802.15.4e amendment is the introduction of Time Slotted Channel Hopping (TSCH) communication, which effectively mitigates multi-path fading and interference on a given frequency. While the standard specifies TSCH operation, the design of TSCH schedules is left open, as it depends on application requirements and network topology.

In practical industrial wireless networks, communication is often organized in a time-slotted manner with centrally coordinated scheduling. Time-slotted deterministic scheduling is particularly relevant in such networks due to strict requirements on reliability, latency, jitter, and power [27], [28]. Deterministic scheduling is supported by industrial standards such as WirelessHART [5], ISA100.11a [29], and IEEE 802.15.4e TSCH [4]. TSCH, widely adopted in industrial and low-power wireless sensor networks [30], employs deterministic slot-based schedules and channel hopping to enhance reliability and mitigate interference. Moreover, it implements periodic scheduling policies with a period called a slotframe [4], which is a contiguous set of slots that repeats over time.

In this thesis, we consider time-slotted multi-hop networks, motivated by [4], in which a centralized controller at the sink node determines sampling and transmission decisions subject to interference constraints. This structure reflects real-world deployments and enables the systematic design of optimized scheduling policies to minimize AoI and AoII. The main assumptions considered in this thesis are as follows. The network is modeled as a directed tree network. We consider a convergecast communication setting, in which the sink node collects information packets from all source nodes in the network. Time evolves in discrete slots, indexed by positive integers. Each generated packet contains the monitored process value (or the state corresponding to the source) along with a timestamp indicating its generation time. Transmissions over the links are assumed to occur on the same communication channel, which is equipped with a single half-duplex radio interface. We further assume that packet transmission over each link is successful with probability p_s . Additional assumptions specific to each problem setting are presented in the corresponding chapters.

1.2 Literature Survey

AoI is a metric used to quantify the freshness of information available at a receiver [10, 11]. Unlike packet delay, AoI measures the staleness of the received information. This makes

it particularly suitable for applications that require timely status updates. Over the past decade, there has been a rapidly growing body of work analyzing AoI in queuing systems [10, 11, 31, 32]. AoI has been extensively studied in scheduling problems for single-hop networks due to their analytical tractability [33, 34]. Related research has also considered modeling and optimizing the freshness of query responses from data warehouses [35], designing network architectures to limit cache staleness [36], and implementing distributed QoS routing using aged or imprecise network state information [37]. In a single-hop setting, each source communicates directly with the sink, and typically only one node transmits in a given time slot to avoid interference [38], [39], [40], [41]. Prior work [42] considered stationary randomized policies for minimizing the average AoI in multi-hop networks with multiple source-destination pairs under general interference conditions, assuming transmissions occur over interference-free multi-hop links. In [43], the problem of minimizing the age of a single flow over interference-free multi-hop networks was studied. A ring topology is considered in [44], where all nodes act as both sources of update packets and remote estimators, and lower bounds on the peak and average AoI are obtained. A two-hop energy harvesting network, aiming to minimize AoI with fixed transmission delays, is considered in [45]. Liu et al. [46] addressed the problem of minimizing AoI while satisfying throughput requirements. Farazi et al. [47] investigated AoI minimization in a wireless network using a minimum connected dominating set (MCDS) for all-to-all broadcast flows, considering multiple sources, multi-hop communication, and time-slotted scheduling. Bastopcu [48] characterized the average freshness in a two-hop caching model as a function of content and cache update rates, while age-optimal cache update policies for two-hop networks were explored in [49]. In status update systems, besides AoI, other performance metrics of information freshness include AoII [3], Version AoI (VAoI) [50], and Age of Actuation (AoA) [51], with each metric capturing different aspects of timeliness and the significance of the information.

Although AoI captures timeliness, it does not directly measure whether the receiver's estimate accurately reflects the true state of the source. In particular, minimizing AoI does not necessarily minimize estimation error for Markovian sources [12]. To address this limitation, alternative metrics such as Age of Synchronization (AoS) [13] and effective age [14] have been proposed. However, these approaches do not fully capture the information structure of the source and the current estimate at the receiver, which are essential in remote estimation applications. To address this issue, the AoII metric was introduced in [3], which measures information freshness while accounting for the correctness of the information available at the receiver. AoII is a semantic metric that reflects the correctness

of the transmitted information beyond traditional performance measures such as delay and throughput [15]. Prior work on AoII in single-hop point-to-point networks has explored various scheduling scenarios. In [52], [53], optimal scheduling policies were developed to minimize AoII in the remote estimation of symmetric binary Markov sources over links with fixed delay. The work in [54] extended this to unknown Markov source models, designing scheduling policies to minimize AoII without full source knowledge. A point-to-point link with random delay was studied in [55], where scheduling policies were again aimed at minimizing AoII. Finally, [56] considered the problem of minimizing the average AoII (AAoII) under power constraints, highlighting the impact of energy limitations on scheduling design. Let us now briefly review some prior work on AoII in single-hop networks. In [57], a decentralized policy was designed to minimize AAoII in a slotted ALOHA network. AoII was characterized for a slotted ALOHA scenario without feedback, where the source processes are modeled as two-state Markov chains, in [58]. The work in [59] considered the design of a distributed CSMA protocol that minimizes AoI in a single-hop network and extended it to the AoII metric for source processes modeled as independent symmetric two-state Markov chains. Finally, [60] studied real-time tracking of a partially observable Markov source in an energy-harvesting system with communication constraints in a single-hop network.

Distributed scheduling for AoI minimization in single-hop networks, utilizing distributed convex optimization and CSMA-based protocols, is studied in [61], [59]. Reinforcement learning (RL)-based approaches for minimizing AoI and generalized freshness in single-hop networks have been studied in [62]. However, RL-based approaches for multi-hop networks remain largely unexplored. In [9], the authors study AoI minimization in multi-hop networks and consider three scheduling policies: stationary randomized, Age-Difference, and Age-Debt. AAoI minimization using RMAB formulations has been studied for single-hop networks in [63], [64], [65], and for single-hop broadcast networks in [66]. However, this approach has not been explored for multi-hop networks. To the best of our knowledge, there is no prior work on AoII minimization in multi-hop networks.

Recent studies have also extended information-freshness optimization beyond the conventional assumption that the performance penalty is a monotonically non-decreasing function of AoI. Shisher et al. showed that, in real-time supervised learning, inference error can be a non-monotonic function of AoI and developed Gittins- and Whittle-index-based scheduling policies for minimizing general AoI-dependent inference errors, including non-monotonic functions [67]. Similarly, Ari et al. considered goal-oriented remote inference with two-way delay and memory and developed an index-based threshold pol-

icy that accommodates both monotonic and non-monotonic age-dependent inference-error functions [68]. Related generalized age penalties have also been studied through AoII, which captures application-dependent aging characteristics and has motivated Whittle-index-based scheduling approaches [69]. In addition, Salimnejad et al. investigated age-dependent performance in a two-user multiple-access setting, further illustrating the need to consider generalized age-related performance measures in communication systems [70].

Table 1.1 provides a summary of the problems addressed in each chapter, highlighting their problem settings, nature, performance metrics, and information requirements.

Table 1.1: Summary and comparison of the information-freshness studies presented in this thesis.

	Topology & Metric	Interference Model	Link Losses	Information Required	Periodic
Chapter 2	AoI (Multi-hop Tree)	Complete interference / K -hop interference	Yes	Network topology, activation sets, link success probabilities, received packet contents, and past link activations	No
Chapter 3	AoII (Multi-hop Tree)	Complete interference / K -hop interference	Yes*	Network topology, activation sets, source-state change probabilities, AoII estimate, and AoI of each source at the central node	No
Chapter 4	AoI (Multi-hop Tree)	Complete interference / K -hop interference	Yes	Tree topology, hop distances, interference constraints, packet/buffer status, node states, and scheduling history	Yes

Continued on next page

	Topology & Metric	Interference Model	Link Losses	Information Required	Policy Nature
Chapter 5	AoII (Multi-hop Tree)	Complete interference / K -hop interference	Yes*	Topology, interference constraints, period, source-state dynamics, and delivery delay	Yes

* Link losses are discussed as an extension in these chapters.

1.3 Thesis Objectives, Outline, and Contributions

1.3.1 Thesis Objectives

To summarize the discussion in the previous section, there is limited amount of work which considers AoI or AoII and its optimization for multihop networks. Furthermore, policies which are available often require global state information, such as AoI of sources at intermediate nodes and therefore cannot be easily implemented in practical networks. Therefore, the main objective of this thesis is the design and analysis of centralized sampling and scheduling policies for minimizing information freshness metrics such as AoI and AoII in multi-hop wireless networks under interference and packet losses. We consider two major classes of scheduling policies (periodic deterministic and non-periodic) as illustrated in Figure 1.3.

1. Development of non-periodic centralized scheduling policies: for non-periodic scheduling policies a usual approach is to formulate the problem as a sequential decision problem (Markov decision process or Partially observable Markov decision processes). These formulations are usually intractable in general. In this thesis, one of our objectives is to investigate alternate formulations that are analytically tractable and lead to centralized policies that make scheduling decisions on the basis of information in received packets.
2. Development of periodic centralized scheduling policies: for deterministic periodic policies, again a usual approach is to formulate the problem as an integer programming problem. In this thesis, another objective is to obtain low-complexity scheduling policies that have performance close to the optimal solutions obtained from the integer programming formulations.

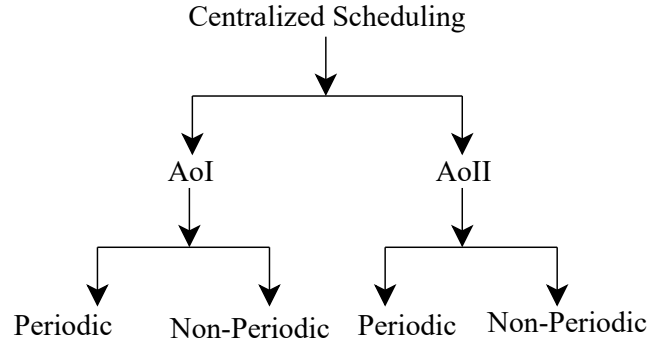


Figure 1.3: Illustration of the thesis research problem on the design of centralized sampling and scheduling policies for minimizing information freshness metrics, such as AoI and AoII, in multi-hop wireless networks. The network consists of multiple source nodes organized in a tree topology and connected to a sink (destination) node through intermediate (parent) nodes. Under centralized control, the sink node determines when source nodes sample monitored processes and generate update packets, which are subsequently forwarded to the destination through the routing tree. This thesis investigates the design of centralized sampling and scheduling policies for minimizing AoI and AoII, considering both periodic and non-periodic scheduling strategies.

3. Furthermore, we aim to obtain a detailed performance comparison of proposed policies and policies from prior work. We also aim to derive baseline policies (which may not be implementable), fundamental analytical characterizations of AoI and AoII performance for multihop networks, as well as upper and lower bounds on AoI and AoII to enable performance analysis and comparison.

1.3.2 Thesis Outline

The main technical contributions of this thesis are organized as outlined below.

Chapters 2 and 3 describe the design of priority-index-based scheduling policies that minimize information freshness metrics, namely the average AoI and average AoII, in a multi-hop sensor network with interference and link losses. The network operates in a convergecast manner, where spatially distributed sensors monitor physical processes, generate status updates, and forward packetized samples to a sink node over a routing tree. The sink forms remote estimates of the monitored processes based on the received updates, and the timeliness of these estimates is quantified using AoI and AoII for Chapters 2 and 3, respectively.

For minimizing the average AoI in Chapter 2, a direct formulation of the joint sampling

and scheduling problem as a Markov Decision Process (MDP) leads to prohibitive computational complexity due to the large state space consisting of the age of every source at every intermediate node in the network. To obtain a tractable solution, we **reformulated the problem as a Restless Multi-Armed Bandit (RMAB)** under a reasonable restriction that maintain only one packet per source in the network at a time. This reformulation enables the development of a **priority-index-based policy** that jointly determines which sources to sample and which links to activate in each time slot. An approach to obtaining priority indices for RMABs is to compute Whittle indices using optimal policies for individual MDPs, which consider policies for activation of each restless arm on its own. However, these optimal policies depend on state information that is not available to a centralized scheduler at the sink. Using our definition of the index, we show how to obtain a priority index based on an analytically tractable policy that can be implemented centrally. The proposed scheduling policy is analytically characterized through a derived priority index, where each source is assigned a value based on its current AoI, and the source with the highest index, indicating the greatest need for an update, is selected for transmission. Extensive simulations are conducted to evaluate the performance under varying network sizes and link reliability conditions. The results demonstrate that the proposed policy achieves lower AAoI compared to existing multi-hop AoI-aware policies. In particular, the proposed priority-index-based policy outperforms prior work as the number of sources increases and link reliability decreases, highlighting its robustness and scalability.

For minimizing the average AoII in Chapter 3, a direct formulation leads to a partially observable Markov decision process (POMDP), since the centralized scheduler does not directly observe the AoII evolution of the source nodes. This makes a direct formulation of the joint sampling and scheduling problem computationally intractable. To obtain a tractable solution, we **reformulate the problem as an RMAB** under a reasonable restriction that only one packet per source is maintained in the network at any time. This reformulation enables the development of a **priority-index-based policy** that jointly determines which sources to sample and which links to activate in each time slot. **Genie-aided policies**, which assume access to full state information, are introduced as a baseline for performance comparison. The indexability of the resulting RMAB problem is rigorously analyzed, and novel closed-form expressions for the corresponding Whittle indices are derived. Motivated by these analytical results, implementable index-based sampling and scheduling policies are proposed. These policies rely only on information available at the destination and are therefore suitable for realistic centralized implementations. Extensive simulation studies are conducted across a variety of network topologies, interference models, and

link reliability conditions. The results demonstrate that the proposed index-based policies achieve significant improvements in AAoI compared to heuristic approaches, such as randomized and AoI-based scheduling policies, and perform close to genie-aided policies.

Overall, Chapters 2 and 3 establish a tractable priority-index-based framework for minimizing the information freshness metrics AoI and AoII, respectively, in multi-hop wireless networks. The proposed approach is designed to operate using only the information available at the central node, thereby ensuring feasibility for practical implementation.

In **Chapters 4 and 5**, we develop **novel periodic deterministic sampling and scheduling algorithms for minimizing information freshness metrics**, namely the average AoI and AoII, respectively, in time-slotted multi-hop wireless sensor networks for remote estimation applications. We consider a convergecast communication paradigm over tree network topologies, where sensor nodes periodically sample physical processes and forward their status updates to a sink over multi-hop wireless links under general K -hop interference constraints. The network operates in a time-slotted manner with periodic deterministic scheduling, motivated by time-slotted protocols such as IEEE 802.15.4e TSCH. These chapters focus on periodic deterministic policies due to their practical relevance in industrial deployments, where predictable and repeatable transmission schedules are preferred.

For AoI problem, remote estimation accuracy at the sink is evaluated using AAoI metric, which quantifies the timeliness of status updates. Focusing on practically implementable periodic deterministic policies, we derive an explicit analytical characterization of the AAoI as a function of the scheduling period and the source-to-sink hop distances. This characterization provides structural insight into how network topology and scheduling periodicity jointly influence information freshness. The AAoI minimization problem under periodic deterministic policies is formulated as an integer programming problem. To obtain a computationally tractable performance baseline, the integer program is relaxed to a linear program, yielding numerical lower bounds on the achievable AAoI. In addition, novel analytical lower bounds are derived for tree networks operating under K -hop interference constraints, establishing fundamental performance limits for periodic policies. Extensive simulations demonstrate that the proposed periodic deterministic algorithms achieve AAoI values close to the derived lower bounds for a given period. Furthermore, performance comparisons with existing multihop AoI-aware scheduling policies, including Age-Difference and Age-Debt, show that the proposed methods provide consistently lower AAoI across a wide range of network settings. Finally, the framework is extended to more general AoI-based performance metrics, illustrating the broader applicability of the proposed policies.

For AoII problem, the sink forms real-time estimates of all monitored processes based

on received status updates, and estimation performance is quantified using the AAoII, a metric that captures both information timeliness and correctness. Unlike classical AoI, AAoII accounts for the mismatch between the true process state and the remote estimate, making it particularly suitable for control and monitoring applications in industrial IoT systems. A novel analytical characterization of AAoII under such policies is developed, expressed explicitly in terms of the scheduling period and the packet delivery delay associated with each sensor node. Leveraging this characterization, we derive a fundamental analytical lower bound on the achievable AAoII for any periodic deterministic sampling and scheduling policy in the considered network. To complement the lower bound, an analytical upper bound on the achievable AAoII is established using a periodic scheduling policy. For the case of networks with complete interference, simulation results demonstrate the near-optimality of this policy with respect to AAoII. Furthermore, the performance of periodic deterministic policies is compared with that of a non-periodic policy through extensive simulations. The results reveal that periodic policies achieve lower AAoII in different instances, highlighting their effectiveness for structured industrial wireless networks.

Overall, Chapters 4 and 5 provide both theoretical performance limits and practically implementable periodic scheduling policies for minimizing AAoI and AoII in multi-hop industrial sensor networks under general K -hop interference constraints.

1.3.3 Thesis Contributions

This thesis contributes to the design and analysis of centralized sampling and scheduling policies for improving information freshness in multi-hop wireless networks. The work considers practical wireless network settings with interference constraints and packet losses, and investigates both AoI and AoII as measures of information freshness. The main contributions of this thesis are summarized as follows:

1. Developed centralized sampling and scheduling for minimizing AoI and AoII in multi-hop wireless networks. For non-periodic scheduling, analytically tractable formulations based on RMAB models, under a restriction that allows at most one packet per source in the network at any time, were investigated. These formulations led to priority-index-based scheduling policies that can be implemented using information available at the sink node.
2. Developed periodic deterministic scheduling policies for minimizing AoI in multi-hop networks under interference constraints. The scheduling design was formu-

lated using optimization-based approaches, and efficient policies were proposed that achieve performance close to optimal solutions.

3. Derived fundamental performance bounds for the information freshness metric AoII under periodic scheduling. Analytical lower and upper bounds were established, which can be used to evaluate and benchmark the performance of any periodic scheduling policy designed to minimize AoII.
4. Established analytical characterizations of information freshness in multi-hop networks. New analytical results were obtained for both AoI and AoII under periodic and non-periodic scheduling policies, providing insights into the impact of network topology, interference constraints, packet losses, and source dynamics on system performance.
5. Conducted extensive performance evaluations and comparisons with existing scheduling policies from the literature. The proposed approaches consistently demonstrated significant performance improvements and, in several scenarios, achieved performance close to the derived theoretical bounds.

The results in this thesis contribute to the understanding of information freshness in multi-hop wireless networks and provide a foundation for timely remote estimation in future industrial and cyber-physical systems.

Chapter 2

Priority-Index Based Sampling and Scheduling Policies for Minimization of Average Age of Information in Time-Slotted Multi-hop Networks

2.1 Introduction and Prior Work

A key aspect of modern Internet of Things (IoT) applications is the remote estimation of physical processes monitored by sensors or source nodes. The spatially distributed source nodes sample the physical processes of interest and generate data packets containing these samples. We consider a convergecast network scenario, where packets are collected by intermediate nodes and forwarded to the sink over multi-hop wireless networks. The sink node receives packets containing the samples, which are used to obtain remote estimates of the monitored processes. We use AoI as a metric to quantify the freshness of information in such remote estimation systems [1]. In this chapter, we consider the design of centralized sampling and scheduling policies for minimizing the average AoI (AAoI) in multi-hop networks. The network is assumed to experience both interference and random link losses. We propose a novel priority-index-based policy derived from a RMAB formulation of the joint sampling and scheduling problem in multi-hop networks. The proposed policy demonstrates improved performance compared to existing AoI-aware multi-hop scheduling policy from prior work.

*This chapter is based on the manuscript by Nibin Raj and Vineeth B.S., “Priority-index based policies for minimizing average AoI in multi-hop network convergecast with complete interference and link losses”, in IEEE Wireless Communications Letters, vol. 15, pp. 4215-4219, 2026, doi: 10.1109/LWC.2026.3715677.

2.1.1 Prior Work

The remote estimation of processes observed by sensors at a sink for estimation or control is an important task in IoT and cyber-physical systems. The AoI metric has been introduced in such applications as a performance measure, as it captures information freshness in contrast to traditional metrics such as delay and throughput. A comprehensive survey on AoI can be found in [33]. AoI has been analyzed in various queuing models [32], [71], [72], [73]. An extensive work on AoI for single-source and single-destination scenarios is presented in [34]. The problem of minimizing AoI in multiple-access networks with a single base station, under both reliable and unreliable channels, where only a subset of links can be activated at a time, has been studied. The optimality of three classes of policies: randomized, maximum-weight, and Whittle index-based has been analyzed in [74], [75], [76].

The study of AoI in multi-hop networks has received limited attention. The problem of minimizing the AAoI under stationary randomized policies is considered in [42], whereas interference-free multi-hop information flows are studied in [43]. Other studies have investigated specific network topologies and communication settings, including ring topologies [44] and time-slotted multi-source multi-hop networks [47]. The problem of AoI minimization in single-hop networks using distributed scheduling mechanisms has been studied in [61] and [59], which employ distributed convex optimization and CSMA-based access schemes. Recent work in [62] demonstrated the potential of reinforcement learning for optimizing AoI and generalized freshness metrics in single-hop networks. However, corresponding RL-based solutions for multi-hop networks have not been extensively investigated. AoI minimization in multi-hop networks was investigated in [9], where the authors analyzed the performance of three scheduling policies, namely the stationary randomized, Age-Difference, and Age-Debt policies. Restless multi-armed bandit (RMAB) formulations for minimizing the average AoI (AAoI) have been extensively studied in single-hop networks [63], [64], [65], as well as in single-hop broadcast networks [66]. However, the application of RMAB-based approaches to AAoI minimization in multi-hop networks remains largely unexplored. Table 2.1 provides an overview of single-hop network sampling and scheduling policies using classical AoI minimization, machine learning (ML)-based, or reinforcement learning-based methods.

Table 2.1: Overview of sampling and scheduling policies reported in prior work.

Model discussed	Metric used and (Network Type)	Policy
A single source generates time-stamped status updates that are delivered to a monitor through a single-server FCFS queue [10]	AoI (Single-hop)	Optimal update generation policy
A source generates update packets and transmits them to a destination over a channel, with access to channel status via ACKs, and generates updates only when the channel is idle to avoid queueing [11].	AoI (Single-hop)	Zero-Wait policy
An energy-harvesting source sends status updates to a monitor through a single-server FCFS system [77]	AoI (Single-hop)	Lazy update policy
A base station delivers generate-at-will status updates to multiple clients in a slotted downlink system, adaptively switching between OMA and NOMA [78].	AoI (Single-hop)	Adaptive NOMA/OMA AoI-optimal scheduling policy
Single-hop wireless links with pairwise interference and stochastic channels [61].	AoI (Single-hop)	Distributed stationary scheduling policy
A single-hop wireless network where multiple sources contend for a shared channel using CSMA, incurring random access delays [79].	AoI (Single-hop)	CSMA-based randomized access policy

Continued on next page

Model discussed	Metric used and (Network Type)	Policy
Large-scale single-hop status update system with centralized scheduling [80].	AoI (Single-hop)	Cyclic (round-robin based) scheduling policy
A single-hop vehicular network where vehicles transmit status updates to a roadside unit over a shared wireless channel, considering freshness and task completion deadlines [81].	AoI and completion time (Single-hop)	Joint AoI and completion-time optimization policy
A single-hop WiFi monitoring network using IEEE 802.11 DCF contention [82].	AoI (Single-hop)	DCF-based scheduling policy
A single-hop heterogeneous CSMA network with unsaturated sources, modeled using a stochastic hybrid system (SHS) [83].	AoI (Single-hop)	CSMA random access (DL-augmented SHS) policy
A single-hop wireless D2D network with multiple interfering links under full frequency reuse, where each link generates time-sensitive updates and transmission success depends on mutual interference [84].	AoI (Single-hop)	DL-based stationary randomized scheduling policy
Learning-based forecasting with multiple transmitters sending correlated features to a receiver using ERM-based supervised learning (e.g., LSTM) with time-varying AoI [85].	AoI of features; Training and testing loss (Single-hop)	Joint training with AoI-aware feature augmentation

Continued on next page

Model discussed	Metric used and (Network Type)	Policy
Real-time supervised learning system where a sensing node transmits features to a learner for inferring a time-varying target. A freshness-aware ERM is used, where feature values and AoI are inputs to the neural network. A buffer-based transmission model allows selecting previously generated features [67].	AoI; Time-average inference error (Single-hop)	1. Gittins index policy (optimal for single source) 2. Whittle index policy (low-complexity for multi-source)
Edge-enabled wireless IoE network where multiple devices generate status updates, scheduled via a two-stage process with device selection and transmission over shared wireless resources [86].	AoI (Single-hop)	Deep reinforcement learning-based scheduling policy
Single-hop V2I downlink where an RSU schedules one vehicle per slot. The AoI-aware scheduling problem is formulated as a Markov Decision Process and solved using Soft Actor-Critic (SAC) [87].	AoI (Single-hop)	Soft Actor-Critic (SAC) based scheduling policy
A remote monitoring URLLC system, where a central controller schedules one sensor at a time to transmit time-sensitive updates over a stochastic network and formulates the AoI optimization problem using a reinforcement learning-based scheduling framework [88].	AoI, AoI threshold violation probability (Single-hop)	RL-based scheduling using Actor-Critic algorithm.

2.1.2 Contributions

We first consider a network with complete interference and link losses, where transmission over each link is successful with probability p_s .

1. The main contribution of this chapter is the centralized policy π_{Index} , which determines sampling and link activation decisions based on a priority index.
2. Through simulations, we demonstrate that π_{Index} achieves lower AAoI compared to existing policies such as stationary randomized, Age-Difference, and Age-Debt [9]. The performance improvement of the proposed policy becomes more pronounced as the network size increases or the link success probability p_s decreases, demonstrating the effectiveness of the proposed π_{Index} policy.
3. Furthermore, compared to prior policies such as Age-Difference and Age-Debt [9], the proposed π_{Index} policy utilizes only the AoI information available at the sink node, whereas these prior policies require AoI information from all nodes in the network at any time.
4. We also extended the proposed priority index-based policy to networks with K -hop interference constraints. Simulation results demonstrated that the proposed π_{Index} policy, achieves significant performance gains over the Age-Difference policy.
5. Additionally, extensive simulations were conducted across a variety of network scenarios, including heterogeneous networks with link-dependent success probabilities $p_{s,l}$, symmetric tree networks, single-hop networks, two-hop networks, and multi-flow scheduling networks. Across all considered scenarios, the proposed index-based policy, π_{Index} , consistently achieved significant performance improvements over the Age-Difference policy.

We note that usual approaches for obtaining an AAoI optimal policy for a multi-hop network leads to MDP formulations (assuming that AoI at every node for every source in every slot is known to the scheduler as a state vector, the formulation is a POMDP if AoI is not directly observed). The exact solutions of such MDPs are intractable. We show that there is a simpler RMAB formulation for multi-hop networks by considering policies that maintain only one packet per source in the network at a time. The policy π_{Index} is justified using this RMAB. The restriction is justified by the improved AAoI performance for all values of p_s and motivated by the analytical tractability of the resulting RMAB. We discuss

how the RMAB formulation can also be used for K -hop interference models illustrating the utility of this approach to policy design.

An approach to obtaining priority indices for RMABs is to compute Whittle indices using optimal policies for *individual MDPs*, which consider policies for activation of each restless arm on its own. We observe that optimal policies for individual MDPs in our problem introduce dependencies on state information that are not available to a centralized scheduler implemented at the sink. Using our definition of π_{Index} we demonstrate how a heuristic priority index could be obtained for individual MDPs using an analytically tractable and centrally implementable policy rather than an optimal policy. This is a major difference from prior work on Whittle index policies, including our prior work¹, in which we obtained multi-hop sampling and scheduling policies for minimizing average age of incorrect information (AAoII) (rather than AAoI) from Whittle index policies for an RMAB formulation.

Notations: The notations used in this chapter are given in Table 2.2.

Table 2.2: List of symbols and their descriptions used in Chapter 2

Symbol	Description
$[M]$	Set of integers $\{1, 2, \dots, M\}$.
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Directed tree representing the multi-hop network, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of directed edges.
$\mathcal{S}$	Set of source nodes (sensor nodes) that generate packets for the root node.
$ U $	Cardinality of the node set U .
$(i, j) \in \mathcal{E}$	Directed edge from node i to node j , representing a communication link from i to j .
$(s, (i, j))$	Link activation on link (i, j) for source s .
$(s, (s, j))$	Sampling of source s and activation of link (s, j) .
$\mathcal{P}(s)$	Set of nodes along the path from source node s to the root node 0.
$A_i^s(t)$	AoI at node $i \in \mathcal{P}(s)$ for source s at time slot t .
$g_i^s(t)$	Generation time of the latest packet from source s received at node i until the beginning of slot t .

¹Nibin Raj and Vineeth Bala Sukumaran, "Design of Centralized Scheduling Policies for Minimizing Average Age of Incorrect Information in Multi-Hop Wireless Networks With Interference," in IEEE Access, vol. 14, 2026, doi: 10.1109/ACCESS.2026.3674431.

$\mathcal{I}_t$	Information available to the scheduler at the beginning of time slot t , based on which scheduling decisions are made.
p_s (or q_s)	Probability that a packet transmission from node i to node j along link (i, j) is successful (or unsuccessful) for source s .
$\overline{A}^\pi(s)$	Average AoI of source s under scheduling policy π .
$\overline{A}^\pi$	AoI averaged across all sources for scheduling policy π .
$U(t)$	Activation set at time t .
h_s	Hop distance of source s from sink node 0.
$A^s(t)$	AoI of source s at node 0 at time t .
$H^s(t)$	Hop distance from node 0 to the node currently buffering source s 's packet at time t .
$S^s(t)$	Time elapsed since the currently buffered packet of source s was generated.
$\overline{D}^\pi(s)$	Average fraction of time that source s is scheduled under policy π .
$\overline{D}^\pi$	Sum of $\overline{D}^\pi(s)$ over all sources under policy π .
τ_s	Threshold value associated with source s .
K	A positive integer in K -hop interference constraint.

2.2 System Model and Problem Statement

The network is modeled as a directed multihop tree graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of nodes and the set of directed communication links are denoted by $\mathcal{V}$ and $\mathcal{E}$, respectively. Tree network topologies are commonly utilized in practical network deployments, with protocols such as RPL [89]. We consider the network with convergecast communication, where the sink node 0 collects information from all nodes in a set $\mathcal{S}$, called the sensor nodes. Also, let $|\mathcal{V}| = M + 1$, indicating that there are M nodes in the network other than the root node 0, and $|\mathcal{S}| \leq M$. An element $(i, j) \in \mathcal{E}$ indicates that there exists a communication link between nodes i and j , through which packets can be forwarded in the tree network. Time is assumed to evolve in discrete slots, indexed by $t \in \mathbb{Z}^+$. In a given time slot t , a communication link (i, j) can transmit at most one packet corresponding to a source $s \in \mathcal{S}$ from node i to node j , which is referred to as a *link activation* and is denoted by $(s, (i, j))$. Here, in $(s, (i, j))$, if $s = i$, it indicates that the source node s samples and generates a packet, which is then forwarded to node j . The packet contains a time stamp corresponding to its generation time along with the data. Another possibility is $s \neq i$,

which means that the latest packet from source s currently present in node i 's buffer is transmitted from node i to node j in time slot t . If no such packet is present in node i 's buffer, then no transmission occurs. Also, we assume that all generated packets are of the same size.

We assume that transmissions over all links in the network occur on the same communication channel, and that each node is equipped with a single half-duplex radio interface. We consider a link success probability for every link in the network. Specifically, for any link activation $(s, (i, j))$, i.e., the transmission of a packet from node i to node j , the transmission is successful with probability p_s and fails with probability $q_s = 1 - p_s$. If a packet from source s is successfully transmitted from node i to node j , then node j stores the packet in its buffer, replacing any older packet of source s if present. If the transmission of a packet from source s from node i to node j fails, then (i) node i retains the packet of source s in its buffer if $i \neq s$, and (ii) if $i = s$, then a new packet will be generated when $(s, (s, j))$ is scheduled in its next activation. We consider both the complete interference model and the K -hop interference model. Under complete interference, only a single link can be activated at any given time². In contrast, the K -hop interference model allows multiple links to be activated simultaneously, provided that any two active links are at least K hops apart, thereby avoiding interference. We initially consider the complete interference model and then extend to the K -hop model.

The activation of a link in time slot t is denoted by $U(t)$. The evolution of packet transmission and sampling decisions (which are of the form $(s, (s, .))$) over the network forms a sequence or pattern over time, which is denoted as $(U(t), t \in \mathbb{Z}_+)$. This sequence is called *multi-hop sampling and scheduling policy*, denoted as π and generated by a *scheduler*.

AoI is defined as the time elapsed since the generation of the most recently received update. We consider a multihop tree network, where a source node s forwards its packets to the destination node 0 through a multihop path. Let $\mathcal{P}(s)$ denote the set of nodes along the path from s to 0. Then, the AoI at a node $i \in \mathcal{P}(s)$ at time slot t is denoted by $A_i^s(t)$. The initial age is assumed to be zero, i.e., $A_i^s(0) = 0$. The AoI is given by,

$$A_i^s(t) = t - g_i^s(t) \quad (2.1)$$

where $g_i^s(t)$ denotes the generation time of the most recently received packet from source

²The derived policy can be extended to K -hop interference scenarios by observing that, the direct links connecting 0 to its child nodes and the links which are within K hops of such direct links still cannot be scheduled together (as in complete interference). A priority index policy can then be defined for such networks (see Section 3.3).

s at node i . It is assumed that the initial value is zero, i.e., $g_i^s(0) = 0$.

Let $\bar{A}^\pi(s)$ be the AAoI for a source s at the node 0 under policy π , defined as:

$$\bar{A}^\pi(s) = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[A_0^s(t) | A_0^s(0) = 0] \quad (2.2)$$

where the initial AoI is assumed to be 0.

We now define AAoI for the system under policy π as,

$$\bar{A}^\pi = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \bar{A}^\pi(s) \quad (2.3)$$

The scheduling decision depends on the information available to the scheduler. Specifically, at the beginning of time slot t , the scheduler makes its decision based on the information set $\mathcal{I}_t$ available to it. The information $\mathcal{I}_t$ depends on whether the scheduler is centralized or distributed. In this chapter, we focus on the design of centralized scheduling policies, where the scheduler is implemented at the sink node 0 and makes decisions based on the information available at node 0. Centralized scheduler implementation is practically supported by many existing protocols, for instance, Path Computation Element in IEEE 802.15.4e TSCH (Time-Slotted Channel Hopping) [4] and Network Manager in WirelessHART [90] enable centralized policy implementation. We now represent the set of all such centralized policies by Π . For any policy $\pi \in \Pi$, we assume that it has knowledge of the network topology, interference constraints, and the link success probability p_s . Also, for any $\pi \in \Pi$, the information set $\mathcal{I}_t$ consists of all packets, including their contents, that have been successfully received at node 0 before time slot t , as well as the past link activation decisions $(U(\tau), \tau < t)$. Under this assumption, and using packet generation time stamps, $(A_0^s(\tau), \tau \leq t), \forall s$, is known to the scheduler.

Additionally, when $p_s = 1$, i.e., in the reliable link case, the evolution of packets in the network is completely determined by the scheduling decisions $(U(\tau), \tau < t) \in \mathcal{I}_t$. Since there are no transmission failures, every scheduled transmission is successful; hence, the scheduler at node 0 can exactly track the evolution of the AoI for packets from each source s along the path $\mathcal{P}(s)$. Moreover, once a packet from source s is received at node 0, its AoI increases deterministically with time until a new packet is received. Therefore, using the packet generation time stamps and past scheduling decisions, the scheduler can infer $(A_i^s(\tau), \tau \leq t)$ for all $i \in \mathcal{P}(s)$ and for all s . Furthermore, the scheduling decisions also provide information on whether node i contains a packet from source s at time t .

However, when $p_s < 1$, the AoI values of sources other than the sink node 0 is not fully known to node 0, even with the knowledge of $(U(\tau), \tau < t)$, due to possible transmission failures caused by link losses. For our discussion, we also find it useful to define a class Π_G of *genie* schedulers with $\mathcal{I}_t$ containing $(A_i^s(t), \forall i \in \mathcal{P}(s), \forall s)$ at slot t .³

In this chapter our objective is to design centralized sampling and scheduling policies to minimize $\bar{A}^\pi$, i.e.

$$\min_{\pi \in \Pi} \bar{A}^\pi \quad (2.4)$$

We formulate the problem of centralized sampling and scheduling to minimize AoI as a MDP, assuming that all links are reliable, i.e., $p_s = 1$. The quantity $\bar{A}^\pi$ denotes the time average of $A_0^s(t)$. We first consider the state vector for the MDP as $(A_0^s(t), \forall s)$. It is not possible to determine the evolution of $A_0^s(t)$ to $A_0^s(t+1)$ under $U(t)$ using only the knowledge of $(A_0^s(t))$. If we include $(U(\tau), \tau < t)$ or the AoI $(A_i^s(t), \forall i \in \mathcal{P}(s), \forall s)$, along with the buffer state indicating whether node i contains a packet from source s at time slot t , as part of the state process, which results in a *Markov* evolution. The average-cost MDP can be formulated with a single-stage cost given by $\sum_{s \in \mathcal{S}} A_0^s(t)$. However, this MDP is challenging to solve due to the large dimensionality of the state space. For example, consider a line network consisting of M nodes in addition to the sink node 0. In this case, the size of the state space grows on the order of M^2 .

When considering the case where $p_s < 1$, even though a policy $\pi \in \Pi$ has knowledge of $(A_0^s(t), \forall s)$, it does not observe $(A_i^s(t), \forall i \in \mathcal{P}(s), \forall s)$ for nodes other than the sink. Due to link losses, transmissions may not be successful. Hence, one can maintain a *beliefs* over (A_i^s) for $i \neq 0$, which results in a POMDP formulation. However, this formulation is, in general, not analytically tractable. RL based methods provide a possible approach; however, in this chapter, we focus on analytically tractable policies.⁴ In the following, we show that (2.4) can be formulated as a RMAB problem for a subset of policies in the class Π_G .⁵

³We note that only for $p_s = 1$, the Age-Difference scheduler $\in \Pi$, while for $p_s < 1$, the Age-Difference scheduler $\in \Pi_G$.

⁴In this chapter, we consider networks with complete interference. We note that more general interference models, such as K -hop interference, allow multiple simultaneous link activations, leading to large action spaces that are again challenging for learning-based approaches.

⁵An important observation is that an MDP formulation for policies in the class Π_G also results in a state space with the same dimensionality as that discussed for Π when $p_s = 1$.

2.3 RMAB Formulation and a Priority-Index Policy

We consider a network with complete interference, i.e., at each time slot, the scheduling policy selects a single link activation. We focus on a class of policies denoted by $\Pi_{1P} \subset \Pi_G$, in which, once a packet from a source is generated, no new packet is generated from that source until the previously generated packet is received at the sink node 0.

For a policy $\pi \in \Pi_{1P}$, the scheduler's decision at each time slot is thus reduced to selecting a source for transmission. Under this policy, the evolution of the system can be described as follows: In a tree network topology, each source node s has a unique path to the sink node 0. Let $H^s(t)$ represent the hop distance from the sink node 0 to the node currently holding the packet of source s at time slot t . The value of $H^s(t)$ uniquely identifies which node is buffering the current packet of source s , since each source has at most one packet present in the network. Let $S^s(t)$ denote the time elapsed since the current packet from source s in the network was generated. This quantity is useful for tracking the evolution of the AoI, $A_0^s(t)$. If source s has no packet in transmission, then $S^s(t) = 0$ and $H^s(t) = h^s$, where h^s denotes the hop distance of source s from the sink node 0. For the system, we define the state vector as $((A^s(t), S^s(t), H^s(t)), \forall s \in \mathcal{S})$, where, in this section, we use $A^s(t)$ to represent $A_0^s(t)$. In addition, we define $D^s(t)$ as an indicator variable that takes the value 1 when source s is scheduled at time slot t under policy π , and 0 otherwise. In Table 2.3, we define the evolution of the state vector $(A^s(t), S^s(t), H^s(t))$ as a function of $D^s(t)$ from time slot t to $t + 1$.

For a policy $\pi \in \Pi_{1P}$, the state vector $((A^s(t), S^s(t), H^s(t)), \forall s \in \mathcal{S})$ evolves as a Markov chain (MC). The state space given by the Cartesian product $\times_{s \in \mathcal{S}} (\mathbb{Z}_+ \times \mathbb{Z}_+ \times [h^s])$. The action space is $\mathcal{S}$, since at each time slot the scheduler must choose a source s to be scheduled. For a given state vector $((a_s, s_s, h_s), \forall s)$, we define the single-stage cost as $\sum_s a_s$. We note that the AAoI minimization problem (2.4) can be formulated as an average-cost MDP with state transitions as described in Table 2.3. This MDP is a RMAB problem. Let $\overline{D}^\pi(s)$ denote the average fraction of time that source s is scheduled under policy π , defined as

$$\overline{D}^\pi(s) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[D^s(t)] \quad (2.5)$$

Also, let $\overline{D}^\pi$ denote the average fraction of time that system is scheduled under policy π , defined as

$$\overline{D}^\pi = \sum_s \overline{D}^\pi(s) \quad (2.6)$$

Table 2.3: Evolution of $S^s(t)$, $A^s(t)$, and $H^s(t)$ from t to $t + 1$ as a function of $D^s(t)$ for source s (note: w.p. is for “with probability”).

	$D^s(t) = 0$	$D^s(t) = 1$
$S^s(t + 1)$	$S^s(t) + 1,$ if $H^s(t) < h^s$, 0, if $H^s(t) = h^s$.	w.p. q_s : $S^s(t) + 1$, if $H^s(t) < h^s$, $S^s(t)$, if $H^s(t) = h^s$. w.p. p_s : $S^s(t) + 1$, if $H^s(t) < h^s$, 0, if $H^s(t) = 1$.
$A^s(t + 1)$	$A^s(t) + 1$	w.p. q_s : $A^s(t) + 1$. w.p. p_s : $A^s(t) + 1$ if $H^s(t) < h^s$, $S^s(t) + 1$, if $H^s(t) = 1$.
$H^s(t + 1)$	$H^s(t)$	w.p. q_s : $H^s(t)$ w.p. p_s : $H^s(t) - 1$, if $H^s(t) < h^s$, h^s , if $H^s(t) = 1$.

An important and useful approach to such problems is due to Whittle [91], where the hard constraint that only a single source can be scheduled (i.e., $\forall t \sum_s D^s(t) = 1$) is relaxed into an average constraint as follows:

$$\min_{\pi \in \Pi_{1P}} \bar{A}^\pi \text{ such that } \bar{D}^\pi \leq 1. \quad (2.7)$$

This problem can be formulated as an MDP (with the same state space and transitions as described above) with an average cost of $\bar{A}^\pi + \lambda \bar{D}^\pi$, where $\lambda \geq 0$ is a Lagrange multiplier. We refer to this formulation as the Lagrange-MDP (L-MDP). For the L-MDP, the single-stage cost is given by

$$\sum_s a_s + \lambda \sum_s d_s \quad (2.8)$$

where $d_s \in \{0, 1\}$ indicates whether source s is selected for transmission, corresponding to the specific value of $D^s(t)$. The L-MDP decouples into $|S|$ individual MDPs, with each MDP corresponding to a single source. The state of each individual MDP is $(A^s(t), S^s(t), H^s(t))$, and the single-stage cost is $a_s + \lambda d_s$. We refer to the MDP for a single source with the average-cost criterion as the Single-Source MDP (SS-MDP). For a

source s , the objective of SS-MDP is

$$\min_{\pi} \overline{A}^{\pi}(s) + \lambda \overline{D}^{\pi}(s), \quad (2.9)$$

where now π is a policy for a single source that decides $D^s(t)$. The Whittle index approach [63] leverages the solution of the SS-MDP to derive a *good* policy, referred to as the index policy, for the RMAB. The approach to obtaining an index policy consists of the following steps: (i) obtaining an optimal policy for SS-MDP, (ii) showing *indexability* of the obtained optimal policy, (iii) obtaining the Whittle index associated with every state of the SS-MDP [63].

Let a specific instance of state of the SS-MDP be denoted by $x_s = (a_s, s_s, h_s)$, representing $(A^s(t), S^s(t), H^s(t))$. The Whittle index for a given state can be interpreted as a *subsidy for passivity*. For a state x_s , the Whittle index denoted as $w(x_s)$ is defined as the smallest value of λ for which the optimal action for state x_s is to be passive or 0 [63]. The policy for the RMAB selects, at each time slot, the source with the maximum Whittle index. The Whittle index for each source depends on its current state (a_s, s_s, h_s) . The Whittle index is a priority index, determining which source should be scheduled based on its current state, with higher index values corresponding to higher scheduling priority.

We propose a priority-index based policy using the following definition of a index for each state for SS-MDP. We consider (2.9) but only for a subset Π_{τ} of threshold policies on age for a single source. A threshold policy π_{τ_s} on age decides $D^s(t) = 1$ if $A^s(t) \geq \tau_s$ and 0 otherwise, where $\tau_s \in \mathbb{Z}_+$ is the threshold parameter⁶. The priority index $\nu(x_s)$ for a state x_s is defined as the smallest value of λ for which the optimal action for x_s is to be passive, where optimality means that the minimum of the following problem is achieved.

$$\min_{\pi_{\tau_s} \in \Pi_{\tau}} \overline{A}^{\pi_{\tau_s}}(s) + \lambda \overline{D}^{\pi_{\tau_s}}(s), \quad (2.10)$$

The index $\nu(x_s)$ is the Whittle index $w(x_s)$ if threshold policies are optimal.⁷ For a λ , the optimal threshold $\tau_s^*(\lambda)$ for (2.13) is $\arg \min_{\tau_s} \overline{A}^{\pi_{\tau_s}} + \lambda \overline{D}^{\pi_{\tau_s}}$, where $\overline{A}^{\pi_{\tau_s}}$ and $\overline{D}^{\pi_{\tau_s}}$ are the AAOI and fraction of time source s is scheduled for the policy π_{τ_s} . We characterize $\overline{A}^{\pi_{\tau_s}}$ and $\overline{D}^{\pi_{\tau_s}}$ below.

⁶We note that the threshold is not considered to be a function of other components of the state in the SS-MDP.

⁷A main motivation behind defining a priority index policy is that it allows us to define index policies using policies for SS-MDP which are analytically more tractable or which are independent of information not available to a centralized scheduler. For example, here an optimal policy for SS-MDP could be a function of $H^s(t)$ and $S^s(t)$ also, but we consider threshold policies on just $A^s(t)$.

Proposition 2.1. *The AAoI $\overline{A}^{\pi_{\tau_s}}$ for the threshold policy π_{τ_s} for a single source is*

$$\overline{A}^{\pi_{\tau_s}} = \frac{\frac{h^s(3h^s-1)}{2p_s^2} + \sum_{a=h^s}^{\tau_s-1} \rho(a) \frac{(\tau_s-a)}{2} \left(a + \tau_s + \frac{2h^s}{p_s} - 1 \right)}{\delta} \quad (2.11)$$

The average fraction of time that source s is scheduled under policy π_{τ_s} is $\overline{D}^{\pi_{\tau_s}} = h^s/\delta p_s$. Here $\delta = \left(\frac{h^s}{p_s} + \sum_{a=h^s}^{\tau_s-1} \rho(a)(\tau_s - a) \right)$ and $\rho(a)$ is $\binom{a-2}{h^s-2} p_s^{h^s-1} q_s^{a-h^s}$ for $h^s \geq 2$. For $h^s = 1$ we have $\rho(1) = 1$ and $\rho(a) = 0, \forall a \neq 1$.

The characterization is obtained using Markov renewal reward theorem (MRRT) [92]; a proof outline is given in Appendix 2.A.

The above characterization can be used to obtain $\tau_s^*(\lambda)$. We defined $\nu(x_s)$ as the smallest value for λ such that the threshold $a_s + 1$ is optimal for (2.13), since for λ less than such a $\nu(x_s)$ the action at a_s is 1. We now show that this smallest value of λ or $\nu(x_s)$ for which the threshold is $a_s + 1$ can be obtained using the characterization of $\overline{A}^{\pi_{\tau_s}}$ and $\overline{D}^{\pi_{\tau_s}}$ from Proposition 2.1.

Proposition 2.2. *The priority index $\nu(x_s)$ for state $x_s = (a_s, s_s, h_s)$ is*

$$\nu(x_s) = \frac{\overline{A}^{\pi_{a_s+1}} - \overline{A}^{\pi_{a_s}}}{\overline{D}^{\pi_{a_s}} - \overline{D}^{\pi_{a_s+1}}}. \quad (2.12)$$

Proof. We defined the priority index $\nu(x_s)$ as the smallest value of λ for which it is optimal to be passive under a threshold policy for the problem ⁸:

$$\min_{\pi_{\tau_s} \in \Pi_{\tau}} \overline{A}^{\pi_{\tau_s}}(s) + \lambda \overline{D}^{\pi_{\tau_s}}(s), \quad (2.13)$$

where Π_{τ} is the class of threshold policies and τ_s denotes the threshold for a threshold policy (a policy that serves if the current age $a_s \geq \tau_s$). For threshold policies the smallest value of λ for which it is optimal to be passive at $x_s = (a_s, s_s, h_s)$ is that smallest value of λ for which the optimizing threshold for (2.13) is $a_s + 1$. That is,

$$\overline{A}^{\pi_{\tau_s}}(s) + \lambda \overline{D}^{\pi_{\tau_s}}(s) \geq \overline{A}^{\pi_{a_s+1}}(s) + \lambda \overline{D}^{\pi_{a_s+1}}(s) \quad (2.14)$$

⁸A large value of $\nu(x_s)$ implies that we continue to be active in a state x_s until we place a large penalty on service, so that we should give priority to service if the current state is x_s compared to another state with a smaller value of $\nu(\cdot)$.

Or, we have that

$$\lambda \geq \frac{\overline{A}^{\pi_{a_s+1}}(s) - \overline{A}^{\pi_{\tau_s}}(s)}{\overline{D}^{\pi_{\tau_s}}(s) - \overline{D}^{\pi_{a_s+1}}(s)} \quad (2.15)$$

We have that the smallest value of λ such that $a_s + 1$ is optimal is when $\tau_s = a_s$ in the above expression. This leads to $\nu(x_s) = (\overline{A}^{\pi_{a_s+1}} - \overline{A}^{\pi_{a_s}})/(\overline{D}^{\pi_{a_s}} - \overline{D}^{\pi_{a_s+1}})$. $\square$

It is important to note that the index depends only on a_s and not on (h_s, s_s) .

Priority index policy: We define the priority-index policy π_{Index} as choosing the source s with maximum value of $\nu(x_s)$ at time t (ties are broken randomly). We note that although π_{Index} was obtained by considering $\Pi_{1P} \subset \Pi_G$, $\pi_{Index} \in \Pi$ since it only uses the current $A^s(t)$ which is available at 0.

2.4 Extension to K-hop Interference Model

In this section, we illustrate how the RMAB based approach can be extended to the K -hop interference model. Consider a line network with M nodes, we note that the first $K + 1$ links operate under complete interference conditions. For example, for 2-hop interference model, the links $(1, 0), (2, 1), (3, 2)$ cannot be scheduled together. Consider the sources $1, 2, \dots, K + 1$. Their packets can be transmitted to 0 using the links $(K + 1, K), \dots, (2, 1), (1, 0)$ under complete interference. Furthermore, the packets from $K + 2, \dots, M$ are forwarded to the node $K + 1$ and buffered there. These packets are also transmitted using the above links under complete interference. Suppose $A_0^s(t)$ denotes the AoI of source s at node 0 and $A_{K+1}^s(t)$ denotes the AoI of source s at node $K + 1$ (only for sources $s \in \{K + 2, \dots, M\}$). Since the index in Proposition 2.2 is a function of the AoI a_s , we use the notation $\nu(a_s)$ here. For all sources s , for a link $l \in \{(K + 1, K), \dots, (2, 1), (1, 0)\}$ we define a priority value p_l^s at slot t as follows:

$$p_{l_s}^s = \nu(A_0^s(t)), \forall s \in \{1, \dots, K + 1\}.$$

For sources $s \in \{K + 2, \dots, M\}$ we compute indices by considering these sources as being at hop distance $K + 1$ itself (since their packets are buffered at $K + 1$). We define the following priority values:

$$p_{l_s}^s = \nu(A_0^s(t) - A_{K+1}^s(t)), \forall s \in \{K + 2, \dots, M\}.$$

For all $s \in \{1, \dots, M\}$, the link l_s is the link at which the source's latest packet is buffered. For any other l , p_l^s is set as 0. Also,

$$s_l^* = \arg \max p_l^s.$$

The priority values of sources at the links $((K+2, K+1), (K+3, K+2), \dots, (M, M-1))$ are defined using Age-Difference. That is, $p_l^s = \max(A_j^s - A_i^s, 0)$, where $l = (i, j)$. Then, the priority value of a link activation vector $\mathbf{v}_1$ is obtained as:

$$\gamma \sum_{l=(i,j) \in v_1, i \leq K+1} p_l^{s_l^*} + \sum_{l=(i,j) \in v_1, i > K+1} p_l^{s_l^*}. \quad (2.16)$$

Here, γ is an empirically chosen weight factor for combining the priority values obtained from priority indices with the priority values obtained from Age-Difference. Empirically, we have observed that choosing $\gamma = 1/(\max(M - K, 0) + 1)$ gives good AAoI performance. The proposed scheduler then chooses a link activation vector which has the maximum value of (2.16) in every slot t .

2.5 Simulations and Performance Analysis

In this section, we present the simulation setup and performance analysis. Simulations are used to evaluate the performance of AAoI, denoted by $\overline{A}^\pi$, for the proposed index policy π_{Index} . The results are compared with existing policies from prior work, including stationary randomized, Age-Difference, and Age-Debt [9]. The generation of network topologies is described as follows. We consider *line*, *multiline*, and *tree* networks for different network sizes determined by M . The topologies are constructed such that the maximum hop distance across sources is comparable across different network types. A line network is a linear graph with nodes indexed from 0 to M , where each node i transmits to node $i - 1$ for $i \in \{1, 2, \dots, M\}$. We consider $M \in \{10, 15, 20, 25, 30\}$. A multiline network is a tree consisting of b top-subtrees⁹, where each top-subtree is a line network. Let n_i denote the number of nodes in the i th top-subtree, for all $i \in [b]$. The values $\{n_i\}$ are sampled from a multinomial distribution with M trials, b mutually exclusive outcomes, and equal probabilities, such that $\sum_{i=1}^b n_i = M$. In the experiments, we set $b = 2$ and generate 100 random graphs for each M . The maximum value of M is limited to 50 to ensure that the maximum hop distance remains comparable to that of line networks. Tree topologies are

⁹A tree network consists of *top-subtrees*, where each top-subtree is a subtree rooted at a child of node 0.

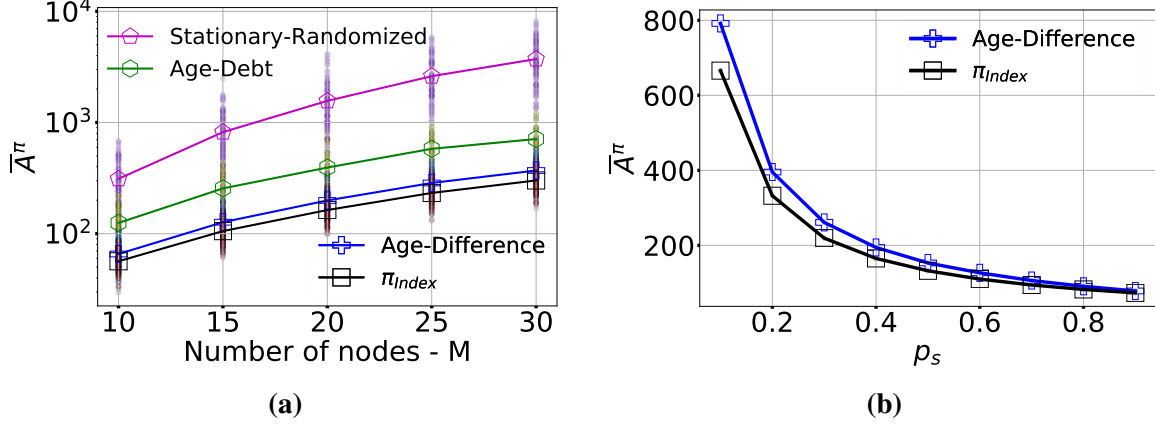


Figure 2.1: Comparison of AAoI: (a) stationary randomized, Age-Debt, Age-Difference, and π_{Index} policies for $p_s = 0.3$ on random tree networks versus M ; (b) Age-Difference and index policies versus p_s for a random tree network with $M = 30$.

generated by uniformly sampling from all possible tree topologies with M nodes using a Prüfer sequence-based construction method [93]. For each value of M , 100 random graphs are generated. The maximum value of M is limited to 100 to maintain a maximum hop distance comparable to that of line networks.

First, we compare the AAoI for networks with a fixed probability of link success $p_s = 0.3$ and varying values of M . The results are shown in Fig. 2.1(a). The AAoI values obtained for these graphs are shown along with the average AAoI across all graphs, where the thick lines represent the average values evaluated at the M values indicated by the markers. The results demonstrate that the proposed π_{Index} policy achieves the smallest AAoI among all considered multihop policies. Specifically, for $M = 30$, the AAoI under the Age-Difference, Age-Debt, and stationary randomized policies is higher than that of the proposed index policy by at least 22.35%, 135.00%, and 1075.48%, respectively. Therefore, in the following, we compare the proposed policy only with the next best policy, namely Age-Difference. Next, we study the AAoI performance of the proposed index policy π_{Index} and the Age-Difference policy as a function of the link success probability p_s in Fig. 2.1(b) for a single network. The results demonstrate the effectiveness of the proposed index policy π_{Index} under unreliable link conditions. Note that we assume only a single packet per source is present in the network at any given time. For $p_s = 0.9$, the AAoI under the Age-Difference policy is 7.01% higher than that under π_{Index} , while for $p_s = 0.1$, it is 18.90% higher.

Also, we compare the AAoI of the π_{Index} and Age-Difference policies for line, multilink, and tree networks under different link success probabilities $p_s \in \{0.3, 0.6, 1\}$ and

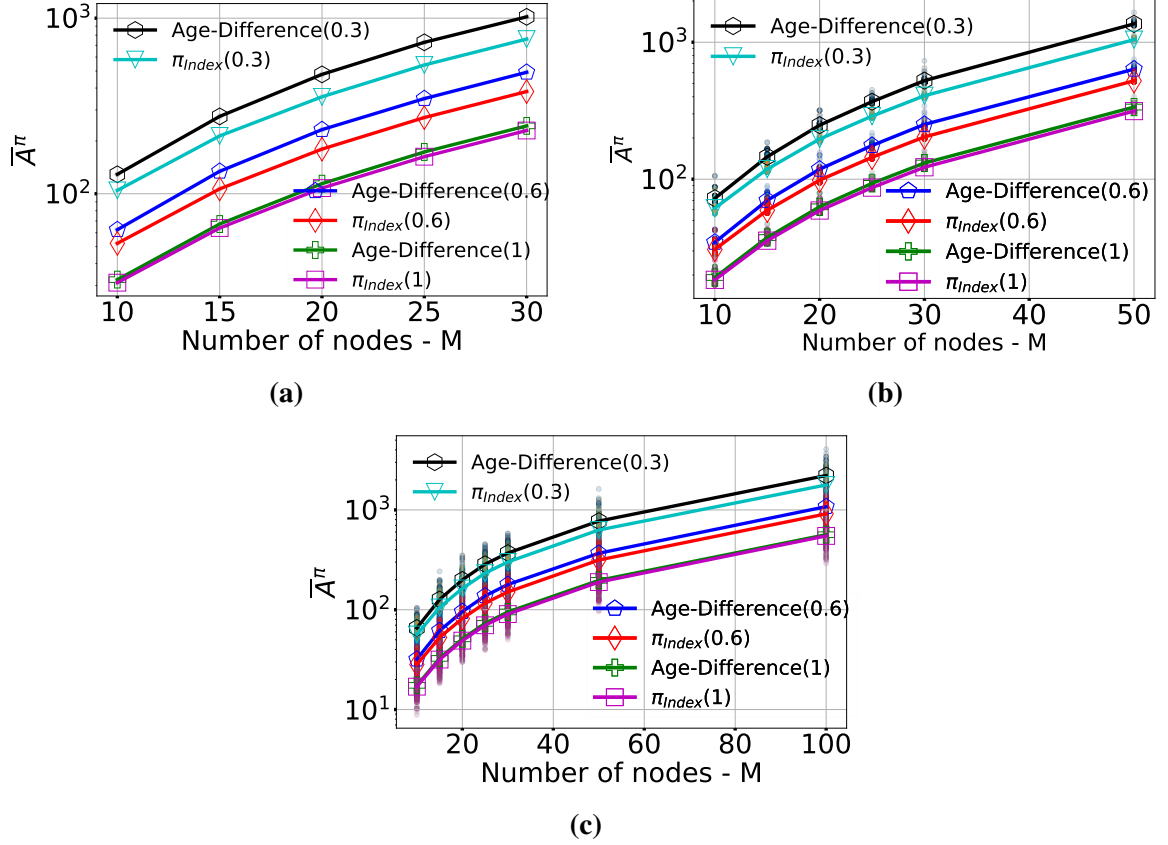


Figure 2.2: Comparison of AAoI: (a) line network, (b) multiline network, and (c) tree network under the Age-Difference(p_s) and $\pi_{\text{Index}}(p_s)$ policies for $p_s \in \{0.3, 0.6, 1\}$ and varying M .

varying values of M , as shown in Fig. 2.2. The results show that the proposed policy π_{Index} outperforms the Age-Difference policy in all cases. In particular, the performance gap increases with increasing M and decreasing p_s , which highlights the effectiveness of π_{Index} . For the line network with $M = 30$, the Age-Difference policy yields a higher AAoI than the π_{Index} policy by 6.25%, 28.75%, and 33.87% at $p_s = 1, 0.6$, and 0.3 , respectively. For the multiline network with $M = 50$, the corresponding AAoI increase is 22.22% at $p_s = 1$, 112.77% at $p_s = 0.6$, and 317.13% at $p_s = 0.3$. Similarly, in the tree network with $M = 100$, the Age-Difference policy exceeds π_{Index} in AAoI by 16.21%, 166.03%, and 382.12% for $p_s = 1, 0.6$, and 0.3 , respectively.

We evaluate the performance of the proposed K-hop extension for π_{Index} and Age-Difference policies for different interference ranges, where $K \in \{1, 2, 3, 4, 5, 6, 7, 8, 10, 15\}$. The results are shown in Figure 2.3(a) for a line network with $M = 20$ and $p_s = 0.5$. They demonstrate that the proposed π_{Index} policy achieves lower AAoI compared to the

Age-Difference policy for $K \geq 6$. Next, we evaluate the performance under a 5-hop interference model while varying the number of source nodes, where $M \in \{10, 15, 20, 25\}$ and $p_s = 0.5$. The results shown in Figure 2.3(b) show that the proposed extension to π_{Index} policy achieves smaller AAoI compared to the Age-Difference policy.

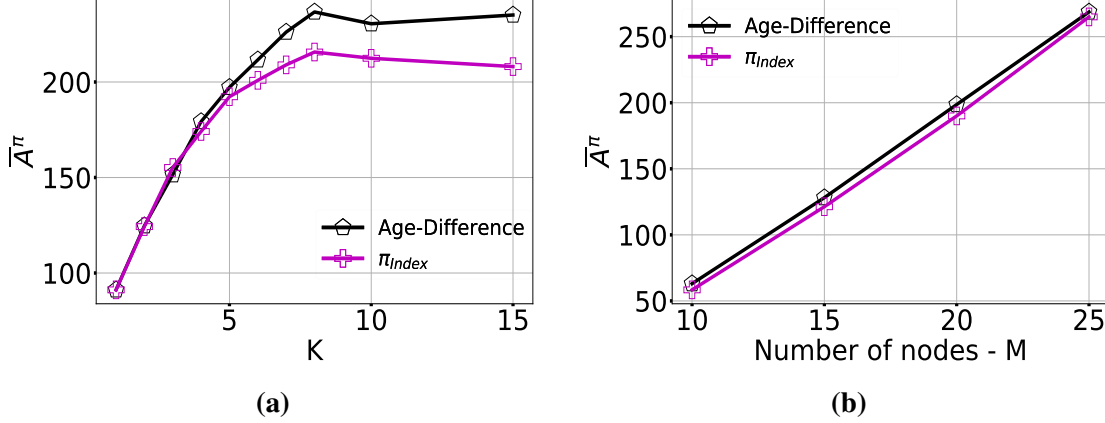


Figure 2.3: Extension to K -hop interference cases: (a) comparison of the simulated AAoI for the proposed π_{Index} and Age-Difference policies as a function of K for a line network with $M = 20$ source nodes and $p_s = 0.5$, (b) comparison of the simulated AAoI for the proposed π_{Index} and Age-Difference policies as a function of M for a line network with $K = 5$ and $p_s = 0.5$.

2.6 Other extensions

2.6.1 Other Network Topologies and Scheduling for Multiple Flows

Heterogeneous network: Additionally, we consider a heterogeneous network scenario in which each link $l \in \mathcal{E}$ has a distinct link success probability $p_{s,l}$, randomly chosen from the range $(0.6, 1)$, under the complete interference model. The results, shown in Fig. 2.4, confirm that the proposed π_{Index} policy outperforms the Age-Difference policy. From the results, it is observed that as M increases, the Age-Difference policy yields a higher AAoI compared to the index policy. Specifically, when $M = 10$, the AAoI under the Age-Difference policy is 8.40% higher than that of the index policy, whereas for $M = 50$, it increases to 15.24%.

Symmetric Tree: We have further analyzed the performance of the proposed index policy, π_{Index} , on fully symmetric tree topologies to more clearly observe the specific impact of increasing hop counts on performance. Accordingly, we conducted additional simu-

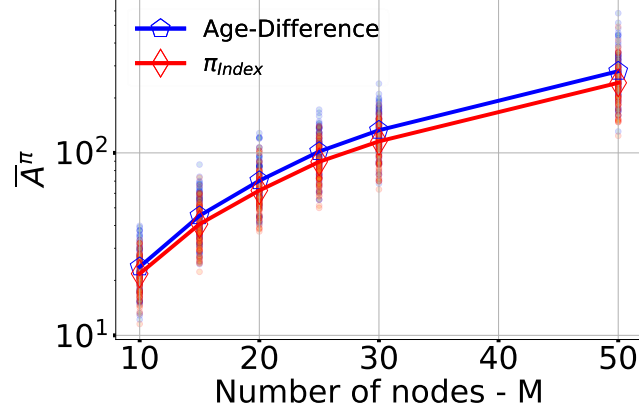


Figure 2.4: Simulated AAoI for a heterogeneous tree network under Age-Difference and π_{Index} policies with complete interference, where each link $l \in \mathcal{E}$ has a link success probability $p_{s,l}$ randomly chosen from the range $(0.6, 1)$.

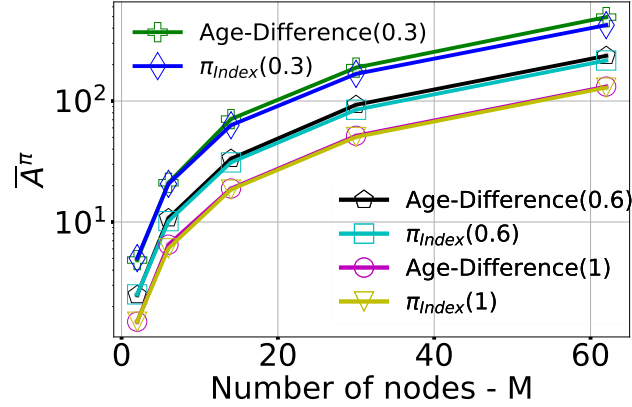


Figure 2.5: Simulation results of the AAoI for a fully symmetric tree network with $M \in \{2, 6, 14, 30, 62\}$ under the Age-Difference(p_s) and $\pi_{\text{Index}}(p_s)$ policies, for $p_s \in \{0.3, 0.6, 1\}$ and different values of M , under complete interference.

lations on fully symmetric tree networks, and the corresponding results are presented in Fig. 2.5. Specifically, we first considered a symmetric tree with two source nodes, 1 and 2, both routed to the sink node 0. We then progressively extended this topology by adding two child nodes, namely 3, 4 and 5, 6, to the existing nodes 1 and 2, respectively. Following the same procedure, the graph was further expanded up to the case of 62 source nodes. Therefore, the number of source nodes considered in our simulations is given by $\{2, 6, 14, 30, 62\}$. The simulation results demonstrate that, for different link success probabilities $p_s \in \{0.3, 0.6, 1\}$, the proposed policy π_{Index} consistently outperforms the Age-Difference [9] policy across all considered network sizes.

Single-Hop Network: A common network topology that has been considered in prior

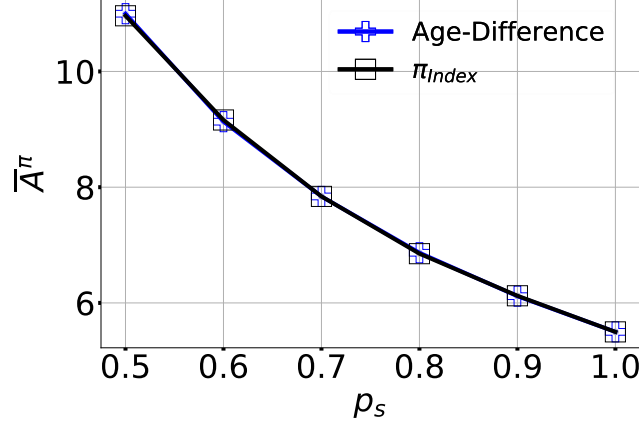


Figure 2.6: Comparison of AAoI for a single-hop network with 10 source-destination pairs for Age-Difference and π_{Index} policies. The AAoI is plotted as a function of the success probability $p_s \in \{0.5, 0.6, \dots, 1\}$ under complete interference.

work is a single-hop network, i.e., there are multiple source destination pairs but they are all separated by a single hop. We compare the performance of our algorithm (which reduces to the Whittle index policy) with Age-Difference in Fig. 2.6. In this case, as noted in prior work [9], Age-Difference and the index policy have similar performance.

2-Hop Network Topology: Another common topology that is especially relevant for complete interference is a 2-hop network, where the source nodes are at most two hops away. Here we consider several cases as illustrated below. For example, Figure 2.7 illustrates the AAoI performance for 2-hop binary tree networks with $M \in \{3, 6, 12\}$ as a function of p_s . The obtained results show that, for $p_s = 0.1$, the percentage improvement achieved by the Index policy (π_{Index}) over the Age-Difference policy is 1.17%, 13.68%, and 11.03% for networks with 3, 6, and 12 source nodes, respectively.

Next, we consider the case of 2-hop binary tree topologies where the link success probability is fixed at $p_s \in \{0.3, 0.5, 0.8\}$, while the number of source nodes M is varied. The results again demonstrate the performance improvement achieved by the proposed π_{Index} policy compared to the Age-Difference policy. As shown in Figure 2.8, the advantage of the proposed π_{Index} policy becomes more evident as the network size increases, particularly when the network becomes more unreliable due to higher link losses (i.e., lower p_s values). The percentage improvement achieved by the π_{Index} policy over the Age-Difference policy is 5.88%, 4.27%, and 3.97% for $p_s = 0.3, 0.5$, and 0.8 , respectively.

Finally, we consider an experiment on an irregular 2-hop tree network. In this setup, a random irregular 2-hop tree topology is generated such that the root node 0 is connected to a randomly selected subset of first-hop nodes, where the number of first-hop nodes m_1

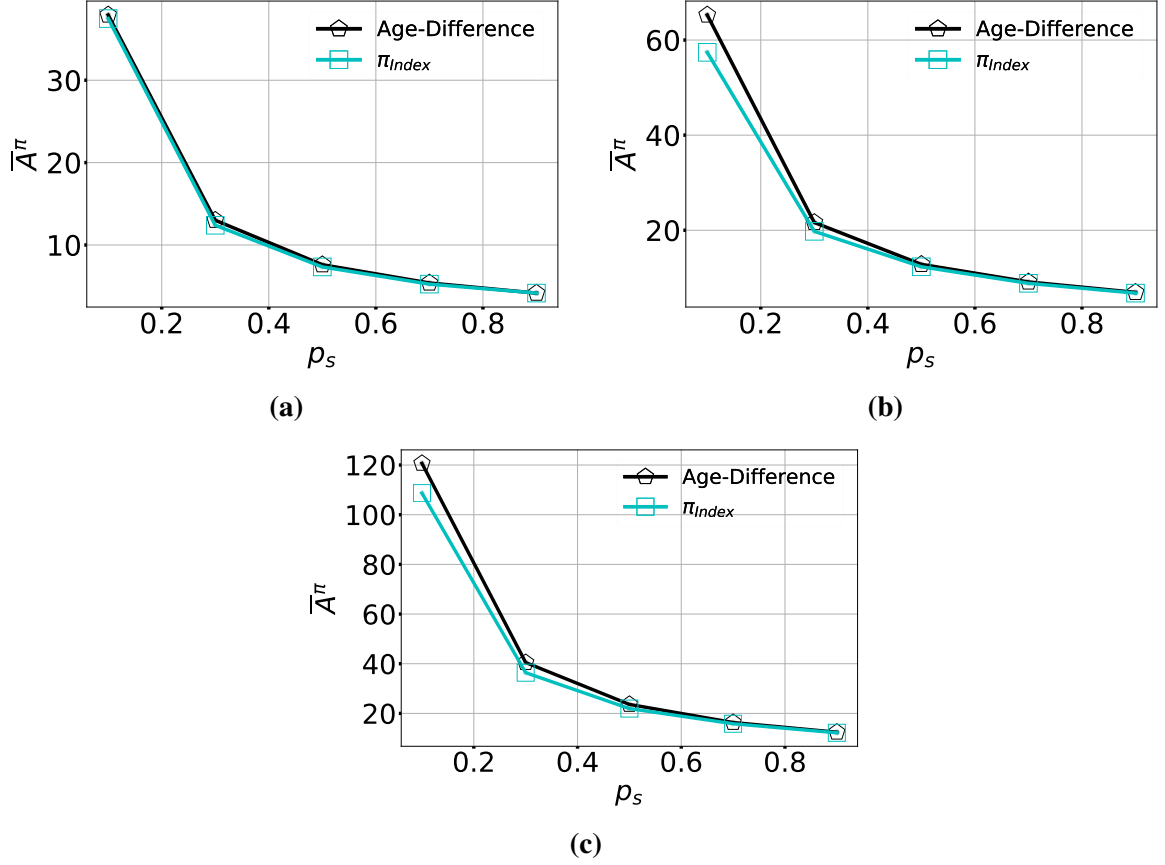


Figure 2.7: Comparison of AAoI for 2-hop binary tree topologies: (a) a 3-source binary tree where node 1 is a child of node 0, and nodes 2 and 3 are children of node 1; (b) a 6-source binary tree where nodes 1 and 2 are children of node 0, and nodes $\{3, 4\}$ and $\{5, 6\}$ are children of nodes 1 and 2, respectively; and (c) a 12-source binary tree where nodes 1, 2, 3, and 4 are children of node 0, and each of these nodes has two child nodes. The results are shown under the Age-Difference and π_{Index} policies under complete interference as a function of link success probability p_s .

is drawn uniformly from $1, \dots, \lfloor M/5 \rfloor$. Each of the remaining $M - m_1$ nodes is independently assigned to one of the first-hop nodes with equal probability, thereby forming the second-hop layer. This construction results in a two-level tree structure in which the root has a relatively small number of children, while the second-hop layer contains a larger number of leaf nodes, leading to an irregular branching structure.

The results obtained for the random irregular 2-hop tree network further demonstrate the effectiveness of the proposed π_{Index} policy compared to the Age-Difference policy. The percentage improvement achieved by the proposed policy is 4.71%, 3.33%, and 1.98% for $p_s = 0.3, 0.5$, and 0.8 , respectively (see Figure 2.9). These comparisons on multiple types of topologies show that the proposed priority index policies have smaller AAoI compared

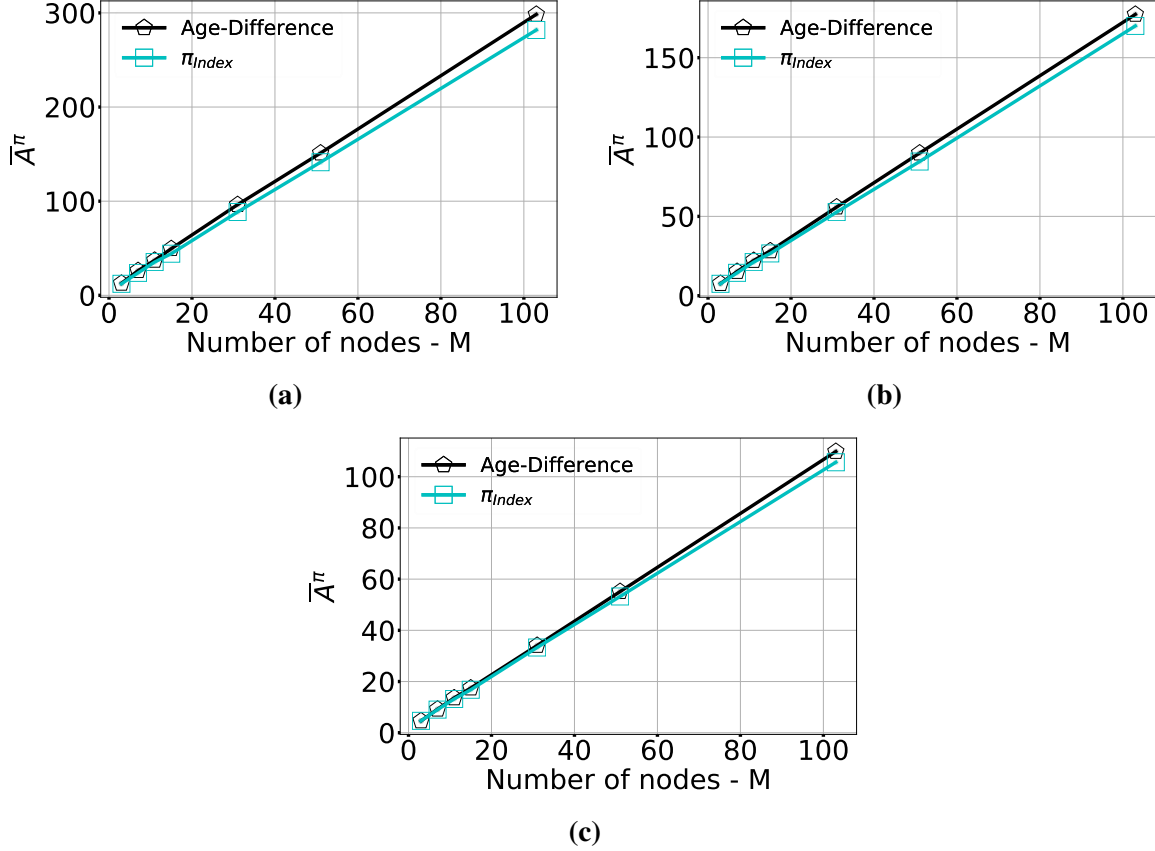


Figure 2.8: Comparison of AAoI for 2-hop binary tree topologies as a function of M , for fixed values of p_s : (a) $p_s = 0.3$, (b) $p_s = 0.5$, and (c) $p_s = 0.8$. The results are shown under the Age-Difference and π_{Index} policies under complete interference.

with policies in prior work.

Multi-Flow Scheduling: Our proposed policy, like Age-Difference, can also be applied for networks with multiple source destination flows. For example, consider the mesh network shown in Fig. 2.10(a). For each destination, a separate routing tree (as shown in blue and red) is established. For complete interference, we compare the AAoI for the Age-Difference policy and the proposed index policy as a function of the success probability p_s in Fig. 2.10(a). We observe that the AAoI for the proposed policy is smaller and the difference increases as the links become less reliable (Fig. 2.10(b)).

2.6.2 Optimization of monotonic and non-monotonic functions of AoI

As discussed, in many applications the performance metric of interest is the average of a general function of the AoI. The function could be monotonic or non-monotonic in AoI

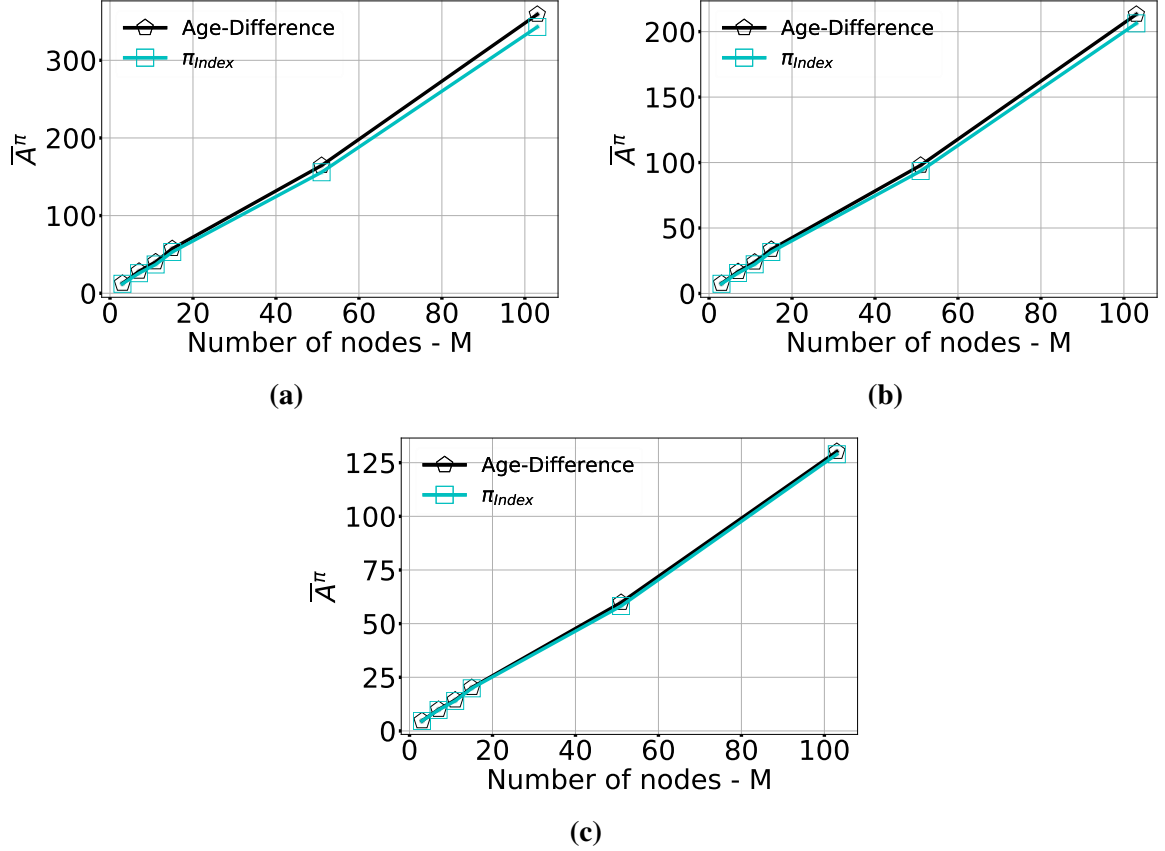


Figure 2.9: A random irregular 2-hop tree topology is generated as follows. The root node (node 0) is connected to a randomly selected subset of first-hop nodes, whose cardinality m_1 is drawn uniformly from $1, \dots, \lfloor M/5 \rfloor$. Each of the remaining $M - m_1$ nodes is independently assigned to one of the first-hop nodes with equal probability, forming the second-hop layer. This results in a two-level tree structure where the root has a small number of children and the second-hop layer contains a larger number of leaf nodes, inducing an irregular branching structure.

[67], [68], [70]. We consider a function f of the AoI and define the average $\overline{F}^\pi(s)$ for a source s at the node 0 under policy π as:

$$\overline{F}^\pi(s) = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[f(A_0^s(t)) | A_0^s(0) = 0] \quad (2.17)$$

where the initial AoI is assumed to be 0. We now define an average for the system under policy π as,

$$\overline{F}^\pi = \frac{1}{|S|} \sum_{s \in S} \overline{F}^\pi(s) \quad (2.18)$$

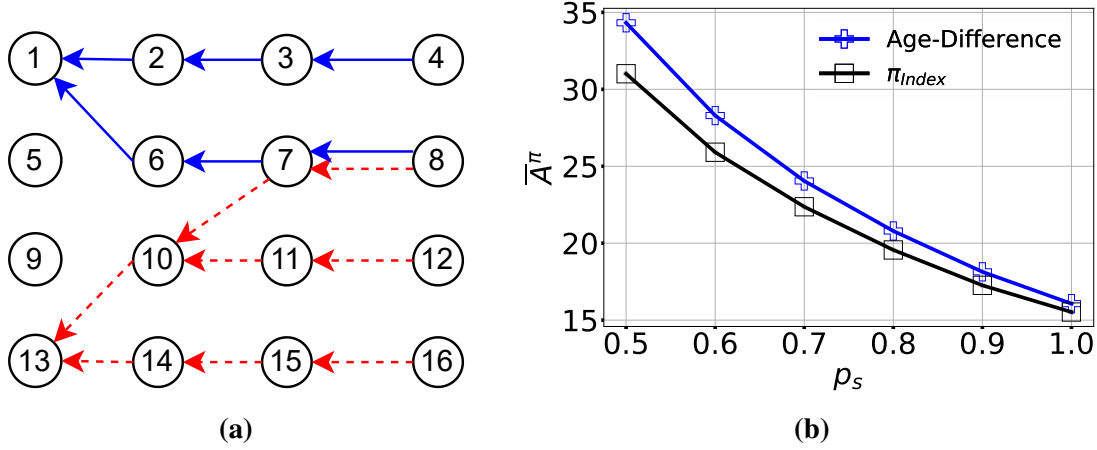


Figure 2.10: (a) Illustration of multiple flows in an example mesh topology graph. The first flow is toward destination node 1 (blue solid thick lines) and the other toward destination node 13 (red dotted lines) (b) Comparison of average AoI for Age-Difference and the proposed index policy as a function of transmission success probability p_s .

The problem that we consider is

$$\min_{\pi \in \Pi} \bar{F}^\pi \quad (2.19)$$

We discuss how the heuristic priority-index policies can be extended to such functions here for the complete interference case and policies $\pi \in \Pi_{1P}$.

First consider the case where threshold policies π_{τ_s} defined on the actual AoI itself are used to obtain priority-index policies. Threshold structures arise naturally in a number of AoI scheduling problems, including problems involving AoI minimization and more general non-decreasing functions of AoI [64], [94]. We again note that these are heuristic priority-index policies and not claimed to be Whittle index policies. The heuristic priority index $\nu(x_s)$ for state $x_s = (a_s, s_s, h_s)$ is defined as:

$$\nu(x_s) = \frac{\bar{F}^{\pi_{a_s+1}} - \bar{F}^{\pi_{a_s}}}{\bar{D}^{\pi_{a_s}} - \bar{D}^{\pi_{a_s+1}}}. \quad (2.20)$$

Here, we note that $\bar{D}^{\pi_{\tau_s}}$ is as in Proposition 2.1 for a policy π_{τ_s} . To obtain $\bar{F}^{\pi_{\tau_s}}$, we modify the derivation of $\bar{A}^{\pi_{\tau_s}}$ in the proof of Proposition 2.1. The Markov renewal process identified in the proof holds since we are considering π_{τ_s} . The average $\bar{F}^{\pi_{\tau_s}}$ can be expressed using MRRT as

$$\bar{F}^{\pi_{\tau_s}} = \frac{\sum_{a=1}^{\infty} \mu_{A,T}(a, t) \mathbb{E}[\mathcal{F}(a, t)]}{\sum_{a=1}^{\infty} \mu_{A,T}(a, t) \mathbb{E}C(a)},$$

where $\mu_{A,T}(a, t)$ is the stationary distribution of (A_m, T_m) and $\mathcal{F}(a, t) = \sum_{\tau=0}^{t-1} f(a + \tau)$. The computation of $\mathbb{E}[\mathcal{F}(a, t)]$ is dependent on the form of the function f and can be computed numerically using the distribution of sum of Geometric random variables. With this definition of priority-indices, the policy π_{Index} can be defined for the network.

Now consider the case as in [67], [68], where the function f is non-monotone. Suppose we consider the scenario of complete interference with policies $\pi \in \Pi_{1P}$. Then, we note that similar to the development in Section 3.3, we can formulate a RMAB but with the objective of minimizing $\bar{F}^\pi$. Again, a Whittle index relaxation leads to SS-MDPs, where for each SS-MDP our objective is to minimize $\bar{F}^\pi + \lambda \bar{D}^\pi$, for policies π for a single source (instead of $\bar{A}^\pi + \lambda \bar{D}^\pi$ in (2.9)). For non-monotonic f , we currently do not have a characterization of optimal policies for SS-MDP. Instead, we suggest a heuristic extension based on [67]. Suppose we consider policies serve a source's packet in contiguous slots until it reaches node 0.¹⁰ We now show that for such policies the evolution of the system is similar to that considered in [67]. Shisher et al. [67] considered non-preemptive scheduling of a source's packet over a channel which introduces IID random delays for different packets. A policy which schedules a source's packets in contiguous slots till it reaches 0 is equivalent to a model in which a source's packet experiences a random delay. The random delay in our case is the random time taken for the packet to traverse h_s hops and reach 0. Shisher et al. considered a non-monotonic function of the AoI as the performance metric. Although, the authors assume a more general *selection from buffer* model, with a buffer size of 1 we recover our model from theirs. Shisher et al. then show that the optimal times at which to transmit packets (or to start scheduling a source in our case) is decided by a threshold policy on a function of the AoI. This function is the Gittin's index of a bandit process with two actions at a slot: (a) stop and transmit at the current slot, or (b) do not transmit and continue to the next slot. If the decision is to stop, then note that the process stops after a random time equal to the channel delay. A multisource extension of this problem is also then considered using a RMAB formulation followed by a Whittle index based relaxation. The Whittle index is obtained in [67, Theorem 5.3] which can then be applied to choose which source is to be scheduled in a slot.

2.6.3 Extension to multi-channel networks

This chapter had considered a single-channel setting to develop the underlying RMAB formulation and the corresponding priority index policy. We note that even under the case

¹⁰We note that since f is non-monotonic, a threshold policy on f may not serve the source in contiguous slots.

of complete interference, we do not have that the optimal solution to the RMAB would lead to an optimal policy for the system. The RMAB formulation is obtained under the additional restriction to policies Π_{1P} that maintain only a single packet per source in the network. This restriction, however leads to implementable policies with good performance. We note that a heuristic index policy can be developed for the scenario where there are multiple orthogonal frequency channels available for the system. Suppose there are C such orthogonal channels available. For each channel, we consider the case where the network has complete interference. Then the priority index policy π_{Index} is defined by first ordering the sources in descending order of $\nu(x_s)$ at time t (ties are broken randomly) and then scheduling the first C sources.

2.7 Conclusions

We proposed priority index based centralized sampling and scheduling policies to minimize the AAoI for multi-hop networks with complete interference and random losses. The priority index is motivated by formulating the problem as an RMAB and then characterizing the index using an age-threshold policy. The proposed policy outperforms the Age-Difference policy [9] across a wide range of link success probabilities p_s . Moreover, it is more practical to implement, as it requires only the AoI of each source at the sink node. The results demonstrate the effectiveness and robustness of π_{Index} under unreliable link conditions, even under the assumption that each source can have at most one packet in the network at any given time.

Furthermore, we compare the AAoI achieved by the proposed π_{Index} policy and the Age-Difference policy in line, multiline, and tree networks with link success probabilities $p_s \in \{0.3, 0.6, 1\}$ for varying numbers of source nodes M . The results show that π_{Index} consistently outperforms the Age-Difference policy, with the performance gap becoming more pronounced as M increases and p_s decreases. In addition, the performance of the proposed policy is evaluated across several other network topologies, including single-hop networks, two-hop tree networks, symmetric tree networks, heterogeneous networks, and multi-flow scheduling networks. In all considered scenarios, the proposed π_{Index} policy achieves a significant reduction in AAoI compared to the Age-Difference policy.

We also investigate the extension of π_{Index} to networks with K -hop interference constraints. Numerical results demonstrate that the proposed policy achieves lower AAoI than the Age-Difference policy across a range of interference levels, with particularly significant gains for $K \geq 6$. In addition, under a 5-hop interference model with varying numbers of

source nodes and link success probability $p_s = 0.5$, the proposed K -hop extension of π_{Index} continues to outperform the Age-Difference policy, yielding consistently lower AAoI.

Appendix

2.A Proof of Proposition 2.1

We consider a system with a single source s at hop distance h^s evolving under π_{τ_s} . We identify a Markov renewal reward process (MRRP) in the evolution to characterize $\bar{A}^{\pi_{\tau_s}}$ and $\bar{D}^{\pi_{\tau_s}}$. The cycles of this MRRP are first identified. Since h^s is finite and the expected time for a successful transmission over a link is $1/p_s$, a packet gets delivered to 0 from s in finite expected time. Suppose the m^{th} packet has been delivered to 0 in slot t . We denote the age after this packet delivery as A_m . For a threshold policy with threshold τ_s , a new packet will be generated and transmitted if $A_m \geq \tau_s$ or after $\tau_s - A_m$ slots if $A_m < \tau_s$. Let r be the parent of s 's at hop distance $h^s - 1$. Since the (s, r) link is lossy, from the time of first generation of a packet, the duration of time taken for a packet (this is the last generated packet which may be different from the first generated packet) to reach 0 is $\sum_{i=1}^{h^s} G_m(i)$, where $G_m(i) \sim \text{Geometric}(p_s)$ represents the random time taken to transmit from i to $i - 1$. The random variables $G_m(i)$ are independent across i . We define renewal cycles as the times between successive packet deliveries to 0. Let $T_m = \sum_{i=1}^{h^s-1} G_m(i)$. Then, the renewal cycles have duration $C_m = \max(\tau_s - A_m, 0) + G_m(h^s) + T_m$, where $G_m(\cdot)$ and T_m are independent of A_m . Since the last generated packet from s needs one slot to be transmitted from s to r , at the end of the m^{th} cycle, we have $A_{m+1} = 1 + T_m$. We obtain a Markov renewal process with $((A_m, T_m))$ and C_m being the associated embedded Markov chain and renewal cycle durations, respectively. To apply MRRT [92], we now identify the expected cumulative AoI and the expected total number of times s is scheduled as costs in a cycle. Suppose we condition on $A_m = a$ and $T_m = t$, and let $C(a, t) = \max(\tau_s - a, 0) + G_m(h^s) + t$. Then the expected cumulative AoI in a cycle is $\mathbb{E}[aC(a, t) + C(a, t)(C(a, t) - 1)/2]$ and the expected total number of times s is scheduled is $\mathbb{E}[G_m(h^s) + t]$. Furthermore, the conditional expected duration of a cycle is $\mathbb{E}C(a, t)$. The expectations are with respect to the distributions of $G_m(\cdot)$. We now discuss how $\bar{A}^{\pi_{\tau_s}}$ can be obtained using MRRT. The derivation of $\bar{D}^{\pi_{\tau_s}}$ is similar. Using MRRT we have that:

$$\bar{A}^{\pi_{\tau_s}} = \frac{\sum_{a=1}^{\infty} \mu_{A,T}(a, t) \mathbb{E}[aC(a, t) + C(a, t)(C(a, t) - 1)/2]}{\sum_{a=1}^{\infty} \mu_{A,T}(a, t) \mathbb{E}C(a, t)}, \quad (2.21)$$

where $\mu_{A,T}(a, t)$ is the stationary distribution of (A_m, T_m) . Since A_m and T_m are independent we have that $\mu_{A,T}(a, t) = \mathbb{P}\{A_m = a\} \mathbb{P}\{T_m = t\}$. We note that $A_m \sim 1 +$

$\sum_{i=1}^{h^s-1} G_{m-1}(i)$ or $1 + T_{m-1}$. We have that $\sum_{i=1}^{h^s-1} G(i)$ is a negative binomial (NB) random variable with $h^s - 1$ successes and parameter p_s . Let $\rho(a) = \mathbb{P}\{A = a\}$. Then, $\rho(a) = \mathbb{P}\{\text{NB} = a - 1\}$, where $\rho(a)$ is $\binom{a-2}{h^s-2} p_s^{h^s-1} q_s^{a-h^s}$ for $h^s \geq 2$. For $h^s = 1$ since $A = 1$ w.p.1, we have $\rho(1) = 1$ and $\rho(a) = 0, \forall a \neq 1$. Now, we simplify the denominator of (2.21). We know $\mathbb{E}[G_m(h^s)] = 1/p_s$ and $\mathbb{E}[T_m] = (h^s - 1)/p_s$. It can be shown that the denominator is $\sum_{a=1}^{\tau_s-1} \rho(a) \left(\tau_s - a + \frac{h^s}{p_s} \right) + \sum_{a=\tau_s}^{\infty} \rho(a) \frac{h^s}{p_s} = \sum_{a=1}^{\tau_s-1} \rho(a) (\tau_s - a) + \frac{h^s}{p_s}$. Similarly, the numerator can be written as a sum of $a \in \{1, \dots, \tau_s - 1\}$ and $a \in \{\tau_s, \dots, \infty\}$ and simplified using the second moment of the sum of geometric random variables (a term $\mathbb{E}[(G_m(h^s) + T_m)^2]$ can be simplified as $\frac{h^s(1-p_s)}{p_s^2} + \frac{(h^s)^2}{p_s^2}$), leading to the final answer.

Chapter 3

Priority-Index Based Sampling and Scheduling Policies for Minimization of Average Age of Incorrect Information in Time-Slotted Multi-hop Networks

3.1 Introduction and Prior Work

The application of remote estimation over wireless networks has gained importance in many use-cases, such as industrial network control [95], video surveillance [96], autonomous driving (unmanned vehicles) [97] and agriculture [98]. In such scenarios, networks are deployed over large areas using low-power devices, resulting in a multihop network topology. Scheduling policies for multihop networks must be carefully designed and optimized based on application-specific, goal-oriented performance metrics. One such metric is the Age of Incorrect Information (AoII) [3, 99], which is particularly relevant for these applications and serves as a useful metric for network optimization. In the literature, prior work on scheduling in wireless networks has primarily focused on information freshness metrics such as Age of Information (AoI) for both single-hop and multihop networks. However, studies on AoII are limited to single-hop networks [52, 53], and, to the best of our knowledge, no work has addressed AoII in multihop networks.

This chapter presents, for the first time, a scheduling policy for minimizing the average AoII (AAoII) in multihop networks. Specifically, we address this gap by designing sam-

*This chapter is based on the manuscript by Nibin Raj, Vineeth B.S., “Design of Centralized Scheduling Policies for Minimizing Average Age of Incorrect Information in Multi-Hop Wireless Networks With Interference,” in IEEE Access, vol. 14, pp. 42956-42974, March 2026, doi: 10.1109/ACCESS.2026.3674431.

pling and scheduling policies to minimize the AAoII in multihop wireless networks. We consider a time-slotted system in which source nodes in the network monitor piecewise-constant, continuous-valued random processes, with changes occurring at random time slots. The source processes are sampled, packetized, and then transmitted to the central node (sink node) over a multihop network with a routing tree topology for remote estimation of each source process. A sampling and scheduling policy controls both the sampling times and the link scheduling for transmitting packets to the central node. We also consider networks with interference, meaning that the link transmission decisions made by the scheduler must satisfy the interference constraints. In this chapter, we restrict our scope to the class of centralized scheduling policies rather than distributed ones. This choice is motivated by practical protocols that employ centralized schedulers, such as IEEE 802.15.4e TSCH (Time-Slotted Channel Hopping) [4] with a Path Computation Element, or WirelessHART [90] with a Network Manager. For ease of implementation and to reduce control signaling overhead, we consider centralized scheduling policies that are implementable at the central node (referred to as *implementable* policies), using limited information obtained from received packets, such as packet timestamps, along with static information about the network topology, interference constraints, and source models. This is in contrast to the Age-Difference policy from prior work [9], which performs scheduling to optimize information freshness in multihop networks by assuming that real-time AoI information at all nodes is available to minimize the average AoI. The AAoII minimization problem for centralized multihop policies can be formulated as a partially observable Markov decision process (POMDP). The problem is partially observable because the centralized scheduler does not directly observe the AoII evolution of the source nodes. Solving such a POMDP is challenging. Therefore, we propose novel, implementable centralized sampling and scheduling policies using an alternative approach. Specifically, we first consider AAoII minimization over a class of policies referred to as genie policies, which have access to additional information. We show that a tractable restless multi-armed bandit (RMAB) formulation can be developed for this class of genie policies, which is then used to derive the corresponding Whittle index policies. Using these genie-based Whittle index policies, we propose implementable policies. The AAoII performance of these policies is evaluated through simulations. Due to the absence of prior policies that minimize AAoII, we compare the AAoII of the proposed implementable policies with other heuristic implementable policies, demonstrating the advantages and utility of our approach. Although Whittle index policies are not directly implementable, they serve as useful baselines for performance comparison.

3.1.1 Prior Work

AoII was initially introduced as a metric for remote estimation in [3], and its utility as a goal-oriented performance measure was further investigated in [99]. Previous studies have primarily considered point-to-point or single-hop communication network settings. AoII minimization for the remote estimation of symmetric binary Markov sources over point-to-point links with transmission delays was investigated in [52, 53], where optimal scheduling policies were derived. An extension to this setting is considered in [54], where the Markov source model is assumed to be unknown. In [55], Chen et al. considered the problem of designing scheduling policies that minimize the AoII for a point-to-point link with random delays, where the scheduler can also decide to preempt the current transmission. AoII minimization subject to power constraints was considered in [56]. We now discuss prior works on AoII in single-hop networks as follows. In slotted ALOHA networks, Nayak et al. considered the design of decentralized policies for minimizing AAoII [57]. The AoII performance of a slotted ALOHA network without feedback was analyzed in [58], assuming that the source processes evolve according to a two-state Markov chain. Tripathi et al. investigated distributed CSMA protocols for minimizing AoI in single-hop networks. In a later extension, the authors considered AoII minimization under a source model comprising independently evolving symmetric two-state Markov chains [59]. In [60], real-time tracking of partially observable Markov sources in a single-hop energy harvesting system is discussed.

We now provide a brief overview of prior works on AoI in multi-hop networks. The AoI of a source node at the central node is defined as the time elapsed since the most recently received update from that source was generated. Several studies have investigated AoI in multi-hop networks under different network settings, including average AoI minimization under stationary randomized policies [42], interference-free multi-hop information flows [43], ring topologies and multi-source time-slotted networks [44, 47], as well as general multi-hop networks and multicast settings [9, 100]. In contrast to these works, we consider the AAoII metric and focus on implementing centralized scheduling policies that utilize only the information available at the central node from the received data packets.

In prior work, RMAB formulations have been used in single-hop networks for designing scheduling policies. For instance, in [101], a time-slotted system is considered where the base station needs to update multiple users simultaneously, with the goal of minimizing the AoII metric under imperfect channel state information. The problem is formulated as an RMAB, and Whittle index-based policies are proposed in [101]. In another work, [102]

proposed a heuristic scheduling policy that minimizes AoII while multiple source nodes send their status updates to a central node. The scheduler does not know the instantaneous state of the sources until the packets reach the central entity. A closed-form Whittle index policy for scheduling in a single-hop multiuser system is proposed to minimize AoII [103]. A detailed summary of prior policies, highlighting the novelty of studying implementable centralized scheduling for minimizing AAoII in multi-hop networks, is provided in Table 3.1.

3.1.2 Contributions

1. The main contribution is the development of two centralized sampling and scheduling policies that minimize the AAoII. These policies are implemented at a central node, which relies only on the information available from the packets received, rather than real-time information from the nodes in the network. Specifically, the policies are designed for multihop networks under complete interference, K -hop interference, as well as a general interference model.
2. A direct formulation of the centralized policy design is shown to be a POMDP, whose solution is challenging, as discussed in Section 3.2. We approach this problem differently by considering policies with additional information, referred to as genie policies, which transform the design problem into a Markov decision process. However, this formulation remains intractable due to the high dimensionality of the associated state space, as also discussed in Section 3.2.
3. A tractable RMAB formulation is proposed for the AoII-minimizing policy design problem by considering genie policies that maintain at most a single packet per source in the network at any time, as discussed in Section 3.3.
4. In Section 3.3, we discuss the indexability of the RMAB and provide a novel analytical derivation of the Whittle indices, which are similar to marginal productivity indices [104]. For the Whittle index derivation, we also obtain analytical characterizations of the AoII under *threshold* policies for multihop networks. These derivations may be of independent interest and are presented in Propositions 3.1 and 3.3.
5. The proposed implementable and centralized policies are obtained from the Whittle index policies.

6. Due to the absence of implementable centralized policies for minimizing AAoII in prior work, we compare the AAoII performance of the proposed policies with heuristic approaches such as randomized and AoI-index-based policies, which also rely only on information from received packets. We show that the proposed policies achieve lower AAoII. For instance, the AoII under the AoI-index-based heuristic is 322.28% of that achieved by the proposed policies for tree topologies with 5-hop interference. Detailed comparisons are presented in Section 3.6.

For the class of genie policies, we compare the performance of the proposed index policies with the Age-Difference policy [9], evaluated in terms of AAoII, and show that the proposed policies achieve lower AAoII. This also demonstrates that the RMAB formulation is a reasonable approach for the AoII policy design problem. A summary of the proposed policies and their information requirements for scheduling is provided in Table 3.1.

Table 3.1: Sampling and scheduling policies in this chapter and comparisons with prior work. The last three policies are implementable at the central node.

Policy	Multi-hop (X or ✓)	Metric used	Comments: Information Required for Scheduler
Sample-at-Change [52]	X	AoII	Source state change and Channel status (via ACK)
Zero-Wait [52]	X	AoII	Only channel availability (ACK), No source state needed
Learning-based scheduling with parameter estimation [54]	X	AoII	Observed feedback (ACK/NACK) History of transmissions and outcomes Online parameter estimation
Preemptive scheduling [55]	X	AoII	Source-change arrivals, Channel status and transmission progress
Threshold-based policy [56, 101, 102]	X	AoII	AoII value, Channel success probability, Source transition probabilities

Continued on next page

Policy	Multi-hop (X or ✓)	Metric used	Comments: Information Required for Scheduler
Decentralized probabilistic transmission policy [57]	X	AoII	AoII value, Channel state or success probability
Semantic-aware sampling and transmission [60]	X	Distortion, AoII	Belief state, Transition model, Observations
Dynamic programming and Index [103]	X	AoII, QAoII	AoII/QAoII of each user, Channel contention model, QoS constraints
Stationary-Randomized [9]	✓	AoI	Network topology, Set of activation sets, Probability distribution over activation sets.
Age-Difference [9]	✓	AoI	Network topology, Set of activation sets, AoI of each source at intermediate nodes.
π_{Index} [This chapter]	✓(Centralized-Genie)	AoII	Network topology, Set of activation sets, Source state change probabilities, AoII of each source.
$\pi_{\text{Index-P}}$ [This chapter]	✓(Centralized-Genie)	AoII	Network topology, Set of activation sets, Source state change probabilities, AoII of each source, Process evolution of each source.
π_{H} (main cont.) [This chapter]	✓(Centralized)	AoII	Network topology, Set of activation sets, Source state change probabilities, AoII estimate of each source at central node.
$\pi_{\text{H-P}}$ (main cont.) [This chapter]	✓(Centralized)	AoII	Network topology, Set of activation sets, Source state change probabilities, AoII estimate, AoI of each source at central node.

Continued on next page

Policy	Multi-hop (X or ✓)	Metric used	Comments: Information Required for Scheduler
π_{H-A} (heuristic) [This chapter]	✓(Centralized)	AoII	Network topology, Set of activation sets, Source state change probabilities, AoI of each source at central node.

Notations: The notations used in this chapter are given in Table 3.2.

Table 3.2: List of symbols and their descriptions used in Chapter 3

Symbol	Description
$[M]$	Set of integers $\{1, 2, \dots, M\}$.
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Directed tree representing the multi-hop network, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of directed edges.
$\mathcal{S}$	Set of source nodes (sensor nodes) that generate packets for the root node.
$ U $	Cardinality of the node set U .
$(i, j) \in \mathcal{E}$	Directed edge from node i to node j , representing a communication link from i to j .
$(s, (i, j))$	Link activation on link (i, j) for source s .
$(s, (s, j))$	Sampling of source s and activation of link (s, j) .
$\mathcal{I}_G$	Interference graph.
K	Parameter defining the K -hop interference model, with $K \in \{1, 2, \dots\}$.
$U(t)$	Activation set at time slot t .
u_k	Element of the activation set $U(t)$ representing a link activation of the form $(s, (i, j))$.
$X^s(t)$	State of the monitored process for source $s \in \mathcal{S}$ at time slot t .
p_c^s (or q_c^s)	Probability that the process $X^s(t)$ transitions (or does not transition) to a new state at the beginning of slot t , where $q_c^s = 1 - p_c^s$.
$f_X^s(x)$	Probability density function used to sample the new state of $X^s(t)$ when a transition occurs.
$G^s(t)$	Generation time of the last successfully received packet from source s at the root node 0.
$X^s(t)$	Actual state of the monitored process for source $s \in \mathcal{S}$ at time slot t .

$\hat{X}^s(t)$	Remote estimate of the process state $X^s(t)$ at the root node 0 for source s .
$A^s(t)$	AoII of source s at the root node 0 at time slot t .
$V^s(t)$	Latest time slot (including t) at which the remote estimate $\hat{X}^s(t)$ of source s was correct.
π	Scheduling policy that determines the evolution of $\hat{X}^s(t)$ and $A^s(t)$.
$\bar{A}^\pi(s)$	Time-averaged AoII of source s under scheduling policy π .
$\bar{A}^\pi$	Average Age of Incorrect Information (AAoII) of the entire system under scheduling policy π , i.e., averaged across all sources.
$H^s(t)$	Hop distance from node 0 to the node currently buffering source s 's packet at time t .
$S^s(t)$	Time elapsed since the currently buffered packet of source s was generated.
$\bar{D}^\pi(s)$	Average fraction of time that source s is scheduled under policy π .
$\bar{D}^\pi$	Sum of $\bar{D}^\pi(s)$ over all sources under policy π .
τ_s	Threshold value associated with source s .
Π_{C-0}	Class of centralized sampling and scheduling policies (centralized at node 0) that use the network information available at node 0.
Π_{C-GA}	Class of centralized-genie policies (C-GA) that, in addition to the information available to Π_{C-0} , also have access to the AoII values $A^s(t)$ for all s .
Π_{C-G}	Class of centralized genie policies (C-G) that, in addition to the information available to Π_{C-GA} , also know the full state history $X^s(t)$ for each source s .
Π_{C-G-1P}	Subset of Π_{C-G} consisting of policies that maintain at most one packet per source in the network.
$\Pi_{C-GA-1P}$	Subset of Π_{C-GA} consisting of policies that maintain at most one packet per source in the network.
Π_{C-A}	Class of centralized policies (C-A) that, in addition to the information available to Π_{C-0} , also have access to the AoI of every source at every node in the network at time t .

3.2 System Model and Problem Statement

We consider the same network model as in Chapter 2. Specifically, the network is modeled as a directed tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of communication links in the routing tree. All other assumptions: with respect to links, the time-slotted nature of the system evolution, the interference model, and the definitions of the sampling and scheduling policy definitions remain the same and are not repeated here.

We now discuss the source and packet generation model in detail. For each source $s \in \mathcal{S}$, the state of the monitored process at time t is denoted by $X^s(t) \in \mathbb{R}$. The process $X^s(t)$ is modeled as a piecewise constant process. For example, $X^s(t)$ may represent the slope in a piecewise linear model used for sensor output modeling [16]. The evolution of $X^s(t)$ is described as follows. At the beginning of time slot t , the process $X^s(t)$ transitions to a new state with probability p_c^s , by independently sampling from a probability density function $f^s_X(x)$. For example, a uniform distribution $U(-m_s, m_s)$, with $m_s > 0$, may be used. Otherwise, the process remains in its current state with probability $q_c^s = 1 - p_c^s$. The evolution of the process $X^s(t)$ at time $t + 1$ is given by,

$$X^s(t+1) = \begin{cases} X^s(t), & \text{w.p. } q_c^s, \\ \sim f^s_X(x), & \text{w.p. } p_c^s. \end{cases} \quad (3.1)$$

In a time slot t , if the schedule includes the link activation $(s, (s, j))$, then the source node s samples the process $X^s(t)$ and generates a packet containing the value $X^s(t)$, along with its generation time slot embedded in it. This packet is then forwarded to node j and subsequently forwarded to the central node over the multihop network for remote estimation. Let $G^s(t)$ denote the generation time of the most recently successfully received packet from source s at node 0. The remote estimate of the process $X^s(t)$ at node 0 is denoted by $\hat{X}^s(t)$. We define the estimate as $\hat{X}^s(t) = X^s(G^s(t))$, i.e., the value carried by the most recently received packet from source s , similar to the definition of the remote estimate in [102]. The main assumptions in the network and source model has been added as a Table 3.3.

We consider AoII as the performance metric, motivated by the remote estimation application. The AoII of a source s at node 0 is defined as a function $A^s(t)$ as follows:

$$A^s(t) = t - V^s(t), \quad (3.2)$$

where $V^s(t)$ denotes the most recent time slot up to and including time t at which the remote

Table 3.3: A summary of the main assumptions used in this chapter

Main Assumptions
1. Multihop network model, Convergecast over a tree routing topology
2. Interference models: Complete, K-hop, and general interference models
3. Scheduler: decision on multiple links that satisfy the K-hop interference constraints which can be activated simultaneously.
4. Source process: Piecewise constant process with random change times.

estimate of source s , $\hat{X}^s(t)$, was correct, i.e., $V^s(t) = \max \left\{ t' : t' \leq t \text{ and } X^s(t') = \hat{X}^s(t') \right\}$. The evolution of $\hat{X}^s(t)$ and $A^s(t)$ depends on the policy π . The AAoII for a source s under policy π is defined as:

$$\overline{A}^\pi(s) = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[A^s(t) | A^s(0) = 0], \quad (3.3)$$

where we assume that the initial AoII $A^s(0)$ is 0. The AAoII for the system under policy π is defined as,

$$\overline{A}^\pi = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \overline{A}^\pi(s) \quad (3.4)$$

Remark 3.1. For a source s , once the AoII $A^s(t)$ is positive, then the AoII becomes zero at a slot $t' > t$ (i.e. $A^s(t') = 0$) if the remote estimate $\hat{X}^s(t')$ and $X^s(t')$ become equal when a packet is received at t' . For other source process models considered in prior work such as discrete finite state Markov chains, $A^s(t)$ may become 0 without a packet reception after being positive if $X^s(t)$ first changes and then returns to a state which is equal to $\hat{X}^s(t)$. For such source processes, since characterizing AoII may be complex, [105] introduced the AoII+ metric which measures the time that has elapsed since the first change. The AoII characterization that we obtain for our simplified source model is the AoII+ for finite state Markov chains (with same probability of leaving any state).

We consider node 0 as a special node that performs both remote estimation and centralized scheduling. We denote by Π_{C-0} the set of centralized sampling and scheduling policies, which have knowledge of the network topology, interference constraints, and source model parameters such as $p_c^s, \forall s \in \mathcal{S}$. Here, $C-0$ indicates that the policies are centralized at node 0. Using the packets received at node 0, a policy $\pi \in \Pi_{C-0}$ is assumed to know the estimates $(\hat{X}^s(\tau), G^s(\tau))$ for all $\tau \in \{0, \dots, t\}$ at the beginning of time slot t . At each time slot t , the policy π selects an activation set $U(t)$, which may be a randomized function of the information available up to time t . A policy in Π_{C-0} is also called *implementable*.

We consider the problem:

$$\min_{\pi \in \Pi_{C-0}} \overline{A}^\pi. \quad (3.5)$$

We note that a policy $\pi \in \Pi_{C-0}$ cannot directly observe $A^s(t)$ for all s . An approach to solve the problem (3.5) is to formulate it as a POMDP. This requires defining a belief distribution $\mathcal{B}^s(a, t)$ for each source s , where $\mathcal{B}^s(a, t)$ represents the belief at node 0 that the AoII is a at the beginning of time slot t . Such a formulation has been adopted in [102] for single-hop networks. The evolution from $\mathcal{B}^s(a, t)$ to $\mathcal{B}^s(a, t+1)$ can be specified as a function of whether source s is scheduled in time slot t , whether the packet is successfully received at node 0, and the process change probabilities, as described in [102]. For instance, in a single-hop network with reliable link transmissions, scheduling source s in time slot t results in the generation and successful transmission of a new packet to node 0. Consequently, $\mathcal{B}^s(0, t+1) = 1 - p_c^s$, which represents the probability that $X^s(t)$ does not change within the slot, so that $\hat{X}^s(t+1) = X^s(t) = X^s(t+1)$. For the case when $a > 0$, $\mathcal{B}^s(a, t+1)$ can be specified as in [102]. However, this approach cannot be directly extended to multihop networks, as it becomes significantly more challenging. In a multihop network, additional state information or a history of past decisions is required to evolve the belief state from one slot to the next.

To illustrate this, note that the evolution of the belief for a source s depends on whether its packet is received or not. However, determining whether a packet from s is present at a node (say i) one hop away from node 0 requires knowledge of past scheduling decisions, specifically whether such a packet has been forwarded from s to i . This necessitates either maintaining the history of scheduling decisions or introducing additional state variables to track packets from source s within the network¹, which leads to a POMDP with a large state space. Based on this observation, the POMDP approach does not appear to be tractable for multihop networks.

Genie Policies: We consider policies in which the scheduler has access to the AoII $A^s(t)$ or the process state $X^s(t)$ for every source s . Such policies are referred to as *genie* policies. Note that these policies are not implementable (i.e., $\notin \Pi_{C-0}$). However, they serve as a useful analytical tool to derive implementable policies in Π_{C-0} .

Two sets of genie policies are defined, each assumed to have access to additional information compared to what is available to Π_{C-0} . Firstly, Π_{C-GA} , where C-GA denotes Centralized-Genie with AoII information. Policies in this set have access to additional in-

¹In networks with unreliable link transmissions, a packet may or may not be successfully forwarded from a node to its parent in $\mathcal{G}$. Therefore, it may also be necessary to consider belief distributions over packet occupancies at nodes.

formation. In each time slot t , these policies are assumed to know: (a) the packets in the node buffers for every node and source in the network, (b) the packet contents, and (c) the AoII $(A^s(t), \forall s)$. Additionally, such policies can maintain the history of packet and AoII evolution up to time slot t . The second class of policies is denoted by Π_{C-G} , where C-G denotes Centralized-Genie. These policies are assumed to have access to additional information such as $(X^s(0), \dots, X^s(t))$, in addition to the information available to Π_{C-GA} . We consider the following:

$$(I) \min_{\pi \in \Pi_{C-GA}} \bar{A}^\pi \text{ and } (II) \min_{\pi \in \Pi_{C-G}} \bar{A}^\pi. \quad (3.6)$$

The above problems can be formulated as Markov decision processes (MDPs). However, as we shall see, these MDPs are also challenging to solve directly.

MDP Formulation: We consider the problems formulated in (3.6)-I and (3.6)-II. In principle, each of these problems can be formulated as a Markov decision process (MDP). For example, consider problem (3.6)-II. The state of the MDP at time slot t consists of the process states $(X^s(t), \forall s)$, the AoII values $(A^s(t), \forall s)$, the buffer states, and the packet contents for every $s \in \mathcal{S}$ and all nodes in the network. The *action space* is the collection of all activation sets that satisfy the interference constraints. The scheduling policy selects an activation set $U(t)$ at each time slot t , possibly as a randomized function of the network state at that time. The single-stage cost of the MDP is given by $\sum_s A^s(t)$. Given the network state and the activation set in time slot t (determined by the scheduling policy π), the state of the network at time $t + 1$ can be obtained. In principle, the solution to this average-cost MDP yields an optimal policy for minimizing AAoII. However, solving this MDP is challenging due to the high dimensionality of the state space. A similar argument applies to the problem in (3.6)-I. In the following, we show that (3.6)-I, II) can be formulated as RMABs by making a further restriction that only a single packet from a source is allowed in the network at any time t . The RMABs are tractable and corresponding Whittle index policies can be obtained. The Whittle index policies are used to obtain implementable scheduling policies for (3.5).

3.3 RMAB Formulations and Whittle Index based Genie Policies

In this section, we define the RMAB formulation and derive Whittle index-based genie policies. We introduce Π_{C-G-1P} , a subset of Π_{C-G} , which consists of genie policies that

maintain at most one packet in the network for each source. Under these policies, a new packet is not generated until the previously generated packet has been delivered from source s to node 0. The reason for this restriction is described as follows. Any policy $\pi \in \Pi_{C-G}$ is a genie policy with knowledge of $(X^s(\tau), \tau \leq t)$. Therefore, if $X^s(t) \neq X^s(t-1)$, a change has occurred, and such a policy π already knows at the beginning of slot t whether a change has occurred. If a change has occurred, there is no benefit in continuing the transmission of the current packet, as it contains *stale* information. Consequently, even if the packet is delivered, it does not help in reducing the AoI $A^s(t)$. Hence the policy $\pi \in \Pi_{C-G-1P}$ immediately drops or pre-empt the current packet from the source s with $X^s(t) \neq X^s(t-1)$, this is defined as *pre-emption on change* property. If the transmission of the current packet from source s is preempted due to a change in $X^s(t)$, then thereafter, in any time slot (including the current slot), if source s is scheduled, a new packet is generated and transmitted from s to its parent node in the tree.

We also define a subset $\Pi_{C-GA-1P} \subset \Pi_{C-GA}$ of policies for which at most a single packet per source is present in the network, motivated by the definition of Π_{C-G-1P} .

3.3.1 RMAB Formulation for (3.6-I)

We first consider the network under a complete interference model. In Section 3.5, we then discuss cases such as K -hop interference as well as a general interference model. In the complete interference case, for $\pi \in \Pi_{C-GA-1P}$, the scheduling decision in each time slot reduces to selecting a source for transmission. Once a source is selected, the link (i, j) is activated, where node i buffers the source's current undelivered packet. For the policies considered, the system evolution can be tracked using the AoI $A^s(t)$ and two additional state variables, $H^s(t)$ and $S^s(t)$, for each source s . The variable $H^s(t)$ is defined as the hop distance from node 0 to node i , where node i buffers the packet of source s in slot t . Since we consider a tree network, there exists a unique path from each source to node 0. Hence, $H^s(t)$ uniquely identifies the node buffering the source's current packet (as each source has at most one packet in the network). Also, $S^s(t)$ is defined as the time elapsed since the current packet from source s was generated, which is useful for determining the evolution of $A^s(t)$. We note that if source s does not have a packet in transmission, then $S^s(t) = 0$ and $H^s(t) = h^s$, where h^s denotes the hop distance of source s from node 0. For the system, we define the state vector $((A^s(t), S^s(t), H^s(t)), \forall s \in \mathcal{S})$. We also introduce an indicator variable $D^s(t)$, which specifies whether source s is scheduled in slot t ($D^s(t) = 1$, active) or not ($D^s(t) = 0$, passive) under a policy π . The evolution of the state vector from

Table 3.4: Evolution of $(A^s(t), S^s(t), H^s(t))$ from t to $t + 1$ for any source s based on $D^s(t)$ (in columns). Different cases based on the current values of $A^s(t)$, $S^s(t)$, and $H^s(t)$ are shown.

	$D^s(t) = 0$	$D^s(t) = 1$
$S^s(t+1)$	$= \begin{cases} 0, \text{ if } H^s(t) = h^s \\ S^s(t) + 1, \text{ o/w} \end{cases}$	$= \begin{cases} 0, \text{ if } H^s(t) = 1 \\ S^s(t) + 1, \text{ o/w} \end{cases}$
$A^s(t+1)$	$= \begin{cases} \begin{cases} 0, \text{ w.p. } q_c \\ 1, \text{ w.p. } p_c \end{cases} \text{ if } A^s(t) = 0 \\ \dots\dots\dots \\ \begin{cases} A^s(t) + 1, \\ \text{if } A^s(t) > 0 \end{cases} \end{cases}$	$= \begin{cases} \begin{cases} 0, \text{ w.p. } q_c \\ 1, \text{ w.p. } p_c \end{cases} \text{ if } A^s(t) = 0 \\ \dots\dots\dots \\ \begin{cases} A^s(t) + 1, \\ \text{if } (H^s(t) > 1 \text{ and } A^s(t) > 0) \\ \text{or} \\ (H^s(t) = 1 \text{ and } 0 < A^s(t) < S^s(t)) \end{cases} \\ \dots\dots\dots \\ \begin{cases} 0, \text{ w.p. } (q_c)^{S_s(t)+1} \\ A^s(t) + 1, \text{ w.p. } 1 - (q_c)^{S_s(t)+1} \end{cases} \text{ if } (H^s(t) = 1 \\ \text{and } A^s(t) \geq S^s(t)) \end{cases}$
$H^s(t+1)$	$= H^s(t)$	$= \begin{cases} H^s(t) - 1, \text{ if } H^s(t) > 1 \\ h^s, \text{ if } H^s(t) = 1 \end{cases}$

time t to $t + 1$, as a function of $D^s(t)$, is shown in Table 3.4.

For a policy $\pi \in \Pi_{C-GA-1P}$, the process $((A^s(t), S^s(t), H^s(t)), \forall s \in \mathcal{S})$ forms a controlled Markov chain (MC). The state space is the Cartesian product $\times_{s \in \mathcal{S}} (\mathbb{Z}_+ \times \mathbb{Z}_+ \times [h^s])$. The action space is $\mathcal{S}$ since the action in every slot is to choose a source s to be scheduled. For a particular state vector $((a_s, s_s, h_s), \forall s)$ we define a single stage cost as $\sum_s a_s$. We note that the problem of AAoII minimization, as defined in (5.2), can be formulated as an average-cost Markov decision process (MDP), with state transitions as described in Table 3.4.

Whittle's Approach: The MDP is a RMAB problem, which is PSPACE-hard [106]. The computational difficulty arises from the *coupling* between the evolution of individual sources due to the hard constraint that only a single source can be scheduled in each time slot. An approach that is useful for such problems is due to Whittle [91], in which the hard constraint that only a single source can be scheduled is relaxed to an average constraint.

To formalize this, let $\overline{D}^\pi(s)$ be the average fraction of time that source s is scheduled under policy π , given by,

$$\overline{D}^\pi(s) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} D^s(t) \quad (3.7)$$

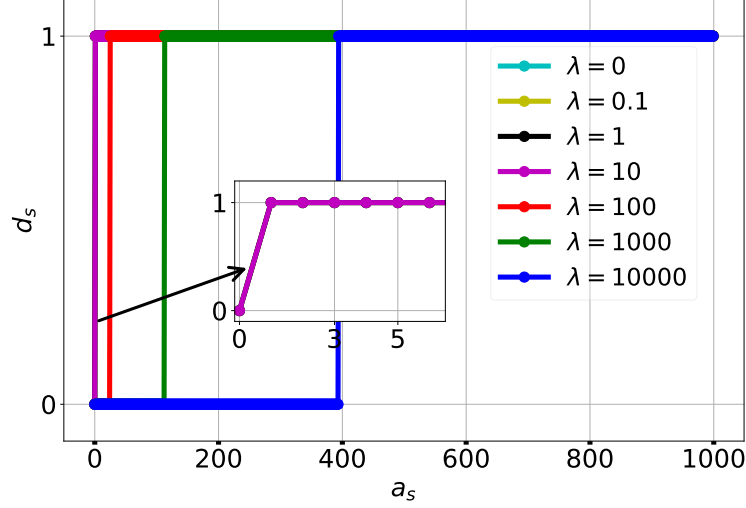


Figure 3.1: Threshold structure of the optimal policy for the MDP for a single source with $h^s = 5$, for different values of λ .

Also, let $\bar{D}^\pi$ be the average fraction of time that system is scheduled under policy π , given by, $\bar{D}^\pi = \sum_s \bar{D}^\pi(s)$. Then, the relaxed problem in Whittle's approach is:

$$\min_{\pi} \bar{A}^\pi \text{ such that } \bar{D}^\pi \leq 1. \quad (3.8)$$

Using a Lagrange multiplier approach, the relaxed problem can be solved as an MDP with average cost $\bar{A}^\pi + \lambda \bar{D}^\pi$, where $\lambda \geq 0$ is a Lagrange multiplier. We denote this MDP as Lagrange-MDP (L-MDP). The single stage cost of L-MDP is given by $\sum_s a_s + \lambda \sum_s d_s$, where $d_s \in \{0, 1\}$ indicates the action of selecting source s and represents a realization of $D^s(t)$. L-MDP decouples to $|\mathcal{S}|$ MDPs, where each MDP corresponds to that of a single source with state $(A^s(t), S^s(t), H^s(t))$ and single stage cost $a_s + \lambda d_s$. We denote the MDP with average cost criterion corresponding to a single source as single-source MDP (SS-MDP). For a source s , the objective of SS-MDP is

$$\min_{\pi} \bar{A}^\pi(s) + \lambda \bar{D}^\pi(s), \quad (3.9)$$

where now π is a policy for a single source that decides $D^s(t)$.

For a source with $h^s = 5$, we illustrate the optimal policies for the SS-MDP in Fig. 3.1. The optimal policies are obtained by solving an equivalent semi-Markov decision process (SMDP) using value iteration [107, Chapter 7]. A detailed discussion is provided in Appendix 3.A.

For a given value of λ , the optimal policy for the SS-MDP is observed to be a *threshold*

policy on age, meaning that the source is scheduled ($D^s(t) = 1$) if $A^s(t) \geq \tau_s$, and not scheduled ($D^s(t) = 0$) otherwise. Here, τ_s is a threshold parameter associated with source s , which depends on s through h^s . In Appendix 3.A, a detailed justification for the threshold property is provided, showing that: (i) any optimal policy for the SS-MDP schedules packets from source s to node 0 in h^s contiguous slots, (ii) this structural result allows the SS-MDP to be reduced to SMDP, and (iii) for the SMDP, threshold policies are optimal. Thus motivated, we proceed under the assumption that the optimal policy for the SS-MDP is a threshold policy. For a threshold policy denoted as π_{τ_s} with threshold τ_s , $\bar{A}^{\pi_{\tau_s}}$ and $\bar{D}^{\pi_{\tau_s}}$ are characterized in the following proposition.²

Proposition 3.1. *The AAOI under the threshold policy π_{τ_s} is given by:*

$$\bar{A}^{\pi_{\tau_s}} = \begin{cases} \frac{0.5p_c q_c^{2h^s} f(h^s, q_c)}{h^s q_c^2} + 0.5h^s q_c^{h^s} - 0.5h^s + h^s q_c^{-h^s} - 0.5q_c^{h^s} \\ \quad + 0.5 + \frac{q_c^{h^s}}{p_c} - \frac{1}{p_c}, & \text{if } \tau_s = 0, \\ \frac{p_c q_c^{-2h^s} (2h^{s^2} - h^s q_c^{h^s} (1+h^s-2\tau_s) + q_c^{2h^s} \tau_s (\tau_s-1))}{2(p_c(h^s q_c^{h^s} + (\tau_s-1)) + 1)}, & \text{if } \tau_s > 0. \end{cases} \quad (3.10)$$

where p_c is the probability that the state process at the source changes, h^s is the hop distance from the source s to the sink, $f(h^s, q_c)$ is a function related to the hop distance and q_c , which is given by $f(h^s, q_c) = \frac{r^{h^s} (h^s + (h^s)^2 (r-1)^2 + 2r - h^s r^2) - 2r}{(r-1)^3 r}$ with $r = \frac{1}{q_c}$. For $\tau_s = 0$, $\bar{D}^{\pi_{\tau_s}} = 1$, while for $\tau_s > 0$, the average fraction of time that source s is scheduled under policy π is $\bar{D}^{\pi_{\tau_s}} = \frac{h^s p_c}{q_c^{h^s} + h^s p_c + (\tau_s-1)p_c q_c^{h^s}}$.

We discuss the proof in Appendix 3.B. Furthermore, a threshold policy with $\tau_s = 0$ is observed to be not optimal for SS-MDP. We first illustrate this for an example in Fig. 3.2b, where we show the $(\bar{D}^{\pi_{\tau_s}}, \bar{A}^{\pi_{\tau_s}})$ achieved for $\tau_s \in \{0, 1, 2, \dots, 450\}$. The value of $\bar{A}^{\pi_{\tau_s}}$ for $\tau_s = 0$ is more than that of $\tau_s = 1$. If $\tau_s = 0$, then the source s is always scheduled, so that $\bar{D}^{\pi_{\tau_s}} = 1$. Furthermore, the behaviour of the policy is periodic, i.e., a packet is generated at $t = 0$ and transmitted hop-by-hop to 0 in h^s slots. Then, again another packet is generated and transmitted to 0 in h^s slots and so on. Intuitively, there are cases where the AoII is 0 when the packet is generated and then the AoII increases during the subsequent h^s slots. The policy with $\tau_s = 1$ generates a packet once the AoII increases from 0, while the policy with $\tau_s = 0$ has to wait for the current transmission of h^s slots to end. Also, since $\bar{D}^{\pi_{\tau_s}}|_{\tau_s=0} \geq \bar{D}^{\pi_{\tau_s}}|_{\tau_s=1}$, a zero-threshold policy is never optimal for SS-MDP.

²We suppress the superscript s in cases where the source s is clear from the context: e.g., p_c instead of p_c^s in the proposition.

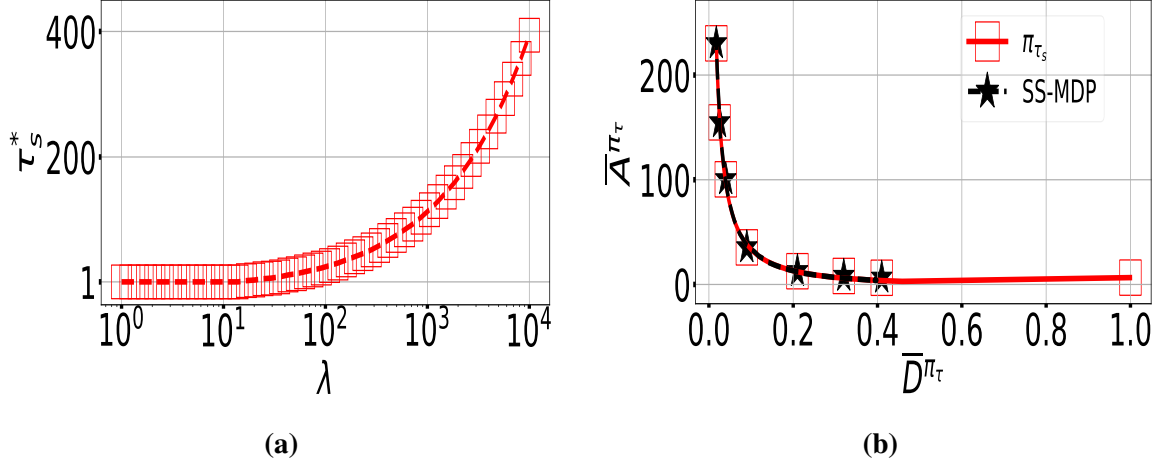


Figure 3.2: (a) Optimal threshold $\tau_s^*(\lambda)$ versus λ for SS-MDP with $h^s = 5$ and $p_c^s = 0.1$. (b) Achievable pairs $(\bar{D}^{\pi_{\tau_s}}, \bar{A}^{\pi_{\tau_s}})$ under threshold policies as $\tau_s \in \{0, 1, \dots, 450\}$. The optimal trade-off obtained from (3.9) using $\lambda_i = 10000^{i/49}$ for $i = 0, \dots, 49$ is also shown.

To obtain a policy for the RMAB, we now discuss indexability using π_{τ_s} . For a given λ , the optimal threshold $\tau_s^*(\lambda)$ is defined as:

$$\arg \min_{\tau_s} \bar{A}^{\pi_{\tau_s}} + \lambda \bar{D}^{\pi_{\tau_s}}, \quad (3.11)$$

where $\bar{A}^{\pi_{\tau_s}}$ and $\bar{D}^{\pi_{\tau_s}}$ are obtained from Proposition 3.1.

Lemma 3.1. *The AAOII $\bar{A}^{\pi_{\tau_s}}$ is monotonically non-decreasing and $\bar{D}^{\pi_{\tau_s}}$ is monotonically non-increasing with τ_s .*

The non-decreasing property of $\bar{A}^{\pi_{\tau_s}}$ can be obtained by considering the terms containing τ_s in the expression from Proposition 3.1. We observe that as $\tau_s \rightarrow \infty$, $\bar{A}^{\pi_{\tau_s}}$ increases linearly with τ_s . Similarly, the non-increasing property of $\bar{D}^{\pi_{\tau_s}}$ can also be inferred from its expression.

Proposition 3.2. *The optimal threshold $\tau_s^*(\lambda)$ is monotonically non-decreasing with λ . Therefore, the RMAB is indexable.*

Proof. The proof proceeds by contradiction. Suppose $\lambda_1 < \lambda_2$ and let $\tau_1 = \tau_s^*(\lambda_1)$ and $\tau_2 = \tau_s^*(\lambda_2)$. Suppose now that $\tau_2 < \tau_1$. From the assumed optimality we have that

$$\begin{aligned} \bar{A}^{\pi_{\tau_1}} + \lambda_1 \bar{D}^{\pi_{\tau_1}} &\leq \bar{A}^{\pi_{\tau_2}} + \lambda_1 \bar{D}^{\pi_{\tau_2}}, \\ \bar{A}^{\pi_{\tau_2}} + \lambda_2 \bar{D}^{\pi_{\tau_2}} &\leq \bar{A}^{\pi_{\tau_1}} + \lambda_2 \bar{D}^{\pi_{\tau_1}}. \end{aligned}$$

Using the monotonicity properties from Lemma 3.1 we then obtain that $\lambda_2 \leq \lambda_1$ which is a contradiction. Therefore, $\tau_s^*(\lambda)$ is monotonically-nondecreasing with λ . Then, λ increases to ∞ , the set of states in which the optimal action is to be passive increases from $\emptyset$ to the whole state space. Therefore, the RMAB is indexable and we proceed to define a policy for the RMAB based on Whittle indices. $\square$

We also illustrate the optimal threshold $\tau_s^*(\lambda)$ as a function of $\lambda \in [0, 10000]$ in Fig. 3.2(a), showing that as λ increases from $-\infty$ to ∞ , the set of states in which the optimal action is to be passive increases from $\emptyset$ to the whole state space.

3.3.2 Whittle Index and Index policy π_{Index}

An instance of the state $(A^s(t), S^s(t), H^s(t))$ for the SS-MDP is denoted as $x_s = (a_s, s_s, h_s)$. The Whittle index $w(x_s)$ for a state x_s is defined as the smallest value of λ for which the optimal action for state x_s is to be passive or 0 [3]. For threshold policies, $w(x_s)$ is defined as the smallest value of λ for which the threshold $a_s + 1$ is optimal. For any $\lambda < w(x_s)$, the optimal action at state a_s is 1. We now show that this smallest value of λ or $w(x_s)$ for which the threshold is $a_s + 1$ can be obtained using the characterization of $\bar{A}^{\pi_{\tau_s}}$ and $\bar{D}^{\pi_{\tau_s}}$ from Proposition 3.1.

Proposition 3.3. *The Whittle index $w(x_s)$ for state $x_s = (a_s, s_s, h_s)$ is*

$$w(x_s) = \frac{\bar{A}^{\pi_{a_s+1}} - \bar{A}^{\pi_{a_s}}}{\bar{D}^{\pi_{a_s}} - \bar{D}^{\pi_{a_s+1}}}. \quad (3.12)$$

Proof. Since the threshold $a_s + 1$ is optimal we have that for any other threshold value τ :

$$\bar{A}^{\pi_{a_s+1}} + \lambda \bar{D}^{\pi_{a_s+1}} \leq \bar{A}^{\pi_{\tau}} + \lambda \bar{D}^{\pi_{\tau}}.$$

This inequality holds only for some values of λ . In order to find the smallest λ , consider $\tau < a_s + 1$. Since $\bar{A}^{\pi_{a_s+1}} \geq \bar{A}^{\pi_{\tau}}$ and $\bar{D}^{\pi_{a_s+1}} \leq \bar{D}^{\pi_{\tau}}$ for $\tau < a_s + 1$ (see Fig. 3.2(b) for an example), we have that any λ for which $a_s + 1$ is optimal should satisfy:

$$\lambda \geq \frac{\bar{A}^{\pi_{a_s+1}} - \bar{A}^{\pi_{\tau}}}{\bar{D}^{\pi_{\tau}} - \bar{D}^{\pi_{a_s+1}}}, \forall \tau < a_s + 1.$$

The smallest value of λ or $w(x_s)$ is the largest value of the RHS in the above inequality. Since the set of optimal achievable points $(\bar{D}^{\pi_{\tau}}, \bar{A}^{\pi_{\tau}})$ is convex (see Fig. 3.2(b) for an

example), the largest value is achieved for $\tau = a_s$. Then, we have that the smallest value of λ or the Whittle index is

$$w(x_s) = \frac{\overline{A}^{\pi_{a_s+1}} - \overline{A}^{\pi_{a_s}}}{\overline{D}^{\pi_{a_s}} - \overline{D}^{\pi_{a_s+1}}}.$$

□

Remark 3.2. We note that the above characterization of the Whittle index is the same as the Marginal Passivity Index proposed by Niño-Mora [104] as well as the index in [108]. We note that our characterization of the index does not require finiteness of the state space, and a nested active set structure obtained by adding single elements. However, our characterization is specific to the problem considered and depends on the nature of the threshold policy.

We propose a baseline policy π_{Index} for the RMAB problem using the Whittle index computed in Proposition 3.3. In slot t , for each source with state x_s , the Whittle index $w(x_s)$ is obtained. The π_{Index} policy schedules that source s with the maximum Whittle index. We note that $\pi_{\text{Index}} \in \Pi_{C-GA}$. The Whittle index policy is a heuristic for the original MDP. Its performance has been shown to be optimal in the asymptotic regime where the number of SS-MDPs goes to infinite (e.g. for AoI in [109]). Our simulations show that our proposed Whittle index based π_{Index} policy, which is a genie policy, motivates the design of implementable policies as well as serving as a useful baseline.

3.3.3 RMAB Formulation for (3.6-II)

For policies in Π_{C-G-1P} we can again show that (3.6-II) is an RMAB. The evolution of the system state, which includes the AoII $A^s(t)$, $\forall s$ and the RMAB formulation, is discussed in detail in Appendix 3.C. Again, using the approach by Whittle, we consider a Lagrangian relaxation, which decouples the RMAB to individual MDPs for single sources. These MDPs are similar to SS-MDPs in Section 3.3. We refer to these MDPs as SS-MDPs in the following. For this SS-MDP, in Appendix 3.C, we show numerically that an optimal policy is a threshold policy on the AoII $A^s(t)$. The threshold policy $\pi_{\tau_s} \in \Pi_{C-G-1P}$ ³ in a slot t , pre-empt^s s 's packet in transit if $X^s(t) \neq X^s(t-1)$ and then schedules s if $A^s(t) \geq \tau_s$. We obtain the following result (similar to Proposition 3.1) for a threshold policy.

³We abuse notation here: Π_{C-G-1P} is a class of genie scheduling policies for a single source.

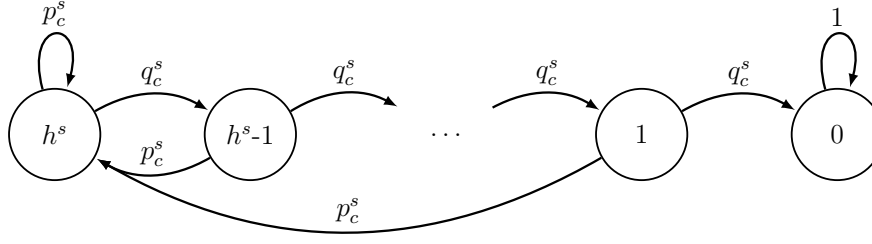


Figure 3.3: Transition diagram of the Markov chain modeling the evolution of the hop distance of a packet from source s when scheduled in contiguous slots. The process starts at h^s , and state 0 denotes successful delivery. With probability p_c^s , a packet is preempted and a new packet starts at h^s ; otherwise, the hop distance decreases by one with probability $q_c^s = 1 - p_c^s$.

Proposition 3.4. *The AAoII of policy π_{τ_s} with pre-emption on change is:*

$$\overline{A}^{\pi_{\tau_s}} = \begin{cases} \frac{1}{2} \frac{\mathbb{E}[T(h^s)] + \mathbb{E}[T(h^s)^2]}{\frac{1}{p_c} + \mathbb{E}[T(h^s)]}, & \text{if } \tau_s = 0, \\ \frac{\frac{\tau_s(\tau_s-1)}{2} + \tau_s \mathbb{E}[T(h^s)] + \mathbb{E}\left[\frac{T(h^s)(T(h^s)-1)}{2}\right]}{\frac{1}{p_c} + (\tau_s-1) + \mathbb{E}[T(h^s)]}, & \text{if } \tau_s > 0. \end{cases} \quad (3.13)$$

where p_c is the probability that the state process at the source changes, and h^s is the hop distance from the source s to 0. The random variable $T(h^s)$ is the random time taken to transmit a packet from s to 0 in contiguous slots with pre-emption on change ($T(h^s)$ is characterized as the first hitting time to state 0 for the Markov chain in Fig. 3.3 with initial state h^s). For $\tau_s = 0$, $\overline{D}^{\pi_{\tau_s}} = 1$, while for $\tau_s > 0$, $\overline{D}^{\pi_{\tau_s}} = \frac{\mathbb{E}[T(h^s)]}{\frac{1}{p_c} + (\tau_s-1) + \mathbb{E}[T(h^s)]}$.

We discuss the proof in Appendix 3.D and proceed under the assumption that an optimal policy is a threshold policy. Indexability for the RMAB is shown as in Section 3.3 using threshold policies.

3.3.4 Index Policy $\pi_{\text{Index-P}}$

We propose an index policy $\pi_{\text{Index-P}} \in \Pi_{C-G-1P}$. The Whittle index is computed as:

$$\frac{\overline{A}^{\pi_{a_s+1}} - \overline{A}^{\pi_{a_s}}}{\overline{D}^{\pi_{a_s}} - \overline{D}^{\pi_{a_s+1}}}, \quad (3.14)$$

for a source with AoII a_s , and $\overline{A}^{\pi_{\tau_s}}$ and $\overline{D}^{\pi_{\tau_s}}$ are computed using Proposition 3.4. The $\pi_{\text{Index-P}}$ policy schedules that source s with the maximum Whittle index (computed using (3.14)) in slot t . The proof follows the same arguments as in Proposition 3.3.

3.3.5 The Age-Difference Policy

We also define a class of centralized policies Π_{C-A} , where C-A denotes Centralized with AoI information. These policies have access to the AoI of every source at each node in the network at slot t , in addition to the information available to Π_{C-0} . For implementation, this requires knowledge of the differences in the AoI of a source between the transmitters and receivers of links. The Age-Difference policy [9] is an element of Π_{C-A} . Although the Age-Difference policy was proposed for AAoI minimization, we evaluate the AAoI performance of this policy in the following.

3.4 Implementable Centralized Sampling and Scheduling Policies

In this section, we present two heuristic policies, π_H and π_{H-A} , belonging to the class Π_{C-0} . The policies are based on the index policies π_{Index} and $\pi_{\text{Index-P}}$.

3.4.1 Policy $\pi_H \in \Pi_{C-0}$

The policy π_H is designed such that it does not require exact knowledge of $A^s(t)$ at the scheduler. Instead, the scheduler maintains an estimate, denoted by $\hat{A}^s(t)$, of the expected AoII, i.e., $\mathbb{E}[A^s(t)]$, for all s , at the root node. The scheduler is assumed to know the source model parameter p_c^s (or equivalently $q_c^s = 1 - p_c^s$).

The estimate is computed as the expected AoII over the possible realizations of the source process. Let a denote the AoII of source s at the generation time of its last successfully received packet⁴. The value of a , together with the source parameter p_c^s and the elapsed time Δt since the last update, is used to compute $\hat{A}^s(t+1)$. The expected AoII at time $t+1$ is then given by:

$$\hat{A}^s(t+1) = \mathbb{E}[A^s(t+1) \mid A^s(t_g^s) = a, \Delta t, p_c^s].$$

where t_g^s is the generation time of the last received update from source s . The expectation

⁴In the protocol considered in this chapter, the source is aware of the scheduling decisions made by the scheduler in every slot. Therefore, the source can compute its remote estimate at node 0 and its own AoII. At each packet generation time, the AoII is embedded in the packet transmitted to node 0. An alternative policy that does not require such computation, but results in a larger AAoII, is discussed in Appendix 3.E.

can be computed as

$$\mathbb{E}[A^s(t+1) \mid A^s(t_g^s) = a, \Delta t, p_c^s] = \begin{cases} \sum_{x=1}^{\Delta t} x (q_c^s)^{\Delta t-x} p_c^s, & \text{if } a = 0, \\ (a + \Delta t) [1 - (q_c^s)^{\Delta t}], & \text{if } a > 0. \end{cases}$$

Thus, the scheduler updates AoII estimate $\hat{A}^s(t)$ for each source s based on the most recently received packet and the elapsed time. We propose heuristic policy π_H as follows. For every source s , π_H computes the Whittle index by substituting $a_s = \hat{A}^s(t)$ (rounded to the nearest integer) into (3.12), instead of using $a_s = A^s(t)$. The scheduler then selects the source with the maximum Whittle index.

To implement π_H , in each slot the central node (which can be assumed to have higher transmit power) broadcasts a control packet from node 0 to all nodes, indicating the identity of the scheduled source (say s). We also assume that packets are assigned sequence numbers; accordingly, the control packet includes the sequence number of the packet to be scheduled. The initial sequence numbers are assumed to be synchronized between the scheduler and each source s . A node that has a packet from source s transmits it. If the sequence number is *new*, a packet with that sequence number is generated at the source and transmitted. The source does not generate a new packet for a sequence number for which it has already generated a packet.

3.4.2 Policy $\pi_{H-P} \in \Pi_{C-0}$

We now propose a heuristic policy $\pi_{H-P} \in \Pi_{C-0}$ based on π_H . We assume that π_{H-P} has knowledge of the source-specific model parameters $p^s c$ (or equivalently $q^s c = 1 - p^s c$). Similar to π_H , the policy π_{H-P} maintains an AoII estimate $\hat{A}^s(t)$ for each source. The definition of $\hat{A}^s(t)$ and its update rule remain the same as those used in π_H . In each slot, π_{H-P} first randomly preempts the packet of source s with probability $p^s e$, independently across slots. These probabilities are policy parameters and are heuristically chosen as $2/p^s c$. We also define another policy belonging to the class Π_{C-A} . For each slot t and source s , the policy computes the delay $d_s = t - t_g^s$, where t_g^s denotes the packet generation time of the current packet in transit for source s . If $d_s > \tau_d^s$, the source process is assumed to have changed, and the policy preempts the current packet. Here, $\tau^s d$, for all s , are delay-threshold parameters, with a heuristic choice given by $\tau^s d = 2/p^s c$. The requirement of knowledge of t_g^s makes this policy an element of Π_{C-A} rather than Π_{C-0} . The Whittle index is then computed by substituting $a_s = \hat{A}^s(t)$ (rounded to the nearest integer) into (3.14), instead

of using $a_s = A^s(t)$. The policy $\pi_{\text{H-P}}$ then schedules the source with the maximum Whittle index.

3.4.3 Policy $\pi_{\text{H-A}} \in \Pi_{C-0}$

We also propose $\pi_{\text{H-A}}$, a heuristic policy similar to π_{H} , in which the AoI of each source at node 0 is used in place of $\hat{A}^s(t)$ for a_s in (3.12) when computing the Whittle index. The AoI of each source at node 0 is obtained from the generation times of the received packets. In each slot, $\pi_{\text{H-A}}$ schedules the source s with the maximum Whittle index.

3.4.4 Complexity of Scheduling Policies

For π_{Index} , π_{H} , and $\pi_{\text{H-A}}$, the decision in each slot requires $\mathcal{O}(M)$ operations. In contrast, both $\pi_{\text{Index-P}}$ and $\pi_{\text{H-P}}$ require $\mathcal{O}(Mh_m^3)$ computational complexity, where $h_m = \max_s h^s$, to determine the source to be scheduled. This is because a system of linear equations must be solved to compute $\mathbb{E}[T(h^s)]$ and $\mathbb{E}[T(h^s)^2]$ in Proposition 3.4, which are required to evaluate the indices.

3.5 Implementable Policies for Networks with K-Hop and General Interference Models

We first consider the K -hop interference model, in which two links can transmit concurrently in a slot if their hop distance is at least K . We then discuss how the proposed π_{Index} and π_{H} policies can be extended to this setting (similar extensions apply to $\pi_{\text{Index-P}}$ and $\pi_{\text{H-P}}$). Under this model, the policies can schedule multiple links concurrently, provided that the corresponding sources satisfy the K -hop interference constraints. We discuss how these multiple links are determined in the following. For any source s , there exists a sequence of links (i.e., a path) through which packets are transmitted to node 0. For example, in the multiline network shown in Fig. 3.16(a), for node 7, the path is $7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 0$. First, we consider π_{Index} . In time slot t , we compute $w_s(t)$ for all sources using (3.12) and Proposition 3.3. Based on $H^s(t)$ and the path corresponding to source s , the transmitting link $L_s(t)$ is determined. Let $\mathcal{L}(t) = \{L_s(t), \forall s\}$ denote the set of all links that can transmit packets for all sources in slot t . For every link $l \in \mathcal{L}(t)$, we also define

$$s_l(t) = \arg \max_s \{w_s(t) : L_s(t) = l\}. \quad (3.15)$$

That is, for each link l that can transmit packets from multiple sources, we select the source with the highest index. If multiple sources share the highest index, one of them is chosen uniformly at random. Only the packet corresponding to the source $s_l(t)$ with the highest index is transmitted if link l is scheduled. To determine the set of links that can be scheduled concurrently, we consider the collection of all maximal independent sets $\mathcal{I}$ of links.

The set of all independent sets is obtained from the interference graph of the tree under the K -hop interference constraint. An auxiliary graph $\mathcal{H}$ is constructed, in which each edge of the original graph $\mathcal{G}$ is represented as a node. An interference matrix is then computed, where the entry at position (i, j) is set to 1 if the corresponding edges interfere, based on their K -hop neighborhood in $\mathcal{H}$. Using this matrix, the interference graph is formed, where each node corresponds to an edge of $\mathcal{G}$, and an edge exists between two nodes if their corresponding edges interfere. The maximal independent sets [110] of this interference graph are then identified. We note that, for a general interference model, the same approach can be applied with $\mathcal{I}$ defined accordingly. We choose

$$i^*(t) = \arg \max_{i \in \mathcal{I}} \sum_{l \in i} w_{s_l(t)}(t), \quad (3.16)$$

We assume that $w_{s_l(t)} = 0$ for any link l that does not carry a packet from any source. The selected set $i(t)$ is the independent set that maximizes the sum of the indices of the corresponding sources. If multiple such independent sets exist, one is chosen uniformly at random. The policy π_{Index} is thus defined as the policy that schedules the set of sources $\{s_l(t), \forall l \in i(t)\}$. Similarly, the policy $\pi_{\text{Index-P}} \in \Pi_{C-G-1P}$ preempts the packet of source s and then schedules the set of sources $\{s_l(t), \forall l \in i(t)\}$. Note that, to obtain $i(t)$, the index $w_s(t)$ is computed using (3.14). To derive implementable policies, we again consider the AoII estimates $\hat{A}^s(t)$. Since the links are lossless, the central node can determine $L_s(t)$ for all s from the history of scheduling decisions. These AoII estimates are then used to compute $w_s(t)$, and subsequently $s_l(t)$.

3.6 Simulations and Performance Analysis

Setup for Simulations: The comparisons are conducted across different network topologies and source parameters. The generation of the network topologies is described as follows. We consider *line*, *multiline*, and *tree* networks for varying network sizes M , with $\mathcal{S} = [M]$. A line network is a linear graph with nodes indexed from 0 to M , where each node i transmits to node $i - 1$ for $i \in 1, 2, \dots, M$. A tree network consists of *top-subtrees*,

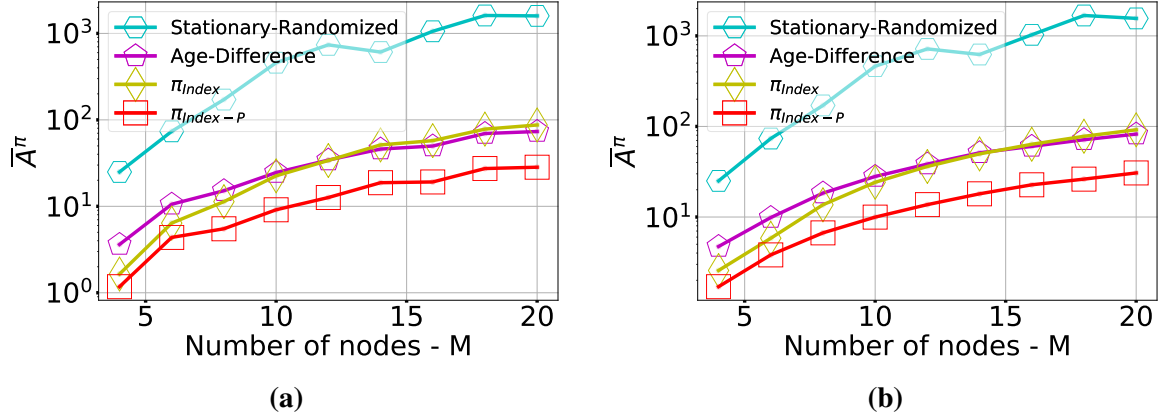


Figure 3.4: Simulated AAOI for line networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under K -hop interference ($K = 5$).

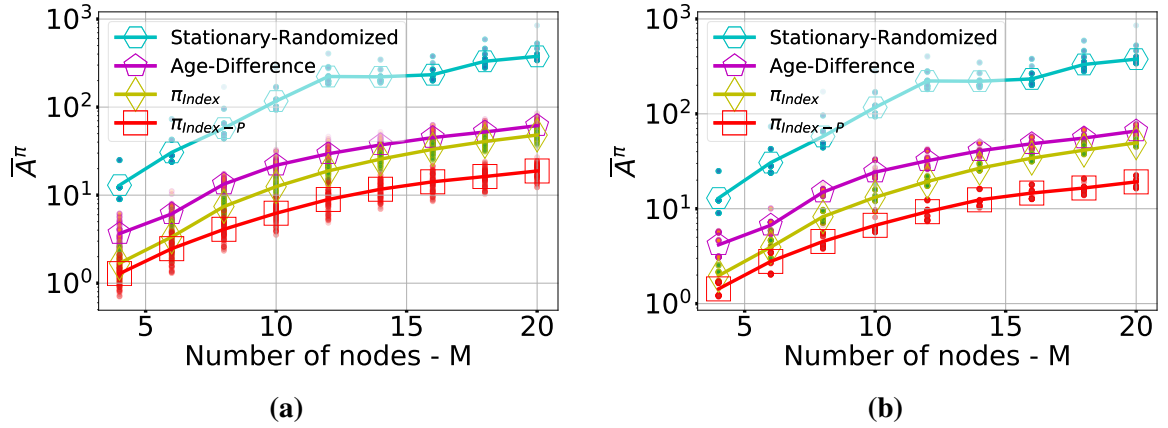


Figure 3.5: Simulated AAOI for multiline networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under K -hop interference ($K = 5$).

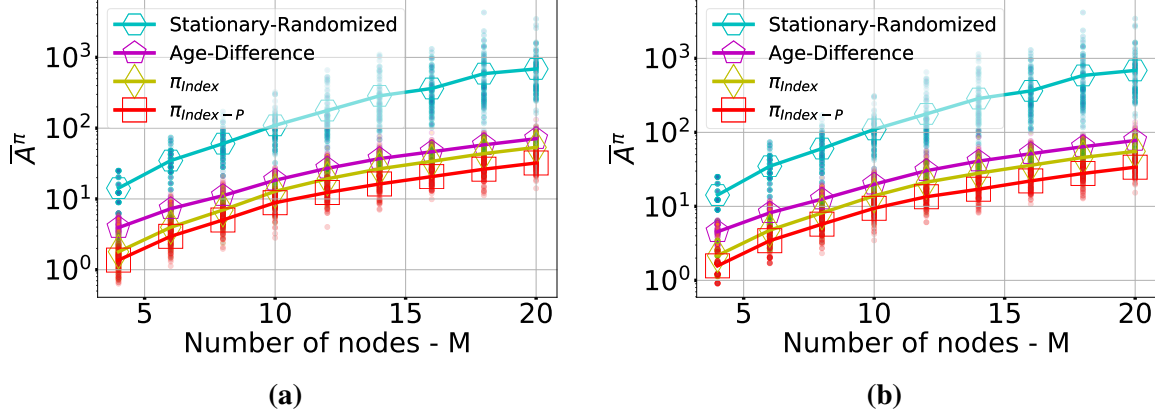


Figure 3.6: Simulated AAOII for tree networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under K -hop interference ($K = 5$).

where each top-subtree is defined as a subtree whose root is a child of node 0. A multiline network is a tree with b top-subtrees, where each top-subtree has a line network structure. For a multiline network, let n_i denote the number of nodes in the i^{th} top-subtree for all $i \in [b]$. The values n_i are sampled from a multinomial distribution with a total of M trials, b mutually exclusive outcomes, and uniform probability for each outcome, ensuring that $\sum_{i=1}^b n_i = M$. In the experiments, we set $b = 2$ and generate 100 random graphs for each value of M . In the experiments, we set $b = 2$ and generate 100 random graphs for each M . Tree topologies are generated by uniformly sampling from all possible trees with M nodes using a Prüfer sequence-based construction method [93], with 100 random graphs generated for every M .

We consider *homogeneous* networks, where each source $s \in \mathcal{S}$ has $p_c^s = p_c \in (0, 1)$, as well as *heterogeneous* networks, where each source s has a parameter $p_c^s \in (0, 1)$. For the homogeneous case, a single p_c is chosen uniformly at random for each network from the range $\left[\frac{1}{3hm}, \frac{1}{2hm}\right]$ for all sources, where $h_m = \max_s h^s$ is the maximum hop distance from the sink node 0. In the heterogeneous case, p_c^s is assigned based on the hop distance h^s : for sources with hop distances less than $hm/2$, $p_c^s = \frac{1}{3hm}$; otherwise, $p_c^s = \frac{1}{2hm}$.

3.6.1 Performance of Whittle Index-Based Policies

In this section, we evaluate the AAOII performance of the Whittle index based baseline policies (π_{Index} and $\pi_{\text{Index}-P}$). Since these policies have access to additional information compared to policies in Π_{C-0} , their AAOII performance intuitively provides a lower bound on the AAOII achievable by policies in Π_{C-0} . We also evaluate the AAOII performance of

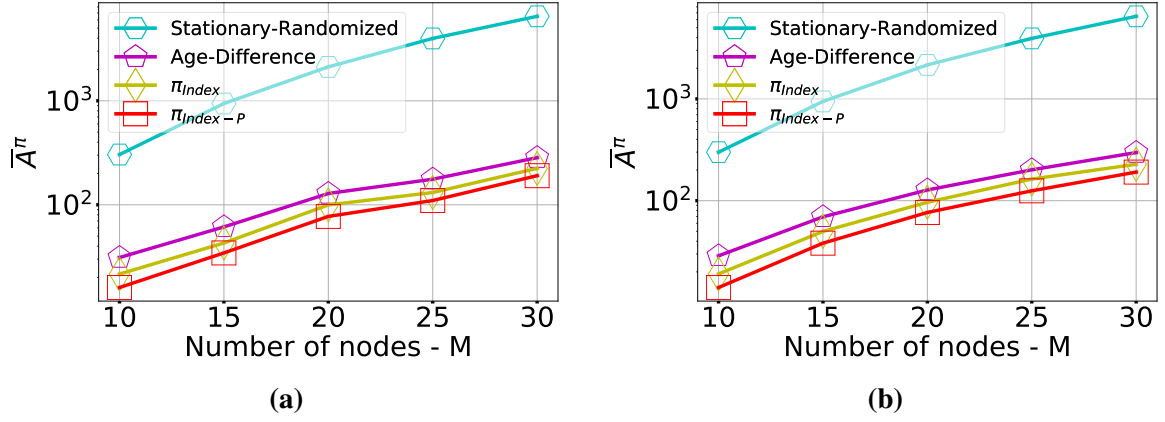


Figure 3.7: Simulated AAOI for line networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under complete interference.

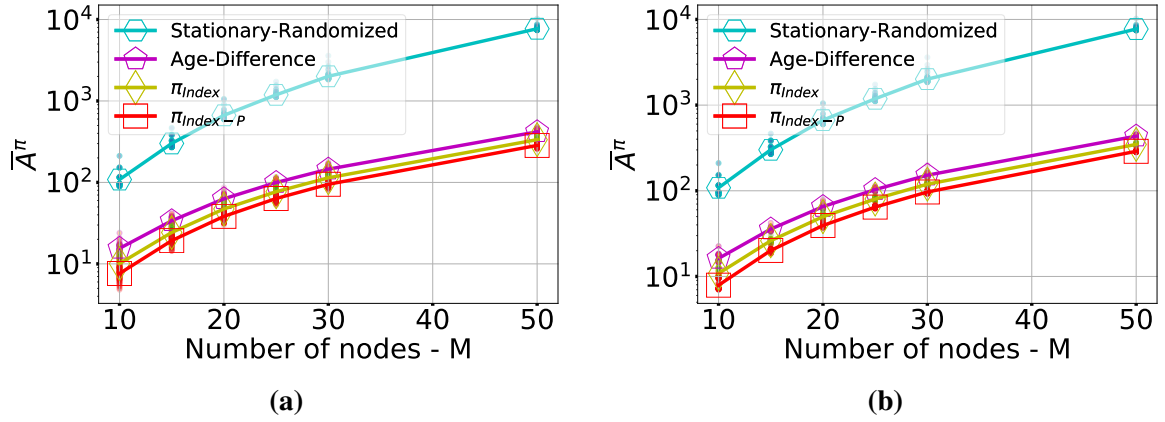


Figure 3.8: Simulated AAOI for multiline networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under complete interference.

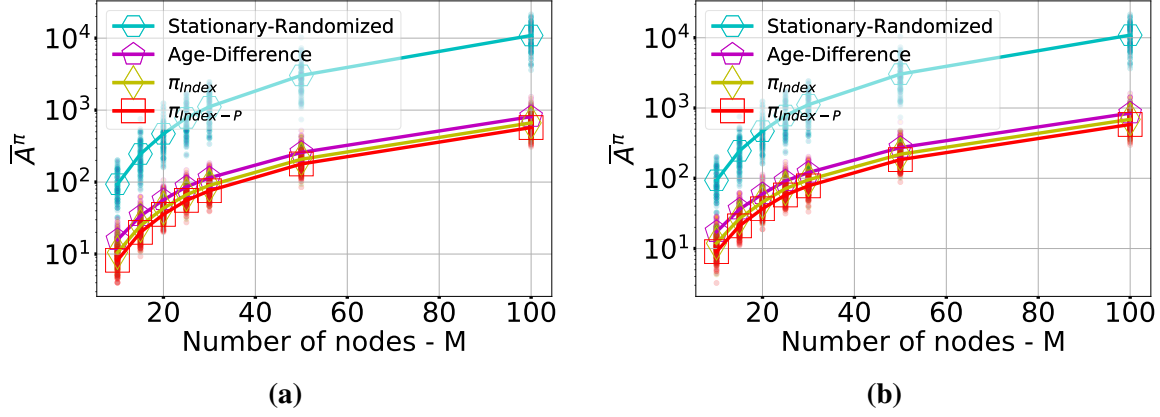


Figure 3.9: Simulated AAOII for tree networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under complete interference.

the Age-Difference and Stationary-Randomized policies⁵ from [9] to assess and justify the RMAB-based approach. In particular, lower AAOII values achieved by π_{Index} and $\pi_{\text{Index-P}}$ indicate the effectiveness of the RMAB framework for genie-aided policies. We note that the AAOII performance of the Age-Difference policy [9] is evaluated, even though the policy was originally designed for the AoI metric.

For line networks, we obtain the average $\bar{A}^\pi$ as a function of M , as illustrated in Fig. 3.4. For multiline and tree topologies, since 100 networks are generated for each M , we visualize $\bar{A}^\pi$ using scatter plots in Figs. 3.5 and 3.6. For each M , there are 100 points with coordinates $(M, \bar{A}^\pi)$, where each value of AAOII corresponds to a specific network realization. The scatter plots illustrate the variation of $\bar{A}^\pi$ across different networks for the same M under a given policy. We also plot the mean value of $\bar{A}^\pi$ across the 100 networks as a function of M for the multiline and tree topologies. The AAOII for line, multiline, and tree networks is illustrated in Figs. 3.4, 3.5, and 3.6, respectively, for both homogeneous and heterogeneous cases under K -hop interference with $K = 5$.

We also evaluate the performance of these policies under a complete interference model. The results for line, multiline, and tree networks are illustrated in Figs. 3.7, 3.8, and 3.9. For both interference models, the plots show that the Stationary-Randomized and Age-Difference policies yield higher AAOII compared to the index-based policies π_{Index} and $\pi_{\text{Index-P}}$. Although we do not have explicit lower bounds on the AAOII for genie-aided policies, these results demonstrate that our RMAB formulations produce policies with lower

⁵For the Stationary-Randomized scheduling policy, an activation set is chosen independently and uniformly at random from the set of feasible activation sets that satisfy the interference constraints in each slot.

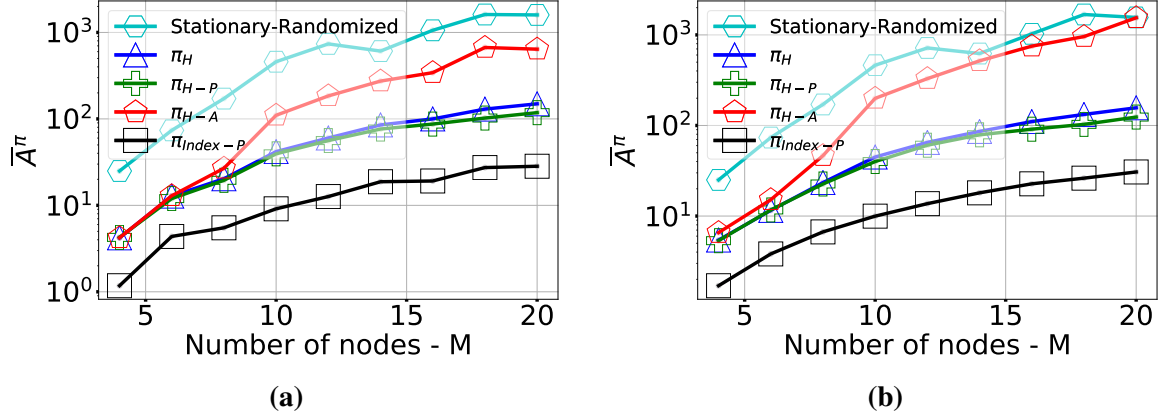


Figure 3.10: Simulated AAoII for line networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under K -hop interference ($K = 5$).

AAoII compared to heuristic approaches.

3.6.2 Performance of Implementable Policies

In this section, we compare the AAoII ($\bar{A}^\pi$) performance of the proposed centralized policies, π_H and π_{H-P} , with the Stationary-Randomized, π_{H-A} , and $\pi_{\text{Index-P}}$ policies. The comparisons are carried out across different network topologies and source parameters under both 5-hop and complete interference models. The results for 5-hop interference are shown in Figs. 3.10, 3.11, and 3.12. For both homogeneous and heterogeneous cases, the Stationary-Randomized policy and π_{H-A} yield higher AAoII compared to the proposed heuristic policies, π_H and π_{H-P} . Notably, for line networks with $M = 20$, the AAoII of π_{H-A} is 442.37% and 1146.88% of that of π_{H-P} for the homogeneous and heterogeneous cases, respectively (see Fig. 3.10). For multiline networks, the corresponding percentage deviations are 331.97% and 817.56% for homogeneous and heterogeneous networks (see Fig. 3.11). Similarly, for tree networks, the deviations of π_{H-A} from π_{H-P} are 322.28% and 493.14% for $M = 20$ (see Fig. 3.12).

We next evaluate the performance of the proposed policies under complete interference, shown in Figs. 3.13, 3.14, and 3.15. Under complete interference, all proposed heuristic policies π_H , π_{H-P} , and π_{H-A} —exhibit similar performance, with $\pi_{\text{Index-P}}$ serving as the lower bound. The Stationary-Randomized policy yields higher AAoII in all cases. In contrast, the AAoII of $\pi_{\text{Index-P}}$ provides a lower bound for the proposed policies, highlighting the advantage conferred by the additional information available to the genie-aided policies. The Stationary-Randomized policy yields a higher AAoII for all cases. In all cases, the

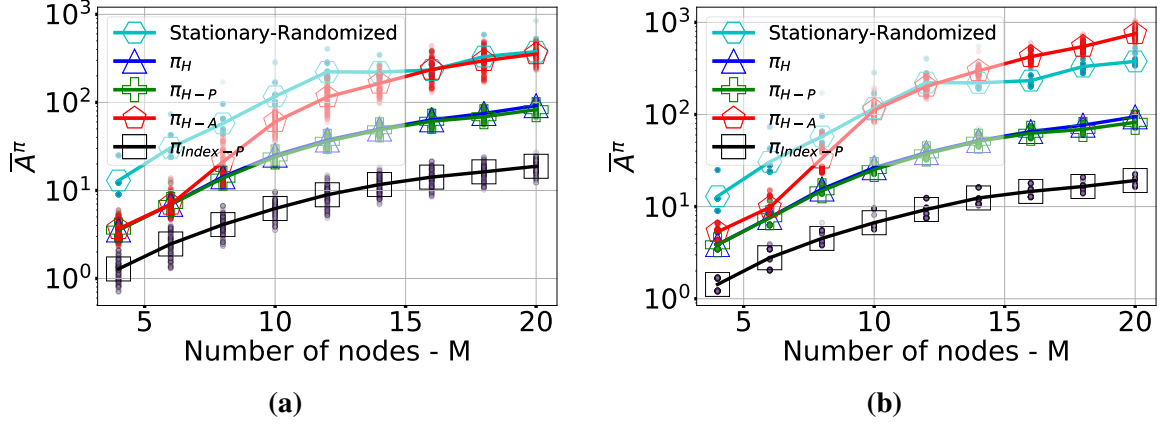


Figure 3.11: Simulated AAOII for multiline networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under K -hop interference ($K = 5$).

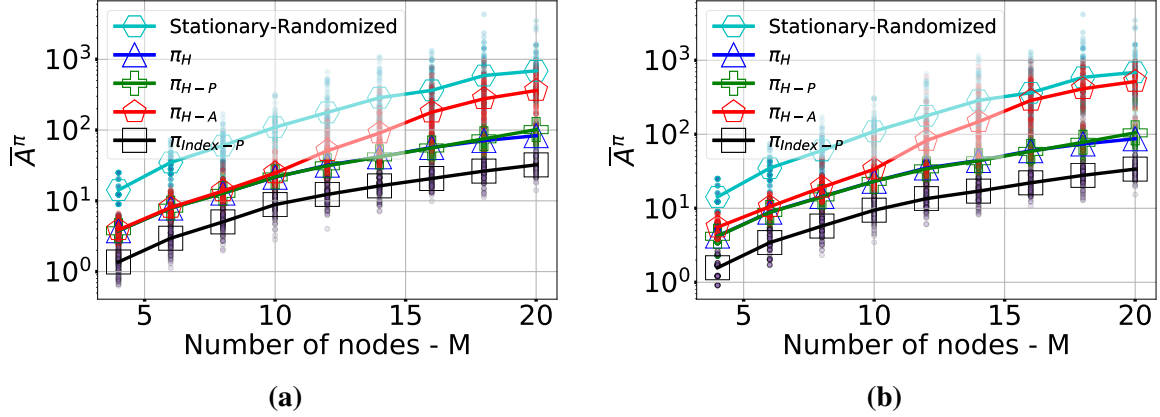


Figure 3.12: Simulated AAOII for tree networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under K -hop interference ($K = 5$).

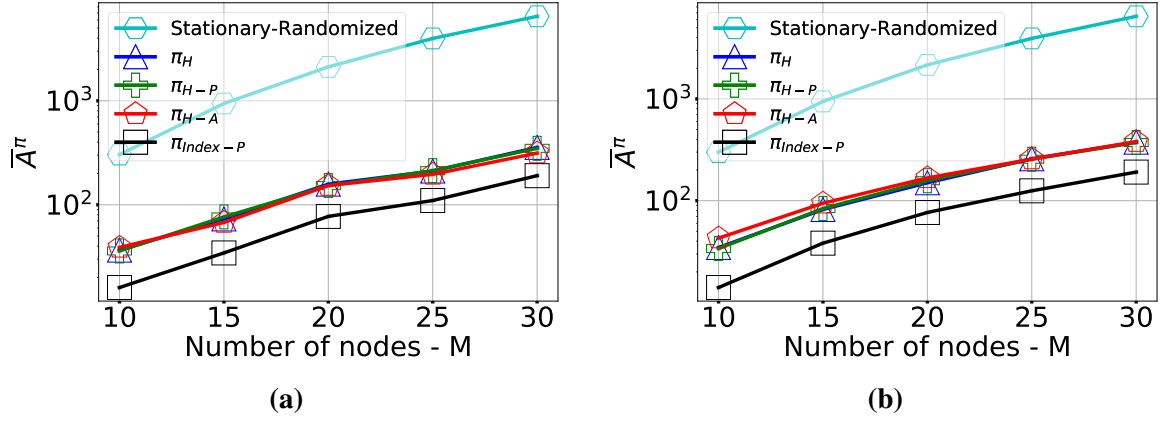


Figure 3.13: Simulated AAOII for line networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under complete interference.

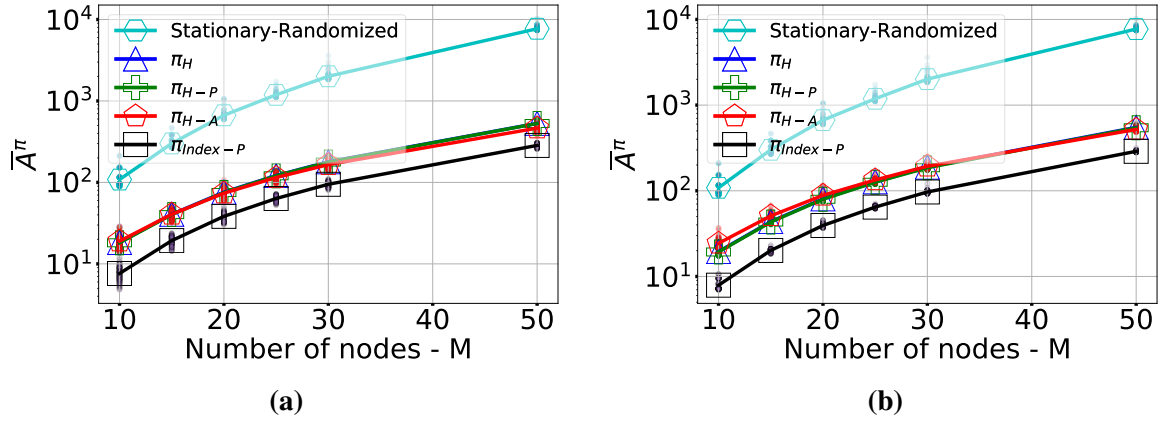


Figure 3.14: Simulated AAOII for multiline networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under complete interference.

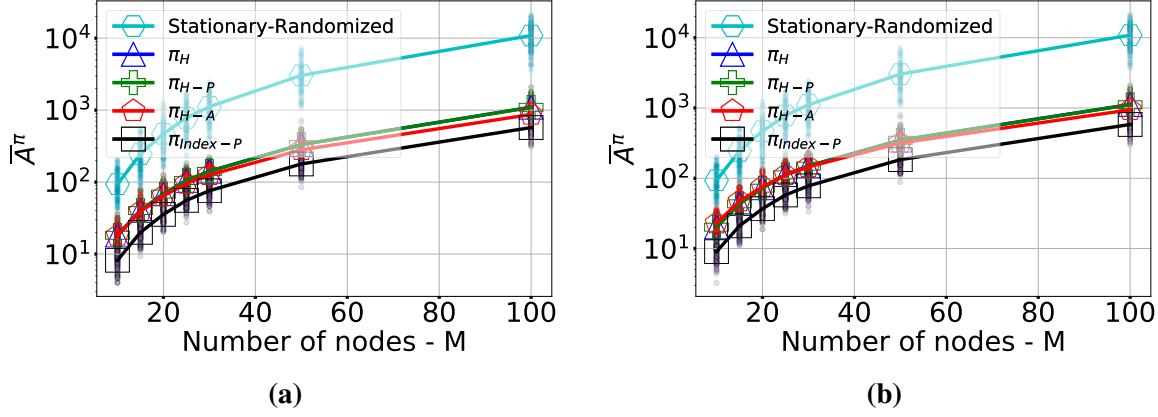


Figure 3.15: Simulated AAOII for tree networks under different scheduling policies: (a) homogeneous and (b) heterogeneous settings. Results are shown for varying M , with $\mathcal{S} = [M]$, under complete interference.

AAoII of $\pi_{\text{Index-P}}$ bounds the AoII for the proposed policies from below; the difference illustrates the advantage due to the information available to genie policies. We also compare the AoII performance of the policies in a multiline network scenario (Fig. 3.16(a)) with two sources, 1, 7, under a complete interference model. The average AoII, $\bar{A}^\pi$, was simulated for both homogeneous and heterogeneous cases across varying values of p_c^s .

In the homogeneous case, both sources are assumed to have the same change probability, p_c^s , selected from the set $\{0.04, 0.10, \dots, 0.52\}$. In the heterogeneous case, the change probability of source 1, p_c^1 , is varied over the set $\{0.01, 0.03, \dots, 0.09\}$, whereas source 7 is assigned a fixed change probability of $p_c^7 = 0.3$. Each p_c value is simulated 10 times, with $\bar{A}^\pi$ shown as scatter points and the mean indicated by a thick line (Figs. 3.16(b) and 3.16(c)). For both cases, the heuristic policy π_H outperforms the Age-Difference policy, illustrating the advantage of π_H , which is explicitly designed for AoII minimization.

We also evaluate the performance of the proposed policies under the 5-hop interference model, using the same multiline network shown in Fig. 3.16(a) and considering homogeneous and heterogeneous source cases for nodes 1 and 7, as in the previous scenario. In both cases (Figs. 3.17(a) and (b)), the heuristic policy π_H achieves lower AAOII than the Age-Difference policy, highlighting instances where an implementable centralized policy outperforms a policy with access to additional information. The gap from the baseline is expected because heuristic policies do not observe AoII assumed by the baseline. Obtaining improved lower bounds that account for the unavailability of AoII information remains an open problem and a potential direction for future research.

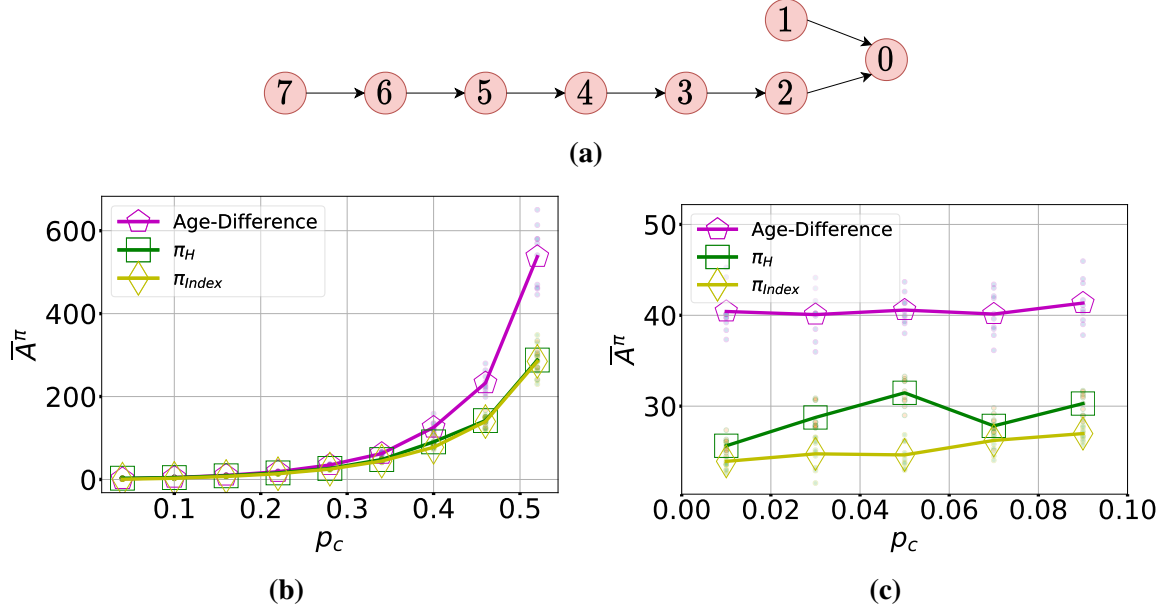


Figure 3.16: (a) Example multiline network topology. Simulated AAOII for a multiline network with two sources, $\{1, 7\}$, under complete interference: (b) homogeneous case with $p_c \in \{0.04, 0.1, \dots, 0.52\}$; (c) heterogeneous case where source 1 has $p_c^1 \in \{0.01, 0.03, \dots, 0.09\}$ and source 7 has $p_c^7 = 0.3$.

3.7 Other extensions

3.7.1 Discussion: Networks with link losses

In this section, we briefly discuss how policies can be obtained for networks with lossy links under complete interference. For a link activation $(s, (i, j))$, the packet is assumed to be successfully received with probability p_s . This discussion can be generalized to the case of link-dependent success probabilities; however, for ease of exposition, we assume that all links have the same p_s . If the transmission is unsuccessful, the packet is retained in node i 's buffer, and retransmission is attempted the next time $(s, (i, j))$ is activated. We propose a heuristic Whittle index-based policy derived from π_{Index} , using threshold policies in Π_{C-G} for single-source MDPs. To compute the index, the expressions for $\bar{A}^{\pi_{\tau_s}}$ and $\bar{D}^{\pi_{\tau_s}}$ for threshold policies must be modified. This is achieved by redefining $T(h^s)$ as the random time to reach state 0 starting from h^s in the Markov chain shown in Fig. 3.18. The Markov chain incorporates the effect of lossy links and determines the time required to successfully deliver a packet within the interval during which the state process remains unchanged. For K -hop and general interference models, genie-aided policies can be obtained using com-

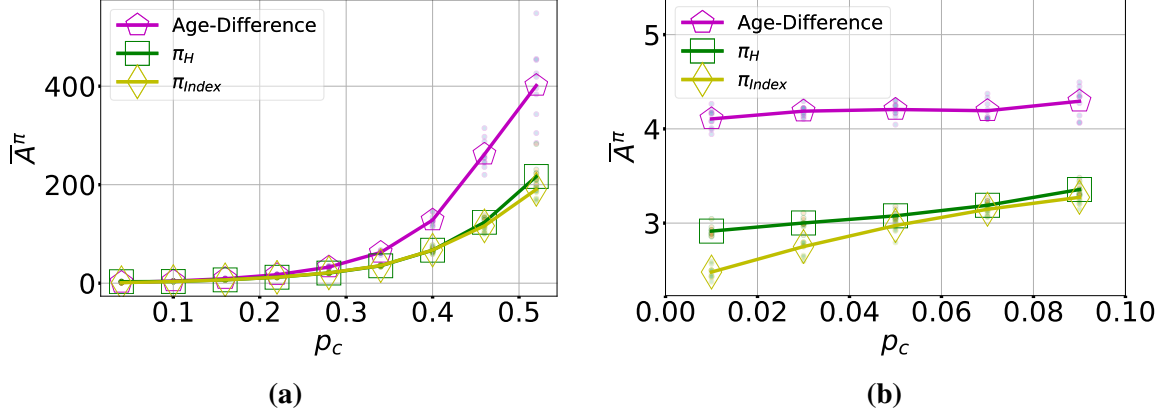


Figure 3.17: Simulated AAoII for a multiline network with two sources $\{1, 7\}$ under a 5-hop interference model: (a) homogeneous case with $p_c \in \{0.04, 0.1, \dots, 0.52\}$, and (b) heterogeneous case where source 1 has $p_c^1 \in \{0.01, 0.03, \dots, 0.09\}$ while source 7 has fixed $p_c^7 = 0.1$.

putations similar to (3.15) and (3.16), with the indices calculated using the modifications discussed above. However, this approach does not generalize to implementable policies, since even with knowledge of the scheduling history, $L_s(t)$ is not available at the central node.

3.7.2 Discussion: Networks with multiple channels

In this section, we consider an extension to the case where there are C orthogonal channels which are available. Consider the case with complete interference. We note that for the set of genie policies, i.e. Π_{C-G} , it is optimal to have only one packet in the system per source at a time (see Section 3.3). Furthermore, for such policies, we have defined the Whittle index based π_{Index} policy under the constraint that only one channel can be used. We note that the extension of π_{Index} to the case of C orthogonal channels is direct (we have a RMAB where C arms can be activated at a time). The policy π_{Index} now schedules C sources with the largest index values in a slot. We also note that prior results [111], [112] on RMABs can be utilized to show that such genie index policies are asymptotically optimal in the regime where both the number of sources $M \rightarrow \infty$ and $C \rightarrow \infty$ such that the ratio C/M is maintained constant. The implementable policies π_H can be extended to the multichannel case again by scheduling C sources. Currently, we do not have any analytical bound for the performance difference between π_H and π_{Index} . Therefore, the asymptotic optimality properties of π_H needs to be investigated.

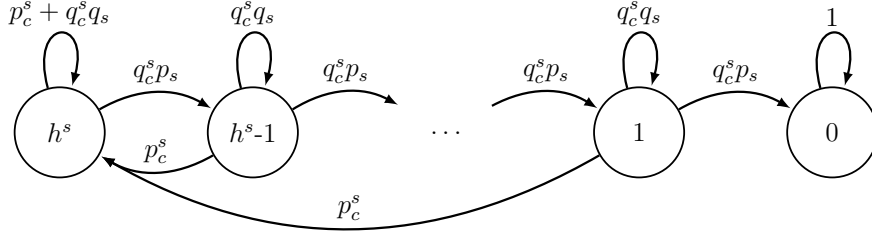


Figure 3.18: Transition diagram of a Markov chain modeling the evolution of the hop distance of a packet from source s under contiguous scheduling in a lossy network. The process starts at h^s , and state 0 denotes successful delivery. Here $q_c^s = 1 - p_c^s$ and $q_s = 1 - p_s$. With probability p_c^s , a packet is preempted and restarts at h^s ; otherwise, the hop distance decreases by one when there is no change (with probability q_c^s) and a successful transmission (with probability p_s).

3.8 Conclusions

In this chapter, we studied the design of centralized multi-hop sampling and scheduling policies for minimizing the AAoII, a goal-oriented metric for remote estimation. Multi-hop transmissions are subject to interference, which is modeled using complete interference, K -hop interference, or general interference graphs. Designing such policies is challenging because a direct formulation of the problem leads to an intractable POMDP, as the centralized scheduler does not have real-time access to information at all nodes in the network. To address this, we proposed two novel centralized sampling and scheduling policies that outperform the heuristic policies. Our approach began with genie-aided policies, which assume full information availability at the scheduler. While this led to MDP formulations, they remain intractable for networks with a large number of nodes due to high dimensionality. To overcome this, we formulated tractable RMAB problems by defining classes of genie policies with the restriction that at most a single packet from each source is maintained in the network at any time. We derived the Whittle index based scheduling policies π_{Index} and $\pi_{\text{Index-P}}$. Based on these index policies, we proposed two novel implementable centralized scheduling policies, π_{H} and $\pi_{\text{H-P}}$, for minimizing AAoII in multi-hop networks. The superior performance of these proposed policies compared to two heuristic policies is demonstrated through simulations. We note that the comparisons are made with respect to heuristic policies, as no prior work in the literature considers AAoII minimization in multi-hop networks. Comparing the AAoII performance of the centralized policies with the index policies highlights the impact of the unavailability of real-time information at each node on

the performance of the π_H and π_{H-P} schedulers. For the class of genie policies, we compare the performance of our index policies with the Age-Difference policy [9] (evaluated for AAoII) and show that our policies achieve lower AAoII. This demonstrates that the RMAB formulation is a reasonable and effective approach for designing AoII-minimizing policies in multi-hop networks with interference and lossless links.

Appendix

3.A Discussion: Justification for Threshold Policies

The state evolution of SS-MDP for a single source is also as in Table 3.4. In the proposition below, we show that an optimal policy for SS-MDP schedules a source's packet in contiguous slots from the source at hop distance h^s to the central node 0. This leads to a simplification while solving SS-MDP.

Proposition 3.5. *Any optimal policy for SS-MDP schedules a packet from source s to 0 in h^s contiguous slots.*

Proof. We use a sample path argument to show that an optimal policy schedules packets from a source s in h^s contiguous slots. Assume that a sample path for the source process $X^s(t)$ is given. Let π^* be an optimal policy for SS-MDP with a given λ (the dependence of π^* on λ is suppressed in the notation). For any sample path of $X^s(t)$ there are an infinite number of slots at which $X^s(t)$ changes (assuming that $p_c^s > 0$). Therefore, π^* schedules the source s infinitely often since otherwise, AAoII would increase without bound.

Consider any time t_i at which π^* schedules the source with the packet at hop distance h^s , i.e., it generates the packet and transmits the packet from the source to its parent. Since each packet requires h^s slots to reach 0 and since π^* is an optimal policy, there exists subsequent times $t_i + \delta_2, t_i + \delta_3, \dots, t_i + \delta_{h^s-1}$ at which the source is scheduled. The duration $\{t_i, \dots, t_i + \delta_{h^s-1}\}$ is denoted as $\text{SD}(t_i)$ (note that $\delta_{(\cdot)}$ depends on t_i). A total of δ_{h^s-1} slots are used by π^* to transfer the packet to 0 of which only h^s slots are actual transmissions.

Table 3.5: Evolution of the embedded AoII A_{m+1} from A_m when scheduling decision $D_m \in \{0, 1\}$ for SMDP. The single stage cost $c(A_m, D_m)$ of the SMDP is the sum of the single stage cumulative AoII cost (2nd column) and λD_m .

		A_{m+1}	Cumulative AoII cost
$D_m = 0$	if $A_m = 0$	0 w.p. q_c 1 w.p. p_c	A_m
	if $A_m > 0$	$A_m + 1$ w.p. 1	A_m
$D_m = 1$	if $A_m = 0$	0 w.p. $q_c^{h^s}$ x w.p. $q_c^{h^s-x} p_c$ for $x \in \{1, \dots, h^s\}$	$p_c \sum_{x=1}^{h^s} \left(q_c^{h^s-x} \frac{x(x-1)}{2} \right)$
	if $A_m > 0$	0 w.p. $q_c^{h^s}$ $A_m + h^s$ w.p. $1 - q_c^{h^s}$	$(1 - q_c^{h^s}) \left(A_m + \frac{h^s(h^s-1)}{2} \right)$

Now, consider the first time t_1 at which π^* schedules the source. Consider another policy π' which is such that it schedules the source at times $t_1, t_1 + 1, \dots, t_1 + h^s - 1$ and then does not schedule the source for $t_{h^s-1} - h^s$ slots. We note that both π^* and π' schedule the source's packet for h^s slots in $\text{SD}(t_1)$. We define π' to not schedule the source before t_1 as well as during the slots between $t_i + \delta_{h^s-1}$ and t_{i+1} . For the durations $\text{SD}(t_i)$, $\forall i > 1$ we define π' similarly as in $\text{SD}(t_1)$. Then, the time average $\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} D^s(t)$ is the same for π^* and π' .

We now consider the AoII evolution under the policies π^* and π' for the given sample path of $X^s(t)$. Let $A^*(t)$ and $A'(t)$ be the AoII evolution under π^* and π' respectively. At time 0, we assume that $A^*(0) = A'(0)$. Until the first slot t_1 the policy π' as defined above also does not schedule the source. Then $A^*(t_1) = A'(t_1)$ and let this value be a . Let δ_x be the number of slots after t_1 at which the process changes but within $\text{SD}(t_1)$, i.e., $\delta_x = \min \{1 \leq \delta \leq \delta_{h^s-1} : X(t_1 + \delta) \neq X(t_1)\}$. If the process does not change, then $\delta_x = \emptyset$. We discuss three cases below.

1. **$a > 0$ and $X^s(t)$ is such that $\delta_x = \emptyset$:** In this case, the state $X^s(t)$ has not changed for the duration $\delta_{h^s-1} + 1$, and $A^*(t_1 + \delta_{h^s-1} + 1) = 0$. Therefore, $A'(t_1 + h^s) = 0$ and $A'(t) < A^*(t)$ for $t \in \{t_1 + h^s, \dots, t_1 + \delta_{h^s-1}\}$. Also, $A'(t_1 + \delta_{h^s-1} + 1) = 0$.
2. **$a > 0$ and $X(t)$ is such that $\delta_x \neq \emptyset$:** In this case, the process has changed while the packet is being scheduled. Thus, delivering the packet does not make the remote estimate match the process. However, if $\delta_x > h^s$, then $A'(t_1 + h^s) = 0$ and for $t \in \{t_1 + h^s, \dots, t_1 + \delta_x - 1\}$, $A'(t) < A^*(t)$. For every $t \in \{t_1 + \delta_x, \dots, t_1 + \delta_{h^s-1}\}$ we have that $A'(t) \leq A^*(t)$ and $A'(t_1 + \delta_{h^s-1}) \leq A^*(t_1 + \delta_{h^s-1})$.
3. **$a = 0$:** In this case, if $\delta_x = \emptyset$, then $A^*(t) = A'(t) = 0$ for all $t \in \{t_1, \dots, t_1 + \delta_{h^s-1}\}$. If $\delta_x \neq \emptyset$, then $A^*(t) = A'(t)$, but their value can be positive. We note that $A'(t_1 + \delta_{h^s-1}) = A^*(t_1 + \delta_{h^s-1})$.

For all cases, we observe that during $\text{SD}(t_1)$, $A'(t) \leq A^*(t)$ and there are cases (which over all sample paths occur with positive probability) for which the inequality is strict. Also, at the end of $\text{SD}(t_1)$, $A'(t_1 + \delta_{h^s-1}) \leq A^*(t_1 + \delta_{h^s-1})$. Now, till the time t_2 , $A'(t) \leq A^*(t)$ as both π' and π^* do not schedule the source. At t_2 with $A'(t_2) \leq A^*(t_2)$ similar arguments can be used to show that $A'(t) \leq A^*(t)$ for all subsequent t in $\text{SD}(t_2)$ and for subsequent SD-s (with strict inequality for similar cases). Therefore, the sample path average $\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} A'(t) < \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} A^*(t)$. Thus, the policy π^* cannot be optimal for SS-MDP. Any optimal policy for SS-MDP should necessarily schedule the source in h^s contiguous slots. $\square$

With the above *structural result* for SS-MDP's optimal policies, we define decision epochs for SS-MDP as either the first slot or slots following slots in which either the source s was not scheduled or a packet was delivered to 0. Suppose $m \in \mathbb{Z}_+$ indexes these decision epochs. At each decision epoch, the scheduler must decide whether to schedule the source (denoted by $D_m = 1$) or not ($D_m = 0$). We denote the time to the next epoch as T_m . Then $T_m = 1$ and h^s for $D_m = 0$ and 1 respectively. The AoII embedded at these decision epochs is denoted as A_m . The evolution of A_m to A_{m+1} under $D_m \in \{0, 1\}$ is Markov (the evolution is shown in Table 3.5) under any stationary policy. Furthermore, if we define single stage costs $c(A_m, D_m)$ as defined in Table 3.5, then the average $\lim_{M \rightarrow \infty} \frac{\sum_{m=0}^{M-1} \mathbb{E}c(A_m, D_m)}{\sum_{m=0}^{M-1} \mathbb{E}T_m}$ is time average AoII. We note that a SMDP [107] can be defined with state A_m , decisions D_m , transition times T_m and single stage costs $c(A_m, D_m)$ in order to obtain an optimal policy for SS-MDP. The above SMDP can be solved using value iteration after data transformation [107]. We illustrate the optimal policy obtained from value iteration for a source with $h^s = 5$ in Fig. 3.1. The optimal policy is observed to be a *threshold policy on age*, i.e., the policy decides to schedule the source if $a_s \geq \tau_s$, where τ_s is a threshold which is a function of s (through h^s). Here note that a_s is a particular value of A_m . A threshold policy with threshold τ_s is denoted as π_{τ_s} . An equivalent policy π_{τ_s} for SS-MDP (which takes decisions in every slot) can be defined, which schedules the source s if $a_s \geq \tau_s$, where a_s is a particular value of $A^s(t)$. The sample paths of π_{τ_s} for SS-MDP are the same as that for SMDP, since π_{τ_s} schedules the source in h^s contiguous slots as $A^s(t)$ is non-decreasing until the packet reaches 0.

3.B Proof of Proposition 3.1

We separately consider two cases for the policy π_{τ_s} : a) $\tau_s = 0$ and b) $\tau_s > 0$.

a) $\tau_s = 0$: Since $A^s(t) \geq 0$, the source is always scheduled. Therefore $\overline{D}^{\pi_{\tau_s}} = 1$. We also note that the evolution of $H^s(t)$ is periodic: $H_s(0) = h^s$, $H_s(1) = h^s - 1, \dots$, $H_s(h^s - 1) = 1$ and then again $H_s(h^s) = h^s$, $H_s(h^s + 1) = h^s - 1, \dots$. We consider the evolution of $A^s(t)$ embedded at $t \in \{0, h^s, 2h^s, \dots, nh^s, \dots\}$. Let A_n denote the embedded age $A_s(nh^s)$. We note that A_n is a Markov chain with transitions as described below: Suppose $A_n = 0$, then $A_{n+1} = x, \forall x \in \{1, \dots, h^s\}$ w. p. $q_c^{h^s-x} p_c$ and $A_{n+1} = 0$ w.p. $q_c^{h^s}$. If $A_n = a > 0$, then $A_{n+1} = 0$ w.p. $q_c^{h^s}$ or $A_{n+1} = a + h^s$ w.p. $1 - q_c^{h^s}$.

Now, to derive the stationary probability π_0 , consider the probability generating func-

tion (PGF) of A_{n+1} ,

$$\begin{aligned}\mathbb{E}[Z^{A_{n+1}}] &= \sum_a \pi_a [\mathbb{E}[Z^{A_{n+1}} | A_n = a]] = \pi_0 \sum_{x=0}^{h^s} Z^x q_c^{h^s-x} p_c + \pi_0 q_c^{h^s+1} \\ &\quad + \sum_{a>0} \pi_a (q_c^{h^s} + (1 - q_c^{h^s}) Z^{a+h^s})\end{aligned}\tag{3.17}$$

where π_a is the stationary joint probability of $A_n = a$.

Setting $Z = 0$, we use the fact that $\mathbb{E}[Z^{A_{n+1}}] \big|_{Z=0} = \mathbb{P}(A_{n+1} = 0) = \pi_0$. Thus the stationary probability π_0 under the threshold policy:

$$\begin{aligned}\pi_0 &= \pi_0 q_c^{h^s} + \sum_{a>0} \pi_a q_c^{h^s} = q_c^{h^s} (\pi_0 + \sum_{a>0} \pi_a) \\ &= q_c^{h^s}.\end{aligned}\tag{3.18}$$

Now, the embedded AoII is obtained as $\mathbb{E}[A_{n+1}] = \mathbb{E}[\mathbb{E}[A_{n+1} | A_n]]$. Conditioning on A_n ,

$$\begin{aligned}\mathbb{E}[A_{n+1}] &= \pi_0 \mathbb{E}[A_{n+1} | A_n = 0] + \sum_{a>0} \pi_a \mathbb{E}[A_{n+1} | A_n = a]. \\ &= \pi_0 \left(\sum_{x=0}^{h^s} x q_c^{h^s-x} p_c \right) + \sum_{a>0} \pi_a (a + h^s) (1 - q_c^{h^s}) \\ &= (h^s + 1) - q_c^{h^s} - \frac{1}{p_c} (1 - q_c^{h^s}) + \frac{(1 - \pi_0) h^s (1 - q_c^{h^s})}{q_c^{h^s}}\end{aligned}\tag{3.19}$$

Substituting $\pi_0 = q_c^{h^s}$ (from (3.18)), we obtain,

$$\mathbb{E}[A_{n+1}] = 1 + h^s q_c^{h^s} - h^s + h^s q_c^{-h^s} - q_c^{h^s} + \frac{q_c^{h^s}}{p_c} - \frac{1}{p_c}\tag{3.20}$$

The cumulative AoII, denoted by $C(a)$, represents the total area under the AoII evolution curve over one renewal interval, given that the system starts in state $A_n = a$. Under the threshold policy, and using the Markov Renewal Reward Theorem (MRRT) [92], the average AoII is given by $\overline{A}^{\pi_{\tau_s}} = \frac{\sum_a \pi_a C(a)}{h^s}$, which is expressed as follows.

$$\begin{aligned}\overline{A}^{\pi_{\tau_s}} &= \frac{\pi_0 \sum_{x=0}^{h^s} \frac{x(x-1)}{2} q_c^{h^s-x} p_c + \sum_{a>0} \pi_a (a h^s + \frac{h^s(h^s-1)}{2})}{h^s} \\ &= \mathbb{E}[A] + \frac{h^s - 1}{2} + \frac{\pi_0}{h^s} \left(\frac{p_c q_c^{h^s} f(h^s, q_c)}{2 q_c^2} \right) - \frac{h^s(h^s - 1)}{2}\end{aligned}\tag{3.21}$$

Finally, using $\mathbb{E}[A]$ from Eq. (3.20), we obtain the final expression as follows.

$$\overline{A}^{\pi_{\tau_s}} = \frac{0.5p_c q_c^{2h^s} f(h^s, q_c)}{h^s q_c^2} + 0.5h^s q_c^{h^s} - 0.5h^s + h^s q_c^{-h^s} - 0.5q_c^{h^s} + 0.5 + \frac{q_c^{h^s}}{p_c} - \frac{1}{p_c} \quad (3.22)$$

where $f(h^s, q_c)$ is a function related to the hop distance and q_c , which is given by $f(h^s, q_c) = \frac{r^{h^s} (h^s + (h^s)^2 (r-1)^2 + 2r - h^s r^2) - 2r}{(r-1)^3 r}$, with $r = \frac{1}{q_c}$.

b) $\tau_s > 0$: We consider the case where the threshold $\tau_s > 0$. We identify a renewal process in the evolution of the MC $(A^s(t), S^s(t), H^s(t))$ and then use renewal reward theorem (RRT) [92] to compute $\overline{A}^{\pi_{\tau_s}}$ and $\overline{D}^{\pi_{\tau_s}}$. The renewal epochs of the process are defined to be times at which MC has the state $(0, 0, h^s)$. It can be shown that starting from an arbitrary state at $t = 0$, the Markov chain reaches $(0, 0, h^s)$ in finite expected time⁶. We now describe each renewal cycle. Consider the MC to be at state $(0, 0, h^s)$ at a time t_0 . Since $\tau_s > 0$, the source is not scheduled. We note that $A^s(t), t \geq t_0$ remains at zero till the first time at which the source process $X^s(t)$ changes. The time taken for the source process to change is a Geometric(p_c) random variable, which we denote as G_1 . The AoI $A^s(t)$ starts incrementing from the slot $t = t_0 + G_1$. We note that a further $\tau_s - 1$ slots are required so that $A^s(\cdot) \geq \tau_s$ and the source s is scheduled. Once $A^s(\cdot) \geq \tau_s$ it is scheduled contiguously for h^s slots, since $A^s(\cdot)$ increments until the packet reaches 0. At the end of h^s slots, $A^s(\cdot)$ drops to 0 w.p. $(q_c^{h^s})^{h^s}$ (which is the probability that the source value has not changed for h^s slots). If it does not drop to 0, then the source is again scheduled contiguously for another h^s slots (since $A^s(\cdot) \geq \tau_s$). This process continues until $A^s(\cdot)$ drops to 0. The number of slots taken for $A^s(\cdot)$ to drop to zero is $h^s G_2$ where G_2 is a Geometric($(q_c^{h^s})^{h^s}$) random variable. We note that whenever the packet reaches 0, $S^s(\cdot)$ and $H^s(\cdot)$ are reset to 0 and h^s respectively. So, the renewal cycle is the random time $G_1 + \tau_s - 1 + h^s G_2$. We note that the renewal cycles are IID with finite mean.

We note that in every renewal cycle, the cumulative duration of time for which the source is scheduled is given by $h^s \mathbb{E}G_2$. Using RRT, we have that

$$\begin{aligned} \overline{D}^{\pi_{\tau_s}} &= \frac{\mathbb{E}[G_2]h^s}{\mathbb{E}[G_1] + (\tau_s - 1) + \mathbb{E}[G_2]h^s} \\ &= \frac{h^s p_c}{q_c^{h^s} + h^s p_c + (\tau_s - 1)p_c q_c^{h^s}} \end{aligned} \quad (3.23)$$

⁶The first renewal cycle has a different distribution, however this does not affect the averages obtained using RRT and is not discussed further here.

Table 3.6: Evolution of $A^s(t)$, $S^s(t)$, and $H^s(t)$ from time t to $t+1$ for policies in Π_{C-G-1P} under Case I: $X^s(t) \neq X^s(t-1)$ and Case II: $X^s(t) = X^s(t-1)$.

		$D_s(t) = 0$	$D_s(t) = 1$
Case I	$S^s(t+1)$	0	0
	$A^s(t+1)$	$A^s(t) + 1$	$A^s(t) + 1$
	$H^s(t+1)$	h^s	h^s
Case II	$S^s(t+1)$	$S^s(t) + 1$, if $H^s(t) < h^s$, $S^s(t)$, if $H^s(t) = h^s$.	$S^s(t) + 1$, if $H^s(t) > 1$, 0, if $H^s(t) = 1$.
	$A^s(t+1)$	0, if $A^s(t) = 0$, $A^s(t) + 1$, if $A^s(t) > 0$.	if $H^s(t) > 1$: 0, if $A^s(t) = 0$, $A^s(t) + 1$, if $A^s(t) > 0$. if $H^s(t) = 1$: 0, if $A^s(t) = 0$, 0, w.p. $q_c^{S^s(t)+1}$ if $A^s(t) > 0$, $A^s(t) + 1$, w.p. $1 - q_c^{S^s(t)+1}$ if $A^s(t) > 0$.
	$H^s(t+1)$	$H^s(t)$	$H^s(t) - 1$, if $H^s(t) > 1$, h^s , if $H^s(t) = 1$.

We further note that, for each renewal cycle, the cumulative AoII is given by,

$$\mathbb{E} \left[\frac{\tau_s(\tau_s - 1)}{2} + \tau_s G_2 h^s + \frac{(G_2 h^s)(G_2 h^s - 1)}{2} \right]. \quad (3.24)$$

. Again using RRT and simplifying, we obtain,

$$\overline{A}^{\pi_{\tau_s}} = \frac{p_c q_c^{-2h^s} (2(h^s)^2 - h^s q_c^{h^s} (1 + h^s - 2\tau_s) + q_c^{2h^s} \tau_s (\tau_s - 1))}{2(p_c(h^s q_c^{h^s} + (\tau_s - 1)) + 1)}$$

3.C RMAB Formulation for Π_{C-G-1P}

We proceed similarly as in Section 3.3 for $\pi \in \Pi_{C-G-1P}$. For the system, we define a state vector $((A^s(t), S^s(t), H^s(t)), \forall s \in \mathcal{S})$. Again, $D^s(t)$ indicates whether source s is scheduled in slot t ($D^s(t) = 1$) or not ($D^s(t) = 0$) under a policy π . The evolution of the state vector is shown for a source in Table 3.6 (every source has a corresponding evolution). The evolution is discussed for two cases: I) $X^s(t) \neq X^s(t-1)$ and II) $X^s(t) = X^s(t-1)$. In case I, the packet is preempted, resetting $S^s(t)$ to 0 and $H^s(t)$ to h^s , while $A^s(t)$ increments to $A^s(t) + 1$. This holds for both $D^s(t) = 1$ and $D^s(t) = 0$. In case II, the evolution of $A^s(t)$, $S^s(t)$, and $H^s(t)$ depends on different conditions; we describe

the evolution of $S^s(t)$ in the following, while that of $A^s(t)$ and $H^s(t)$ can be obtained from Table 3.6. If $D_s(t) = 0$, $S^s(t)$ increments to $S^s(t) + 1$ when the packet is in transit ($H^s(t) < h^s$). However, if the packet remains with the source ($H^s(t) = h^s$), $S^s(t)$ stays unchanged. If $D_s(t) = 1$, $S^s(t)$ increments to $S^s(t) + 1$ when the packet is in transit ($H^s(t) < h^s$) and $H^s(t)$ decrements by 1. However, if $H^s(t) = 1$, $S^s(t)$ resets to 0 and H^s to h^s .

For policies $\in \Pi_{C-G-1P}$, the evolution of $((A^s(t), S^s(t), H^s(t)), \forall s \in \mathcal{S})$ is a controlled Markov chain. The state space is the Cartesian product $\times_{s \in \mathcal{S}} (\mathbb{Z}_+ \times \mathbb{Z}_+ \times [h^s])$. The action space is $\mathcal{S}$ since the action in every slot is to choose the source s to be scheduled. For a particular state vector $((a_s, s_s, h_s), \forall s)$ we define a single stage cost as $\sum_s a_s$. We note that the AAoII minimization problem (5.2) can be formulated as an average cost MDP with state transitions for every s as in Table 3.6 and with the constraint that $\sum_s D^s(t) \leq 1, \forall t$. The rest of the development is similar to that in Section 3.3 leading to the definition of a single-source MDP problem denoted as SS-MDP. For a source s , the objective of SS-MDP is the same as that of (3.9).

We solved the SS-MDP using the value iteration procedure with a truncated state space, where $A^s(t) \in \{0, \dots, 1000\}$, $S^s(t) \in \{0, \dots, 1000\}$, $h^s = 3$, $p_c = 0.3$, and $p_s = 1$. An optimal policy for SS-MDP is shown in Fig. 3.19(a). The policy is a function of a_s , s_s , and h_s ; we show the policy as a function of a_s for different combinations of s_s and h_s . For $\lambda = 1000$, the optimal policy follows a threshold structure on age, scheduling the source if $a_s \geq \tau_s$, where τ_s is a threshold which in general is dependent on both s_s and h_s . However, we observe (numerically) the following: (1) the threshold is dependent on h_s but not on s_s , (2) the threshold is monotone non-decreasing with h_s . Consider the evolution of the system under an optimal threshold policy (as shown in Fig. 3.19(a)). Initially, $h_s = h^s$; therefore, the source is scheduled once the age increases to the τ_s , which is a function of h^s . Once the source is scheduled $h_s = h^s - 1$ for which the threshold is lower. Also, the AoII is an increasing function of time once it is positive and if the packet has not been delivered. Thus, the source is scheduled in contiguous slots, and the evolution of the system is similar to that of a threshold policy with a threshold which is independent of h_s . This is the justification for the claim that an optimal policy for this SS-MDP is a threshold policy.

Additionally, we also compare the optimal values of $\bar{A}^\pi$ and $\bar{D}^\pi$ obtained from the value iteration procedure with those obtained for the threshold policy from Proposition 3.4 in Fig. 3.19(b). The values match, which strengthens the claim. We use a similar approach to derive the Whittle Index from Proposition 3.3, where we use the observation that the set of achievable points $(\bar{D}^{\pi_\tau}, \bar{A}^{\pi_\tau})$ is convex, as shown in Fig. 3.19(b).

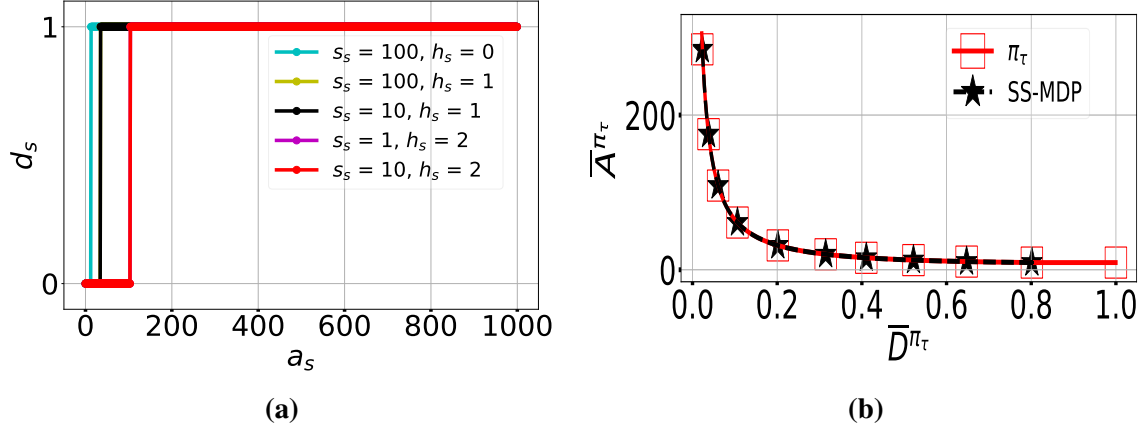


Figure 3.19: (a) Illustration of the threshold nature of the optimal policy for the MDP for a single source with $h^s = 3$ for $\lambda = 1000$, as a function of s_s and h_s . (b) Pairs $(\bar{D}^{\pi_{\tau_s}}, \bar{A}^{\pi_{\tau_s}})$ achieved by the threshold policy as τ_s varies in $\{0, 1, 2, \dots, 600\}$. The optimal $(\bar{D}^{\pi}, \bar{A}^{\pi})$ obtained from (3.9) by varying λ is also shown.

3.D Proof of Proposition 3.4

We separately consider two cases for the threshold τ_s : a) $\tau_s = 0$ and b) $\tau_s > 0$.

a) $\tau_s = 0$: When the threshold is 0, the source is scheduled in every slot. Therefore $\bar{D}^{\pi_{\tau_s}} = 1$. For obtaining $\bar{A}^{\pi_{\tau_s}}$ we identify a renewal process in the evolution $A^s(t)$. Suppose we start with $A^s(t) = 0$. Since $X^s(t)$ changes with probability p_c^s in every slot t independent of the past, the time till the next state change follows a geometric distribution G_1 with mean $\frac{1}{p_c^s}$. At this slot, any current packet is pre-empted and a new packet is generated at s and transmitted. Once the state changes $A^s(t)$ increases. It continues increasing until the slot a packet is delivered such that the state contained in the packet is the source's state at the delivery slot. This delivery time is $T(h^s)$, the random time taken by the Markov chain in Fig. 3.3 with initial state h^s to hit 0. At this slot $A^s(\cdot)$ is 0. We define such slots as renewal epochs. Then, renewal durations are IID and are distributed as $G_1 + T(h^s)$. The expected cumulative AoI in a renewal cycle is $\mathbb{E}[T(h^s)] \mathbb{E}[T(h^s) - 1] / 2$. Applying RRT [92], we obtain that $\bar{A}^{\pi_{\tau_s}}$ is:

$$\bar{A}^{\pi_{\tau_s}} = \frac{1}{2} \frac{\mathbb{E}[T(h^s)] + \mathbb{E}[T(h^s)^2]}{\frac{1}{p_c} + \mathbb{E}[T(h^s)]} \quad (3.25)$$

b) $\tau_s > 0$: For this case also, we identify a renewal process in the evolution of MC $(A(t), S(t), H(t))$. Then, we use RRT [92] to obtain $\bar{A}^{\pi_{\tau_s}}$ and $\bar{D}^{\pi_{\tau_s}}$. The renewal epochs of the process are defined to be slots at which MC has the state $(0, 0, h^s)$. It can be shown that starting from an arbitrary state at $t = 0$, the MC reaches $(0, 0, h^s)$ in finite expected

time. We now describe each renewal cycle. Consider the MC (as shown in Fig. 3.3) to be at state $(0, 0, h^s)$ at a time t_0 . Since $\tau_s > 0$, the source is not scheduled. We note that $A^s(t), t \geq t_0$ remains at zero till the first time at which the source process $X^s(t)$ changes. This time duration is a Geometric(p_c^s) random variable, denoted as G_1 . The AoII $A^s(t)$ starts incrementing from the slot $t = t_0 + G_1$ (in that slot t , $A^s(t) = 1$). We note that a further $\tau_s - 1$ slots are required so that $A^s(\cdot) \geq \tau_s$ and the source s is scheduled. Once $A^s(\cdot) \geq \tau_s$ source s is scheduled, for this MC as shown in Fig. 3.3, suppose $T(h^s)$ denotes the random variable, that the time for the packet to reach 0, starting from h^s . The $A_s(\cdot)$ will increment until the packet reaches 0. Then the state of the system resets to $(0, 0, h^s)$. So the renewal cycle has a duration of $\frac{1}{p_c^s} + \tau_s - 1 + \mathbb{E}[T(h^s)]$. Using RRT, we have $\overline{D}^{\pi_{\tau_s}} = \mathbb{E}[T(h^s)] / ((1/p_c^s) + \tau_s - 1 + \mathbb{E}[T(h^s)])$, since the source is scheduling during $T(h^s)$ slots. We also note that in every renewal cycle the cumulative expected AoII is $\frac{\tau_s(\tau_s-1)}{2} + \tau_s \mathbb{E}[T(h^s)] + \mathbb{E}\left[\frac{T(h^s)(T(h^s)-1)}{2}\right]$. Applying RRT, we obtain $\overline{A}^{\pi_{\tau_s}}$ as in (3.13). To evaluate the derived expressions for $\overline{A}^{\pi_{\tau_s}}$ and $\overline{D}^{\pi_{\tau_s}}$, we numerically compute $\mathbb{E}[T(h^s)]$ and $\mathbb{E}[T(h^s)^2]$. We note that $\mathbb{E}[T(i)]$, satisfies the recursive relation:

$$\mathbb{E}[T(i)] = 1 + \sum_j p_{ij} \mathbb{E}[T(j)], \quad (3.26)$$

where $i, j \in [h_s] \cup \{0\}$, $T(0) = 0$, and p_{ij} are the transition probabilities of the Markov chain in Fig. 3.3. This set of linear equations can be solved to obtain $\mathbb{E}[T(h^s)]$. To obtain $\mathbb{E}[T(h^s)^2]$ we note that $T(i) = 1 + W_i$, where W_i is a random variable that is T_j with probability p_{ij} . We apply z -transforms on both sides of this relationship to obtain:

$$\mathbb{E}[z^{T(i)}] = \mathbb{E}[z \times z^{W_i}] = z \sum_j p_{ij} \mathbb{E}[z^{T(j)}]. \quad (3.27)$$

The second moment of $T(i)$ can be obtained using $z \frac{d}{dz} \left[z \frac{d}{dz} (\mathbb{E}[z^{T(i)}]) \right]$ and then setting $z = 1$. This leads us to the following set of equations:

$$\mathbb{E}[T(i)^2] = 1 + 2 \sum_j p_{ij} \mathbb{E}[T(j)] + \sum_j p_{ij} \mathbb{E}[T(j)^2], \quad (3.28)$$

which can be solved for $\mathbb{E}[T(i)^2]$ once $\mathbb{E}[T(i)], \forall i$ has been obtained.

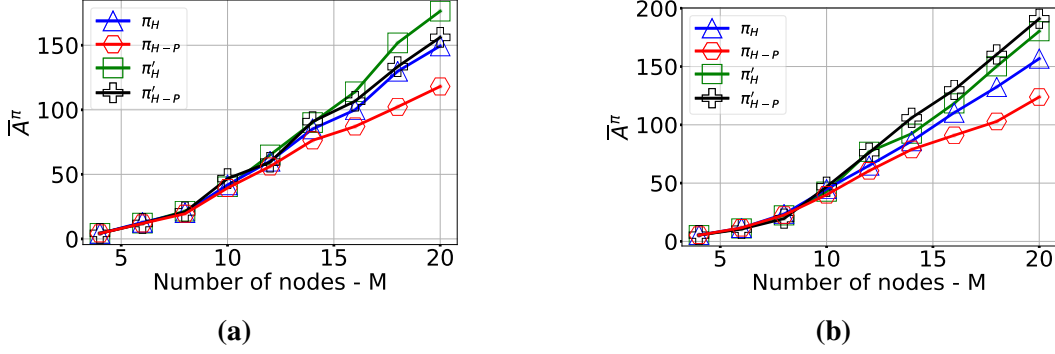


Figure 3.20: Simulated AAoII for line networks under the scheduling policies π_H and π_{H-P} , along with their alternative implementations π'_H and π'_{H-P} , for (a) homogeneous and (b) heterogeneous networks. Simulations are performed for various values of M under 5-hop interference.

3.E Alternative Implementation of π_H

We propose a variant π'_H of the policy π_H that does not require the source to transmit the AoII at a packet generation time. The policy π'_H again maintains an *estimate* of the average $\mathbb{E}A^s(t), \forall s$ with knowledge of the source model parameter p_c^s or $q_c^s = 1 - p_c^s$. We define $\mathcal{B}^s(t) = \mathbb{P}\{\hat{X}^s(t) = X^s(t)\}$, i.e., the probability that the remote estimate is the same as the value of the process for source s (this is similar to the approach in [102]). We then have the following update for $\mathcal{B}^s(t)$.

$$\mathcal{B}^s(t+1) = \begin{cases} (q_c^s)^{(t+1)-t_g^s} & \text{if packet is received,} \\ \mathcal{B}^s(t) q_c^s & \text{if no packet is received.} \end{cases}$$

where t_g^s is the generation time of the received update packet from source s . The scheduler also maintains an AoII estimate $\hat{A}^s(t)$ which is defined as follows:

$$\hat{A}^s(t+1) = (1 - \mathcal{B}^s(t+1))(\hat{A}^s(t) + 1),$$

where we assume $\hat{A}^s(0) = 0$. The policy π'_H computes the Whittle index using (3.12) (by first rounding $\hat{A}^s(t)$ to an integer value), and schedules the source with the maximum Whittle index. A similar approach is used to define the variant π'_{H-P} . A comparison of the AAoII for π'_H and π'_{H-P} with that of π_H and π_{H-P} is shown in Fig. 3.20 for homogeneous and heterogeneous source cases (parameters are chosen as in Section 3.6). The results indicate that π_H and π_{H-P} outperform their corresponding alternative implementations π'_H and π'_{H-P} , respectively.

Chapter 4

Periodic Deterministic Policies for Minimizing Average Age of Information in Time-Slotted Multi-Hop Networks

4.1 Introduction and Prior Work

We consider a convergecast scenario in which multiple sensors monitor physical processes of interest, sample these processes, and generate data packets containing the samples, which are then transmitted to a sink through a multihop wireless network. The remote estimation or reconstruction of the monitored processes at the central node (or sink) is a key problem in modern Internet-of-Things (IoT) applications [1]. Information freshness metrics, such as the Age of Information (AoI) [34], have been used as network-layer measures in such applications [1]. Motivated by recent trends in IoT networks, we consider networks operating in a time-slotted manner with periodic deterministic scheduling, for example, time-slotted protocols such as IEEE 802.15.4e TSCH [4]. In this chapter, we focus on the design of novel periodic deterministic sampling and scheduling policies that minimize the average Age of Information (AAoI) for convergecast in time-slotted multihop tree networks under K -hop interference constraints. We also derive novel numerical and analytical bounds on AAoI performance in such networks.

*This chapter is based on the manuscript by Nibin Raj, Vineeth B.S., and Pratik Sharma, “Periodic Deterministic Sampling and Scheduling Algorithms for Minimum Age of Information in Time-Slotted Multihop Networks,” in IEEE Internet of Things Journal, vol. 12, no. 9, pp. 12958-12974, May 2025, doi: 10.1109/JIOT.2024.3523024.

4.1.1 Prior Work

Remote estimation of processes monitored by sensors at a sink is critical for inference and control in IoT and cyber-physical systems. In such applications, metrics that quantify data freshness, such as AoI, are becoming increasingly important, in contrast to the traditional performance measures like latency and throughput. A broad survey of AoI-related research is available in [33], whereas [34] provides an overview of foundational results for single-source, single-destination systems. While many studies have addressed AoI in multi-source, single-hop networks (e.g., [38], [39], [40], [41]), research on AoI in multi-hop network scenarios remains limited. Some of the studies related to multi-hop networks are reviewed as follows. The problem of minimizing the AAoI in multi-hop networks under stationary randomized scheduling policies was investigated in [42], considering multiple source-destination pairs and general interference constraints, with analytical results derived under the assumption of interference-free multi-hop transmissions. Other studies have investigated AoI in various multi-hop settings, including interference-free single-flow networks [43], ring topologies [44], energy-harvesting two-hop networks [45], throughput-constrained networks [46], and multi-source time-slotted networks [47]. Tripathi et al. [9] considered age minimization in multi-hop networks and proposed three policies: stationary randomized, Age-Difference, and Age-Debt. In contrast, our work focuses on tree topologies under K -hop interference constraints and develops periodic deterministic sampling and scheduling policies to minimize the AAoI. Periodic deterministic policies were studied in [44, 47] for multi-source, multi-sink networks under a complete interference model. In contrast, our work considers a convergecast scenario with a general K -hop interference model. Multi-hop networks also appear in applications such as multilevel caching and edge computing, where the freshness of cached data is critical. Bastopcu [48] characterized the average freshness in a two-hop caching model as a function of content and cache update rates, while age-optimal cache update policies for two-hop networks were explored in [49]. The AoI metric has also been used to evaluate wireless link performance between vehicles and cloud servers [113], and strategies to minimize AoI for data offloading in mobile edge computing have been investigated in [114]. Many of these works focus on two-hop network models, consisting of a content server, cache, and end-user, which contrasts with the more general multi-hop networks with K -hop interference considered in this chapter.

Time-slotted deterministic scheduling is particularly relevant for industrial wireless networks, which impose strict requirements on reliability, latency, jitter, and power [27], [28]. Such deterministic scheduling is supported by industrial wireless standards, including

WirelessHART [5], ISA100.11a [29], and IEEE 802.15.4e TSCH [4]. Notably, protocols like TSCH implement periodic scheduling policies with a period defined by a *slot-frame* [4], which is a contiguous set of N time slots over which the schedule repeats. Motivated by these considerations, we focus on designing periodic deterministic policies to minimize AAoI in wireless multi-hop networks under interference constraints.

Scheduling policies can be either centralized or decentralized, each with its own advantages and trade-offs. Centralized policies can optimize performance metrics with a holistic view of the network and its dynamics, but incur additional communication overhead. Decentralized policies, on the other hand, can adapt more flexibly to changing network conditions, though potentially at the cost of optimality. In this chapter, we focus on centralized scheduling policies for AAoI minimization. Such centralized deterministic policies are supported in standards through entities like the Path Computation Element in 802.15.4e TSCH or the Network Manager in WirelessHART [90]. While deterministic scheduling algorithms for slotted networks have been studied for other performance metrics in works such as [115], [116], [117], our focus here is on AAoI, a goal-oriented metric for designing joint sampling and scheduling policies.

4.1.2 Contributions

We consider a multi-hop tree network with M source nodes, K -hop interference constraints, and reliable link transmissions. Our primary contributions are twofold: (a) the numerical and analytical characterization of AAoI performance for periodic deterministic scheduling policies, and (b) the design of low-complexity heuristic policies that achieve lower AAoI compared to existing multi-hop scheduling approaches, such as the non-periodic Age-Difference policy [9]. The specific contributions of this work are outlined as follows:

1. The AAoI is analytically characterized for periodic deterministic policies satisfying a periodic delivery condition, expressed in terms of the period and delivery delay for each source.
2. For periodic deterministic policies, AAoI minimization is formulated as an integer programming problem for a given period. A linear relaxation of this problem is used to obtain novel numerical lower bounds on the minimum AAoI for that period. By solving the problem across different periods, a lower bound over all possible periods is obtained.
3. Fundamental analytical lower bounds on the minimum AAoI for periodic determinis-

tic policies are derived, which are not available in prior work for the general network model considered. These analytical bounds are shown to closely match the numerical lower bounds obtained above.

4. Two periodic deterministic scheduling policies are proposed for tree networks under K -hop interference constraints. The algorithmic complexity of generating these policies is characterized as a function of K and M , and is significantly lower than that of solving the integer programming formulation.
5. Simulation-based performance analysis demonstrates that the AAoI of the proposed policies is close to the corresponding numerical lower bound for the same period. Moreover, the proposed policies outperform existing multi-hop AAoI-minimizing policies, such as Age-Difference and Age-Debt, which are generally non-periodic.
6. Simulations of the proposed policies over a single period can also be used to obtain upper bounds on the AAoI for periodic deterministic policies.
7. The design framework is extended to general non-decreasing functions of AoI. Analytical and numerical lower bounds on the time-average of such functions are derived, and the performance of the proposed policies is evaluated in this generalized setting.

Notations: The notations used in this chapter are given in Table 4.1.

Table 4.1: List of symbols and their descriptions used in Chapter 4

Symbol	Description
$[M]$	Set of integers $\{1, 2, \dots, M\}$.
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Directed tree representing the multi-hop network, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of directed edges.
$\mathcal{S}$	Set of source nodes (sensor nodes) that generate packets for the root node.
$ U $	Cardinality of the node set U .
$(i, j) \in \mathcal{E}$	Directed edge from node i to node j , representing a communication link from i to j .
$(s, (i, j))$	Link activation on link (i, j) for source s .
$(s, (s, j))$	Sampling of source s and activation of link (s, j) .
$\mathcal{I}_G$	Interference graph.

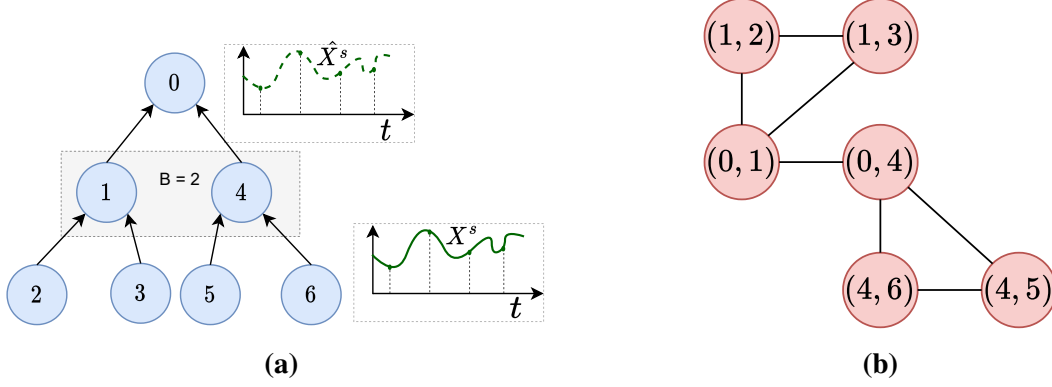


Figure 4.1: (a) Example multihop tree network with 6 nodes and 2 top-subtrees. For node 6, the monitored process X^s and its remote estimate $\hat{X}^s$ at node 0 are shown. Packets carrying samples of X^s are transmitted to node 0. (b) Corresponding interference graph, where nodes represent links and edges indicate mutual interference under 1-hop interference.

K	Parameter defining the K -hop interference model, with $K \in \{1, 2, \dots\}$.
$U(t)$	Activation set at time slot t .
u_k	Element of the activation set $U(t)$ representing a link activation of the form $(s, (i, j))$.
$\mathbb{Z}_+$	Set of positive integers.
$A_0^s(t)$	AoI of source s at the sink node 0 at time slot t .
A_{avg}	Average AoI over all sources.
d_s	Delivery delay for source s .
N	Period of the periodic scheduling policy.
h_s	Hop distance of source s from the sink node 0.
$g_{s,n}$	Indicator variable for packet generation by source s in slot $n \in [N]$.
$a_{l,n}^s$	Indicator variable for link transmission: $a_{l,n}^s = 1$ if a packet generated by source s is transmitted on link l in slot n , otherwise $a_{l,n}^s = 0$.
$\mathcal{R}_{\text{top}}$	Set of all top-subtree root nodes.
B	Number of top-subtree roots, i.e., $B = \mathcal{R}_{\text{top}} $.
$\mathcal{T}_b$	Subtree rooted at node $b \in \mathcal{R}_{\text{top}}$.
n_b	Number of nodes in subtree $\mathcal{T}_b$, i.e., $n_b = \mathcal{T}_b $, with $\sum_{b=1}^B n_b = M$.
N_{lb}	Analytical lower bound on the period N .

4.2 System Model and Problem Statement

We represent the multi-hop network as a directed tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of sensor nodes in the network and $\mathcal{E}$ represents the set of communication links (see Fig. 4.1a). Tree-based topologies are widely used in practical network deployments, for instance in routing protocols such as RPL [89]. In the convergecast setting, the sink corresponds to the root node of the tree, denoted by 0. We assume that the network contains M additional nodes, so that $|\mathcal{V}| = M + 1$. Each node other than the sink is treated as a *source node* that generates data packets¹. In addition to generating its own packets, each node is responsible for forwarding packets originating from its *descendant* nodes². We define *top-subtrees*, where a top-subtree is a subtree of $\mathcal{G}$ whose root is a child of node 0. We also consider specific network structures, including line and multiline networks. A *line network* is a linear topology consisting of nodes indexed from 0 to M , with a directed edge from node i to node $i - 1$ for all $i \in [M]$. A *multiline network* is a tree in which each top-subtree has a line network structure. An edge $(i, j) \in \mathcal{E}$, where $i, j \in \mathcal{V}$, represents a communication link over which node i can transmit data to node j . Each link (i, j) can carry at most one packet at any given time. A packet may either originate at node i (i.e., $s = i$) or be forwarded from another node and stored at node i . All packets are assumed to be of equal size, and the transmission of a single packet requires one time slot. The system evolves in discrete time with slots indexed by $t \in \mathbb{Z}_+$. We assume that all link transmissions take place over a shared communication channel and that each node is equipped with a single half-duplex radio interface. Two link transmissions are said to be concurrent if they occur within the same time slot. We adopt a K -hop interference model for the tree network, where $K \in \{1, 2, \dots\}$. Under this model, concurrent transmissions on two links that are separated by at least K hops do not interfere with each other. In contrast, if two links are separated by fewer than K hops, simultaneous transmissions over these links lead to interference, and the corresponding packets cannot be successfully received. The K -hop interference model is commonly used in the literature to capture interference effects in wireless networks, e.g., [118], [119].

The interference relationships among links can be represented using an interference graph (for example, under 1-hop interference, the corresponding interference graph for the

¹In this chapter, packet generation is not exogenous but is governed by the sampling and scheduling policy, as described later.

²For a node i , a node d is said to be a descendant if it is either a child of i or a descendant of any child of i , defined recursively. For a link $l = (i, j)$, another link $l' = (k, l)$ is considered a descendant link if node k is a descendant of node i .

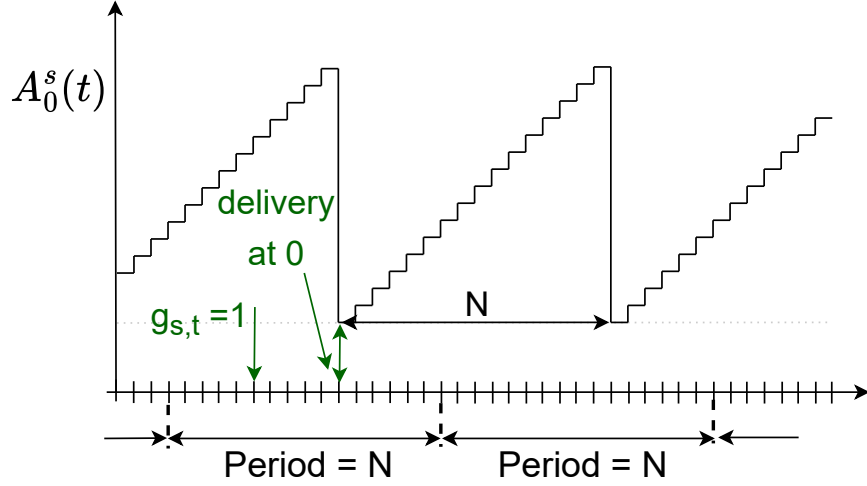


Figure 4.2: Illustration of the periodic deterministic evolution of $A_0^s(t)$ with period N . A packet is generated for source s and delivered with delay $d_s = 5$ slots. The evolution is shown in discrete time at slot boundaries.

tree network in Fig. 4.1a is shown in Fig. 4.1b). We also consider a *complete interference* model, in which every pair of links interferes with each other, so that at most one transmission can occur in a given time slot without interference. For the K -hop interference model, link transmissions that satisfy the interference constraints are assumed to be successful, i.e., packet losses are not considered³.

In this chapter, we study multi-hop sampling and scheduling policies that, in each time slot, determine both the set of sources that generate packets and the set of links that are activated for transmission. Additionally, for each active link that transmits, a decision must be made regarding which source's packet is to be transmitted. To formalize this decision at each time slot t , we define an *activation set* $\mathcal{U}_t$, which represents the set of links that can be scheduled for concurrent transmission under the K -hop interference constraints. The activation set is given by $\mathcal{U}_t = \{u_1, u_2, \dots, u_{|\mathcal{U}_t|}\}$, where each element corresponds to a link (i, j) transmitting a packet generated by source s [9]. Thus, an element $u_k \in \mathcal{U}_t$ is denoted by $(s, (i, j))$ and represents a specific link activation. The set of all feasible activation sets is denoted by $\mathbb{U}$. This set is constructed by first forming an interference graph, where each node represents a link, and an edge between two nodes indicates that the corresponding links cannot be activated simultaneously (see Fig. 4.1b). Feasible activation sets are then obtained by identifying cliques in the complement of this interference graph. For each link (i, j) , the elements u_k are defined by associating the link with all sources s

³An extension to scenarios with packet losses is discussed in Section 4.6.1.

such that s is either node i or a descendant of i .

As an illustrative example, consider the tree network shown in Fig. 4.1a. A feasible activation set under 2-hop interference is given by $\mathcal{U}_t = \{(2, (2, 1)), (5, (5, 4))\}$. An element $u_k \in \mathcal{U}_t$, for instance $(2, (2, 1))$, represents the transmission of source 2's packet over the link from node 2 to node 1. A link activation of the form $(s, (i, j))$ with $s \neq i$ indicates that a packet generated by source s is forwarded from node i to node j in slot t . If $s = i$, it implies that node s generates a new packet and transmits it to node j in the same slot. For example, if $(s, (i, j)) = (4, (4, 0))$, source 4 generates a packet and transmits it directly to the sink. On the other hand, if $(6, (4, 0))$ is activated, node 4 forwards a packet from source 6 to the sink. Packets generated at the source nodes are buffered at intermediate nodes and forwarded hop-by-hop towards the sink node 0. The buffer size at each node for every source is assumed to be infinite⁴.

A *sampling and scheduling policy*, denoted by $\mathcal{U}$, is defined as the sequence of activation sets $(\mathcal{U}_t, t \in \mathbb{Z}_+)$. In this chapter, we focus on centralized scheduling policies that construct transmission schedules based on complete knowledge of the network topology. We further restrict attention to periodic deterministic policies. A periodic deterministic policy $\mathcal{U}_N$ with period N generates a sequence of activation sets $\{\mathcal{U}(0), \mathcal{U}(1), \dots, \mathcal{U}(N-1)\}$, such that $\mathcal{U}_t = \mathcal{U}(t \bmod N)$, $\forall t \in \mathbb{Z}_+$.

Age of Information: To evaluate the performance of a policy, we use the Average Age of Information (AAoI), which quantifies the freshness of information at the sink node 0 for each source. For a source node s , the AoI at the sink at time slot t is denoted by $A_0^s(t)$, with the initial condition $A_0^s(0) = 0$. The AoI evolves as

$$A_0^s(t) = t - g_0^s(t)$$

where $g_0^s(t)$ represents the generation time of the most recently received packet from source s at node 0, and we assume $g_0^s(0) = 0$.

Since we consider periodic deterministic policies, the evolution of $A_0^s(t)$ for all $s \in [M]$ is also periodic with the same period as the policy. A typical AoI $A_0^s(t)$ evolution is illustrated in Fig. 4.2. Whenever the sink receives a new packet, the AoI drops to the time elapsed since the packet's generation; otherwise, it increases linearly over time.

⁴For the periodic deterministic policies considered in this chapter, the required buffer size is at most M . The infinite buffer assumption is primarily used to facilitate the analysis of non-periodic policies.

The AAoI for a source is defined as follows:

$$\text{AAoI}(s) = \lim_{T \rightarrow \infty} \left[\frac{1}{T} \sum_{t=0}^{T-1} A_0^s(t) \right]. \quad (4.1)$$

We also define A_{avg} as the average AAoI across all sources $s \in [M]$, given by,

$$A_{\text{avg}} = \frac{1}{M} \sum_{s=1}^M \text{AAoI}(s)$$

If a source does not generate any packet within a period, or if the generated packet does not reach the sink, then $\text{AAoI}(s)$, and consequently A_{avg} , becomes unbounded⁵. To avoid such cases, we restrict our attention to policies that satisfy a *periodic delivery* condition. A periodic deterministic policy is said to satisfy $\text{PeDe}(s)$ if, under the periodic schedule, at least one packet from source s is successfully delivered to node 0 in every period. A policy is said to satisfy PeDe if it satisfies $\text{PeDe}(s)$ for all $s \in [M]$. More generally, periodic policies that generate and deliver multiple packets from each source (i.e., $m_s \geq 1$ packets from source s per period) can also be considered. In this chapter, we focus on the simplified case with $m_s = 1$,⁶ while the extension to $m_s > 1$ is left for future work.

Objective: Our objective is to design periodic deterministic scheduling policies that minimize A_{avg} , where the minimization is taken over the set of all periodic deterministic policies (with period N , any positive integer) that satisfy the PeDe condition. The minimum value of A_{avg} , if it exists, is denoted by A_{avg}^* .

4.3 Minimization of Average AoI and Lower Bounds for Periodic Deterministic Policies

In this section, we first derive a characterization of $\text{AAoI}(s)$ for policies⁷ that satisfy the PeDe(s) condition. This characterization is expressed in terms of the delivery delay d_s for each source s . The delivery delay is defined as the number of time slots between the

⁵For periodic policies, if a source s does not generate a packet within a period, it never generates one, implying $g_0^s(t) = 0$ for all t . Similarly, if a packet is generated but does not reach the sink within a period, then $g_0^s(t) = 0$ for all t . Hence, as $t \rightarrow \infty$, the AoI, given by $t - g_0^s(t)$, grows without bound, leading to an infinite average AoI.

⁶Allowing m_s for all s to be decision variables in order to minimize A_{avg} leads to a significantly more complex problem.

⁷For brevity, we use the term “policies” to refer to periodic deterministic sampling and scheduling policies.

generation of a packet at source s and its successful delivery to node 0.

Proposition 4.1. *For a policy satisfying $PeDe(s)$ with period N and delivery delay d_s , the AAoI of source s is given by*

$$AAoI(s) = d_s + \frac{N - 1}{2}$$

Proof. For a policy, $A_0^s(t)$ is a periodic deterministic function of t (an example is shown in Figure 4.2). Suppose the period is N . The area under $A_0^s(t)$ for a period is $d_s N + N(N - 1)/2$ (the sum of the areas of a “rectangle” and a “triangle”) as shown in Figure 4.2. Therefore, the average AAoI(s) is the ratio of this area and the period N . Therefore, we have that $AAoI(s) = d_s + (N - 1)/2$. $\square$

The delivery delay satisfies $d_s \geq h_s$, where h_s denotes the hop distance of source s from node 0. For the special case of a tree network under complete interference (i.e., at most one link transmission can occur in a time slot), we derive the following characterization of A_{avg}^* .

Proposition 4.2. *For a tree network with complete interference, we have*

$$A_{avg}^* = \left(\frac{1}{M} \sum_{s=1}^M h_s \right) + \frac{(\sum_{s=1}^M h_s) - 1}{2}$$

Proof. For a network with complete interference, at most one link can be scheduled in any given time slot. Therefore, h_s transmissions are required to deliver a packet from source s to node 0. Consequently, the minimum period required is $N = \sum_{s=1}^M h_s$. Moreover, a delivery delay of $d_s = h_s$ can be achieved by forwarding each packet in contiguous time slots once it is generated. The result then follows directly from Proposition 4.1. $\square$

4.3.1 AAoI Minimization as an Integer Programming Problem

In the following, we formulate the problem of determining a policy that minimizes A_{avg} for a fixed period N as an optimization problem. The slots within a period are indexed by $n \in [N]$. Each node $i \in [M]$ acts as a source and generates one packet in every period. For a source $s \in [M]$, let $g_{s,n} \in \{0, 1\}$ indicate whether a packet is generated in slot $n \in [N]$, where $g_{s,n} = 1$ if a packet is generated, and $g_{s,n} = 0$ otherwise⁸. For a link $l \in \mathcal{E}$, let

⁸Here, we consider a more general setting in which the generation time $g_{s,n}$ may differ from the time at which the packet is first transmitted. However, our numerical results indicate that, for minimizing AAoI, it is optimal to transmit a packet in the same slot in which it is generated.

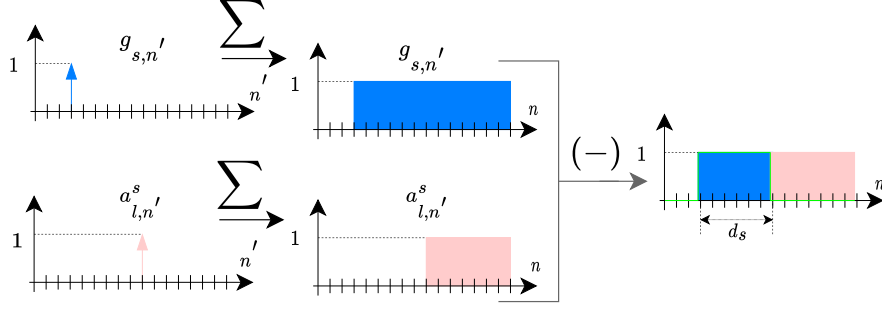


Figure 4.3: Illustration of the delay d_s for a source node s as a function of the decision variables $g_{s,\cdot}$ and $a_{l,\cdot}^s$. For $l \in E^-(0)$, d_s denotes the number of slots between packet generation ($g_{s,\cdot} = 1$) and its reception at node 0 ($a_{l,\cdot}^s = 1$). This delay is obtained by taking the cumulative sums of g and a and computing their difference.

$a_{l,n}^s \in \{0, 1\}$ denote whether a packet from source s is transmitted over link l in slot n . Specifically, $a_{l,n}^s = 1$ indicates that a packet from source s is transmitted on link l in slot n , and $a_{l,n}^s = 0$ otherwise. Thus, for each $n \in [N]$, $s \in [M]$, and $l \in \mathcal{E}$, the variables $a_{l,n}^s$ defines the activation set in slot n . We now describe the constraints that are placed on the sampling and scheduling policy.

Each node generates exactly one packet in every period. Therefore,

$$\sum_{n=1}^N g_{s,n} = 1, \forall s \in [M]. \quad (4.2)$$

Since packet losses are not considered, all packets generated by each source must be delivered to the sink by the end of the period through one of its incoming links. Therefore,

$$\sum_{l \in E^-(0)} \sum_{n=1}^N a_{l,n}^s = 1, \forall s \in [M]. \quad (4.3)$$

For a node $i \in \mathcal{V}$, let $E^+(i)$ and $E^-(i)$ denote the sets of outgoing and incoming links, respectively. Since each node is equipped with a single half-duplex radio interface, for every slot $n \in [N]$ and for all nodes $v \in [M] \cup 0$, we have:

$$\sum_{s=1}^M \left(\sum_{l \in E^+(v)} a_{l,n}^s + \sum_{l \in E^-(v)} a_{l,n}^s \right) \leq 1. \quad (4.4)$$

Consider a link $l \in \mathcal{E}$, and let N_l denote the set of its neighboring links in the interference

graph (for example, in Fig. 4.1b, for the link $(0, 4)$, we have $N_{(0,4)} = \{(0, 1), (4, 6), (4, 5)\}$). Then, the K -hop interference constraint is given by⁹: for all $l \in \mathcal{E}$, $l' \in N_l$, and $n \in [N]$,

$$\sum_{s=1}^M a_{l,n}^s + \sum_{s=1}^M a_{l',n}^s \leq 1. \quad (4.5)$$

We also note that, in any given slot, a node cannot transmit more packets than it has either generated or received from other nodes in all previous slots. This constraint, for every $i, s \in [M]$ and $n \in [N]$, is expressed as

$$\begin{aligned} \sum_{l \in E^+(i)} a_{l,n}^s &\leq \sum_{l \in E^-(i)} \sum_{n' \leq n-1} a_{l,n'}^s + \\ \delta(i-s) \sum_{n' \leq n} g_{s,n'} &- \sum_{l \in E^+(i)} \sum_{n' \leq n-1} a_{l,n'}^s, \end{aligned} \quad (4.6)$$

where $\delta(x) = 1$ if $x = 0$, and 0 otherwise. The PeDe(s) condition is enforced through constraints (4.2), (4.3), and (4.6). For certain values of N , these constraints may not be satisfiable.

In the following, we perform a search over the period N to identify feasible values of N and the corresponding A_{avg}^* . Suppose that, for a given N , all the constraints (including the PeDe condition) are satisfied. Then, from Proposition 4.1, we have $\text{AAoI}(s) = d_s + \frac{N-1}{2}$. We further note that the delivery delay d_s can be expressed in terms of the variables $g_{s,n}$ and $a_{l,n}^s$ as follows (see Fig. 4.3 for an illustration):

$$d_s = \sum_{n=1}^N \sum_{n' \leq n} g_{s,n'} - \sum_{l \in E^-(0)} \sum_{n=1}^N \sum_{n' < n} a_{l,n'}^s, \forall s \in [M]. \quad (4.7)$$

Therefore, for a fixed period N , the objective function A_{avg} can be expressed in terms of the decision variables $g_{s,n}$ and $a_{l,n}^s$. For this fixed N , a policy that minimizes A_{avg} can be obtained by solving the following optimization problem:

$$\begin{aligned} \min_{(g_{s,n}, a_{l,n}^s, \forall s \in [M], n \in [N], l \in \mathcal{E})} \quad & A_{\text{avg}}(N), \\ \text{subject to} \quad & (4.2), (4.3), (4.4), (4.5), (4.6), \end{aligned} \quad (4.8)$$

⁹Note that if $l' \in N_l$, then $l \in N_{l'}$. Therefore, if l is a descendant of l' , we do not consider l' from N_l when implementing the K -hop interference constraints in an optimization problem solver to prevent repetition of constraints.

where we explicitly indicate the dependence of A_{avg} on the period N . For a fixed N , the above problem can, in principle, be solved using integer (binary) linear programming (ILP) methods. Let the optimal value, if it exists, be denoted by $A_{\text{avg}}^*(N)$. The minimum average AoI across all sources is then defined as $A_{\text{avg}}^* = \min_N A_{\text{avg}}^*(N)$. We characterize A_{avg}^* in the following.

Proposition 4.3. *Suppose $N_{\min}$ is any lower bound on the minimum N required for (4.8) to be feasible and $N_{\max} = \sum_s h_s$. Then, we have that $A_{\text{avg}}^* = \min_{N \in \{N_{\min}, \dots, N_{\max}\}} A_{\text{avg}}^*(N)$.*

Proof. We note that considering a period $N < N_{\min}$ is not required since (4.8) is not feasible for such an N . The AAOI for every source s is $d_s + (N - 1)/2$ from Proposition 4.1. For a particular feasible N , we obtain an optimal value $d_s^*(N)$ for an optimal policy. As N increases $d_s^*(N)$ could decrease; for example if $N = \sum_s h_s$, we can achieve $d_s^*(N) = h_s$ (by scheduling packets from a single source in consecutive slots). We note that the only advantage of using larger values of N is that the delivery delay d_s for a source could be reduced. Thus, considering a $N > N_{\max} = \sum_s h_s$ is also not required, since $d_s^*(N_{\max}) = h_s$. $\square$

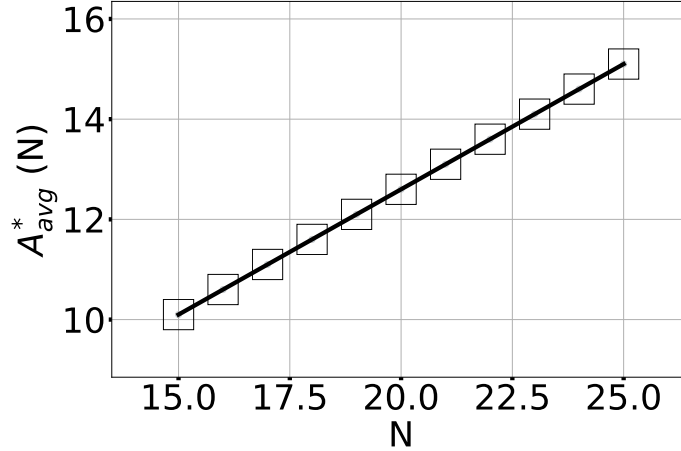


Figure 4.4: Numerically obtained optimal value $A_{\text{avg}}^*(N)$ as a function of N for a multiline network with $K = 2$. The lower bound on the feasible period is $N_{\text{lb}} = 15$.

In the following, we obtain an analytical lower bound N_{lb} on the period for any policy satisfying PeDe (see Proposition 4.6). Thus, we can choose $N_{\min} = N_{\text{lb}}$. Therefore, A_{avg}^* can be solved in principle by solving $N_{\max} - N_{\min} + 1$ optimization problems for fixed N and then finding the minimum. In numerical experiments, we have observed that $A_{\text{avg}}^*(N)$ is monotonically non-decreasing in N (see Fig. 4.4), so that finding the smallest N for

which the optimization problem is feasible is sufficient. However, we note that ILP is NP-hard, so that this method does not scale as M increases. A detailed discussion on this is given in Appendix 4.A.

Numerical lower bounds $A_{glpk-lb}(N)$ and $A_{glpk-lb}$: For a fixed value of N , we obtain a linear program by relaxing the integer constraints in problem (4.8), which is solved using the GLPK solver [120]. Let the optimal value of this relaxed problem be denoted by $A_{glpk-lb}(N)$. We have that $A_{glpk-lb}(N) \leq A_{avg}^*(N)$. We then define

$$A_{glpk-lb} = \min_{N \in \{N_{min}, \dots, N_{max}\}} A_{glpk-lb}(N) \quad (4.9)$$

which serves as a lower bound on A_{avg}^* .

4.3.2 Analytical Lower Bounds

In this section, we present analytical lower bounds on A_{avg}^* . From Proposition 4.1, we note that $AAoI(s)$ is given by the sum of the delivery delay d_s and the term $(N - 1)/2$. Our analytical lower bounds are derived by independently lower bounding d_s and N . We begin by establishing lower bounds on the period N . The lower bounds on N are obtained by determining the minimum number of time slots required to collect packets from all sources at node 0. To do this analysis, we introduce the following notation. Let $\mathcal{R}_{top}$ denote the set of all top-subtree roots, and define $B = |\mathcal{R}_{top}|$. The subtree rooted at $b \in \mathcal{R}_{top}$ is denoted by $\mathcal{T}_b$, and let $n_b = |\mathcal{T}_b|$, so that $\sum_{b=1}^B n_b = M$. An example with $B = 2$ and $n_1 = 3$ is shown in Fig. 4.1a. We define a *t-line* as a linear subgraph of the tree consisting of the nodes along the path from a leaf node in $\mathcal{T}_b$ to its corresponding root $b \in \mathcal{R}_{top}$, and then to node 0 (e.g., $2 \rightarrow b = 1 \rightarrow 0$ in Fig. 4.1a).

Line Networks: We first discuss the lower bound for a line network scenario, before extending the analysis to more complex network topologies such as multiline and tree networks.

Proposition 4.4. *Consider a line network with $M = n_1$ source nodes, i.e., a sequence of nodes $0, 1, 2, \dots, M = n_1$. Any periodic deterministic policy satisfying PeDe has a period N bounded below by N_{lb} , where $N_{lb} = \sum_{j=1}^{K+1} (n_1 - j + 1)$.*

Proof. To obtain a lower bound on the period we consider a relaxed version of the optimization problem (4.8). In this relaxed version of the problem we assume that constraints (4.2) and (4.3) holds. We then try to obtain a required period such that the K hop interference constraint (4.5) and the half-duplex constraint (4.4) holds only for the link $(1, 0)$. We

do not consider (4.6) at all. We note that the required period is then related to the minimum n required for variables $a_{l,n}^s \in \{0, 1\}$ such that the above relaxed constraints are met. Since there are n_1 sources, a total of n_1 packets must be delivered to the sink node 0. These packets are received at node 0 via the link $(1, 0)$, which requires n_1 time slots (or there should be $n_1 \times n_1$ variables $a_{(1,0),n}^s \in \{0, 1\}$). Packets generated by sources $2, 3, \dots, n_1$ must first be forwarded to node 1 before being transmitted to node 0. In particular, node 2 must forward $n_1 - 1$ packets to node 1. Due to the K -hop interference constraint, the transmissions over the link $(2, 1)$ cannot be scheduled concurrently with those over $(1, 0)$. Hence, an additional $n_1 - 1$ slots are required to schedule transmissions over $(2, 1)$, beyond the n_1 slots used for $(1, 0)$ (i.e., an additional $n_1 \times (n_1 - 1)$ variables $a_{(1,0),n}^s = 0$). Similarly, for all nodes $j \leq K + 1$, $n_1 - j + 1$ slots are required to forward packets toward node 1, during which the link $(1, 0)$ cannot be activated due to the K -hop interference constraint. We define

$$t_1 = \sum_{j=1}^{K+1} (n_1 - j + 1)$$

as the *total number* of slots required to collect all n_1 packets at node 0. If $n_1 > K + 1$, additional transmissions over links $(j, j - 1)$ are required for nodes $j > K + 1$. However, these transmissions do not interfere with those on the link $(1, 0)$ and hence are not considered in deriving the lower bound. Therefore, for a single line network, we obtain the lower bound $N \geq t_1 = N_{lb}$. $\square$

Multiline Networks: The above ideas are used to obtain the following lower bound for a multiline network.

Proposition 4.5. *Consider a multiline network with n_b nodes in every subtree b and K -hop interference. Any periodic deterministic policy satisfying PeDe has a period N bounded below by N_{lb} , where*

$$N_{lb} = e_x + \sum_{b \in \mathcal{R}_{top}} e_b,$$

with $e_b = \sum_{k=1}^{\lceil K/2 \rceil} [n_b - (k - 1)]$ and $e_x = \max_{b \in \mathcal{R}_{top}} (f_b - e_{b,o})^+$. The quantity $f_b = t_b - e_b, \forall b \in \mathcal{R}_{top}$, where $t_b = \sum_{j=1}^{K+1} (n_b - j + 1)$, and $e_{b,o} = \sum_{j \neq b} e_j$.

Proof. The proof (similar to that of Proposition 4.4) considers the optimization problem (4.8) but with relaxed constraints. We again impose the constraints (4.2) and (4.3) and completely remove the constraint (4.6). However, we impose the constraints (4.4) and (4.5)

only for the links $(i, 0)$, where $i \in \mathcal{R}_{top}$ is a top-subtree root (i.e., the links connecting the top-subtree roots to 0). To find a lower bound on the period, we again count the number of slots required so that we obtain variables $a_{(i,0),n}^s \in \{0, 1\}$ which can satisfy the constraints. In a multiline network, each top-subtree b (which has a line structure) consists of n_b nodes. Therefore, we need at least $\sum_b n_b$ slots (or an equivalent number of variables need to be set to 0 or 1 for $l = (i, 0), i \in \mathcal{R}_{top}$) to collect all the packets and satisfy (4.2) and (4.5). As in the case of the line network additional slots are required to forward these packets to the top-subtree roots, during which time the link $(i, 0)$ cannot be scheduled (and therefore the variables $a_{(i,0),n}^s$ are required to be 0). We define t_b as a lower bound on the total number of slots required to collect all n_b packets at node 0 from subtree b , where t_b corresponds to line b . Then, from Proposition 4.4 we have that

$$t_b = \sum_{j=1}^{K+1} (n_b - j + 1).$$

However, the total number of slots required to collect the packets is not $\sum_b t_b$ due to the following reason. Consider a node l_1 in a line b_1 and a link l_2 in a line b_2 such that both links are within K -hop of the top-subtree links of their respective lines. However, it can happen that l_1 and l_2 are more than K -hops away from each other. Then, transmissions on l_1 and l_2 can be concurrent. However, we note that if a node j in a line b_1 , with $1 \leq j \leq \lceil K/2 \rceil$, transmits to node $j - 1$, then any node i with $1 \leq i \leq \lceil K/2 \rceil$, whether in the same line b_1 or in any other line, cannot simultaneously transmit to node $i - 1$ due to the interference constraints. Therefore, the slots required for forwarding packets from such nodes j are *exclusive*, meaning that node 0 cannot receive packets from any line during these slots. The number of exclusive slots required for node j is equal to the number of packets it must forward (including its own packet), which is $n_b - j + 1$. We define e_b as the *total number of such exclusive slots* corresponding to line b . We therefore have,

$$e_b = \sum_{j=1}^{\lceil K/2 \rceil} (n_b - j + 1)$$

Recall that, for a single line, we defined t_b as the total number of slots required to collect all packets from line b . Consider the difference $f_b = t_b - e_b$. These correspond to slots during which node 0 cannot collect packets from line b . These slots correspond to transmissions from nodes or links in line b that may be more than K hops away from transmissions occurring in the exclusive slots of other lines. For example, consider two nodes located in

different lines whose hop distance exceeds $\lceil K/2 \rceil$. Such nodes can transmit concurrently within the same slot with no interference. Hence, it may be possible to schedule these transmissions during the exclusive slots of other lines. For a given line b , we define the number of *exclusive slots contributed by other lines* as $e_{b,o} = \sum_{j \neq b} e_j$. If, for a given line b , all f_b slots can be accommodated within $e_{b,o}$, then no additional slots beyond $e_b + e_{b,o} = \sum_{b'} e_{b'}$ are required. Otherwise, an additional $f_b - e_{b,o}$ slots are needed. Thus, in general, the number of additional (or *excess*) slots required for line b is given by $(f_b - e_{b,o})^+$. The maximum number of such excess slots across all lines is $e_x = \max_b ((f_b - e_{b,o})^+)$. Therefore, the overall lower bound on the period is given by the sum of the exclusive slots across all lines, $\sum_b e_b$, and the excess slots e_x .

$$N_{lb} = e_x + \sum_{b \in \mathcal{R}_{top}} e_b \quad (4.10)$$

These correspond to a lower bound on the number of variables $a_{(i,0),n}^s$ required to be 0 or 1 in order to satisfy the constraints (4.4) and (4.3). $\square$

Tree Networks: We now extend the above lower bound to general tree network topologies. We introduce additional notation and informal explanations for the proposition. Our approach again relaxes (4.6) completely in (4.8). We also impose (4.2) and (4.3), and impose (4.4) and (4.5) only on the links connecting top-subtree roots to 0 in (4.8). Recall that, for multiline networks, we defined exclusive slots for each line b . These exclusive slots e_b correspond to the number of slots required to collect packets from nodes that are within $\lceil K/2 \rceil$ hops of node 0 and correspond to the number of variables $a_{(i,0),n}^s \in \{0, 1\}$ required to satisfy the constraints. For a line in a multiline network, the number of packets to be collected from a node at hop distance h is $n_b - h$. For a general tree, let $n_b(h)$ denote the total number of nodes within h hops of node 0 in subtree b . Since the total number of packets in subtree b is n_b , the number of packets that must be forwarded from nodes at hop distance h is given by $n_b - n_b(h - 1)$. Using this characterization, we define the exclusive slots e_b for each subtree b in terms of $n_b(h)$. To define the excess slots for tree networks, we construct an equivalent multiline network representation of the tree. This multiline network has many lines as there are subtrees in the tree. For every subtree indexed by b , we construct a line λ_b in the equivalent multiline network. The line λ_b is assumed to contain as many nodes as the longest t-line in subtree b (if multiple t-lines have the same length, one is chosen arbitrarily). The length of λ_b is denoted by l_b (note that if the tree were a multiline to start with, then $l_b = n_b$). For each line λ_b , we assume that a node at hop distance h forwards

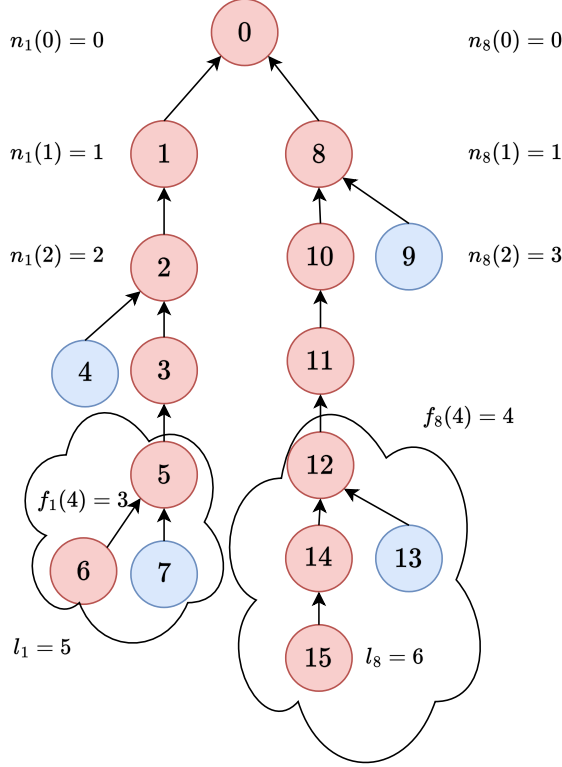


Figure 4.5: Example of a multihop tree network with 15 nodes, 2 top-subtrees, and 5-hop interference. Longest t -lines (red nodes) and their lengths are shown. The values of $n_b(h)$ and $f_b(h)$ for different hop distances are indicated for both subtrees; for example, in subtree 1, $n_b(1) = 1$ and $f_b(4) = 3$.

$f_b(h)$ packets (including its own) to a node at hop distance $h - 1$. In contrast to the multiline case discussed earlier, where a node at hop distance h forwards only the packets from its descendants (including its own), in the equivalent multiline network each node is assumed to forward $f_b(h)$ packets. We note that $f_b(h)$ can be computed as one more than the total number of descendants of the corresponding node n , since the node n must forward both its own packet and those of its descendants. The number of slots required to collect these packets under K -hop interference constraints can be determined using similar ideas as discussed earlier. Using $f_b(h)$, we define the excess slots e_x for the tree network. For illustration, both $n_b(h)$ and $f_b(h)$ are shown for an example tree network in Fig. 4.5. The longest t -lines in each subtree are highlighted using red nodes. Using the above definitions, we now obtain the following analytical lower bound for tree network topologies.

Proposition 4.6. *For a tree network with K -hop interference any policy satisfying PeDe has a period N bounded below by N_{lb} , where*

$$N_{lb} = e_x + \sum_{b \in \mathcal{R}_{top}} e_b,$$

with $e_b = \sum_{h=1}^{\lceil K/2 \rceil} (n_b - n_b(h-1))$, $e_x = \max_{b \in \mathcal{R}_{top}} (f_b - e_{b,o})^+$, and $n_b(h)$ represents the number of nodes in top-subtree b which are within a hop-distance of h from 0 ($n_b(0) = 0$). We have that $f_b = \sum_{h=\lceil K/2 \rceil+1}^{\min(l_b, K)} f_b(h)$ and $e_{b,o} = \sum_{j \neq b} e_j$. Here, l_b denotes the length of the longest t -line in the top-subtree b and $f_b(h)$ denotes the number of packets to be forwarded from a node at hop distance h from 0 in top-subtree b in its longest t -line.

We note that the proof is similar to that of Proposition 4.5 and is provided in Appendix 4.B. For the example network with 5-hop interference shown in Figure 4.5 we have that $n_1(0) = 0$, $n_1(1) = 1$, and $n_1(2) = 2$ for subtree 1. For the subtree rooted at 8 we have $n_8(0) = 0$, $n_8(1) = 1$, and $n_8(2) = 3$. We note that we consider nodes up to a hop distance of 3 in $e_b = \sum_{h=1}^{\lceil 5/2 \rceil} n_b - n_b(h-1)$. Using this definition of e_b , we have that $e_1 = (7-0) + (7-1) + (7-2) = 18$, while $e_8 = (8-0) + (8-1) + (8-3) = 20$. Using $f_b(h)$, we obtain that $f_1 = f_1(4) + f_1(5) = 4$ and $f_8 = f_8(4) + f_8(5) + f_8(6) = 4 + 2 + 1 = 7$. Since the slots for f_1 transmissions are assumed to be concurrent with the e_8 slots and f_8 with e_1 , we have that the excess slots e_x is 0. Therefore, we obtain that $N_{lb} = e_1 + e_8 = 38$ for this example.

We note that an analytical lower bound on the period of a periodic deterministic policy for a multichannel 2-hop interference model was derived in [115].¹⁰ In contrast, our result applies to a single-channel K -hop interference model.

Proposition 4.7. *The minimum AAoI A_{avg}^* is bounded below by $\left(\frac{1}{M} \sum_{s=1}^M h_s\right) + \frac{N_{lb}-1}{2}$, where h_s is the hop distance of source s from 0 and N_{lb} , from Proposition 4.6, is the lower bound on the period for any policy.*

Proof. The proof is direct and is outlined here. We note that A_{avg} can be characterized using AAoI(s) from Proposition 4.1. We bound each of the terms d_s and N from below. For any policy that satisfies PeDe(s), the delivery delay $d_s \geq h_s$ and $N \geq N_{lb}$ from Proposition 4.6 which gives the result. $\square$

We note that Farazi et al. [44] derive a lower bound on AAoI for a network in which each node acts as both a source and a sink for all other nodes, with the additional constraint

¹⁰Gandham et al. [115] assume that different top-subtrees operate on different channels.

that only one node can transmit in a slot (i.e., a complete interference model). Their result coincides with ours for the complete interference case (see also Proposition 4.2). In contrast, the above result for K -hop interference models is more general for convergecast scenarios.

4.4 Periodic Deterministic Sampling and Scheduling Policies for Multihop Networks

In this section, we propose periodic deterministic sampling and scheduling policies for tree networks under K -hop interference constraints that satisfy PeDe(s) for all $s \in [M]$. We first present the algorithm for a simple line network and then extend it to general tree network topologies.

4.4.1 Sampling and Scheduling Algorithm for Line Networks

We propose a sampling and scheduling algorithm for a line network with M source nodes, summarized in Algorithm 1. The algorithm generates a sequence of activation sets $\mathcal{U}_N = (U(0), U(1), \dots, U(N-1))$ to be used within a single period. The policy is implemented by periodically repeating $\mathcal{U}_N$ with period N . In the following, we discuss the objectives and the basic principles underlying Algorithm 1.

Objective of Algorithm 1: Algorithm 1, based on Gandham et al. [115], generates a scheduling policy for a line network and satisfies the following properties:

1. The sequence of activation sets satisfies constraints (4.2), (4.3), (4.4), (4.5), (4.6). Constraints (4.4), (4.5), and (4.6) must hold in every slot, while constraints (4.2), and (4.3) must hold over the period.
2. The policy is periodic and deterministic.
3. The policy aims to minimize A_{avg} as much as possible.

We now discuss the basic principles that enable Algorithm 1 to satisfy the above properties, along with references to specific implementation details. For easier reference, we also assign names to these principles.

P1–(4.4), (4.5): We adopt the approach from [115] to ensure that Algorithm 1 generates a sequence of activation sets satisfying constraints (4.4) and (4.5). First, we discuss how one activation set that satisfies (4.4) and (4.5) can be generated.

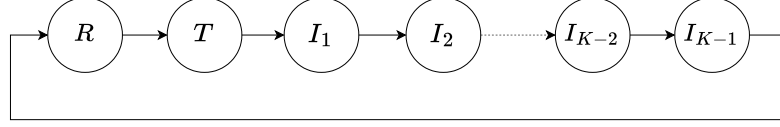


Figure 4.6: State-transition diagram of node states under K -hop interference ($K \geq 2$).

Suppose a link $(i, i - 1)$ is activated, i.e., a transmission occurs from node i to node $i - 1$ in the line network. We define node i to be in a *transmit state*, denoted by T , and node $i - 1$ to be in a *receive state*, denoted by R . These states correspond to the actual physical states of the node radios. In Algorithm 1, nodes are restricted from being simultaneously in T and R . Hence, the half-duplex constraint (4.4) is satisfied.

We now discuss how an activation set satisfying constraint (4.5) can be constructed using node states. Consider a line network with 4 sources, i.e., $0 - 1 - 2 - 3 - 4$, under 2-hop interference. An example activation set that satisfies the interference constraints consists of the link activations $(1, 0)$ and $(4, 3)$. In this case, nodes 0 and 3 are in the receive state R , while nodes 1 and 4 are in the transmit state T . Due to the K -hop interference constraint, nodes within $K - 1$ hops of a node in state R or T cannot transmit or receive; such nodes are set to an *idle state* I . In our example, node 2 is in the idle state I , which can be physically mapped to a state in which the radio is idle or in sleep mode. For a line network, the node state sequence follows a periodic pattern along the node sequence. We start with R, T for nodes 0 and 1, followed by $K - 1$ idle states, and then repeat this pattern along the line. This node state sequence is determined by the hop distance. The initial pattern of states, corresponding to the initial activation set that satisfies (4.5), is generated in lines 2 and 3 of Algorithm 1.

We note that the initial activation set obtained above generates and transmits packets from 1 and 4. We also need other activation sets to generate, transmit, and forward packets from these and other nodes to 0. These other activation sets which satisfy (4.4), and (4.5) can be generated from the above initial activation set by changing the pattern of the node states. For example, if we set 1 and 2 to have the states R and T and use the periodic pattern mentioned above we obtain another activation set. Equivalently, we can generate these states by evolving the state of each node according to one step of the transition diagram shown in Figure 4.6. After one step, we obtain an activation set (for example R, T, I, R, T evolves to T, I, R, T, I) that satisfies both (4.4), and (4.5). If we use two steps of the transition for every node we obtain another activation vector and so on. By this scheme we can generate $K + 1$ different activation sets (including the initial activation set that we

started out with). All these activation sets satisfy (4.4), and (4.5). In Algorithm 1, at the end of every slot, we apply a step of the state-transition for every node (see line 17). For K -hop interference model, we use $K - 1$ different idle states denoted as $I_1, I_2, \dots, I_{K-1}$, where $K \geq 2$ (for $K = 1$, there is no idle state). Thus, in every $K + 1$ slots, we obtain the $K + 1$ activation sets satisfying (4.4), and (4.5) at every slot. The activation sets are then repeated. This is illustrated for an example in Figure 4.7.

P1-(4.6): Consider the sequence of activation sets generated as described in P1-(4.4),(4.5). We note that if, in a given slot, nodes $i + 1$ and i have states T and R (i.e., link $(i + 1, i)$ is active), then in the next slot nodes i and $i - 1$ will have states T and R , respectively, followed by $i - 1$ and $i - 2$, and so on. Once node $i + 1$ generates a packet (line 12 of Algorithm 1), it is forwarded in consecutive slots (line 10) until it reaches node 0. Hence, the scheduling policy does not attempt to transmit a packet before it is generated and satisfies constraint (4.6).

P1-(4.2),(4.3): From lines 10 and 12 of Algorithm 1, it is evident that each source generates at most one packet. As discussed in P1-(4.6), any packet that is generated is delivered to node 0 in consecutive slots. Consequently, constraints (4.2) and (4.3) are also satisfied. A more formal justification is provided in Proposition 4.8.

P(4.2): We note that Algorithm 1 is deterministic. Moreover, it generates a finite-length sequence of activation sets (see line 6). This sequence is then repeated to obtain a periodic scheduling policy.

P(4.3): From Proposition 4.1, we have $\text{AAoI}(s) = d_s + (N - 1)/2$. One way to minimize A_{avg} is to jointly minimize d_s for all $s \in [M]$ and the period length N . We note that the minimum value of d_s is the hop distance h_s of source s . For the sequence of activation sets generated by Algorithm 1, any packet, once generated, is forwarded in consecutive slots until it reaches node 0. Therefore, we have $d_s = h_s$. Algorithm 1 heuristically attempts to reduce N by maximizing the number of link activations in each slot, i.e., by using *maximal* activation sets. Further minimization of N is achieved through the following improvement to Algorithm 1.

P(4.3)-Improvement: We observe that the following modification to Algorithm 1 improves A_{avg} performance. In any slot, consider the activation set generated by Algorithm 1, which corresponds to a specific set of node states. If the activation set results in no packet transmissions as occurs, for example, in slot 4 of Figure 4.7, we modify line 17 of the algorithm. Specifically, if the state update produces an activation set without any transmissions, additional updates are performed to ensure that at least one packet is transmitted in the slot. In Figure 4.7, this modification effectively skips the activation set in slot 4, reducing the

period from 7 to 6 slots. The improved Algorithm 1 thus heuristically attempts to minimize AAoI as much as possible.

Explanation of Algorithm 1: In lines 2-3, the algorithm initializes the node states corresponding to the first activation set that satisfies constraints (4.4) and (4.5) (see P1-(4.3), (4.5)). Then, in each slot, lines 8-16 populate the activation set $U(t)$ based on the current node states. In line 17, the next activation set is generated, and the process repeats until packets from all sources are produced and successfully forwarded to the sink.

Algorithm 1 Algorithm to generate $\mathcal{U}_N$ for a line network

Initialization:
2: Initialize $i.state$ according to a value m_i which is [hop distance of i from 0] modulo $(K + 1)$
We set $i.state$ to be R if $m_i = 0$, T if $m_i = 1$, and I_{m_i-1} if $m_i \geq 2$
4: Initialize slot index $t = 0$
Initialize policy $\mathcal{U}_N = []$
6: **while** slot index $t \leq (K + 1)(M - 1) + 1$ **do**
Initialize activation set $U(t) = []$
8: $S = \{(i, j) \in \mathcal{E} \text{ s. t. } i.state = T, j.state = R\}$
for $(i, j) \in S$ **do**
10: **if** i has a packet from a source s **then**
append $(s, (i, j))$ to $U(t)$
12: **else if** i has not generated a packet **then**
append $(i, (i, j))$ to $U(t)$
14: **end if**
end for
16: append $U(t)$ to $\mathcal{U}_N$
For every i , update $i.state$ according to Figure 4.6
18: $t \leftarrow t + 1$
end while

An example policy for a line network with $M = 3$ under a $K = 2$ hop interference constraint, with node states T , R , and I_1 , is illustrated in Figure 4.7 for a single period. The same sequence of activation sets, $\mathcal{U}_N$, is then repeated periodically. We refer to this policy as SS-Line. Algorithm 1 generates $\mathcal{U}_N$ in $(K + 1)(M - 1) + 1$ steps, where each step (lines 7-18 of the algorithm) requires at most $\mathcal{O}(M)$ operations. Hence, the overall computational complexity of SS-Line is $\mathcal{O}(KM^2)$. Additional properties of the policy are stated in the proposition below.

Proposition 4.8. *Algorithm 1 generates a periodic policy (SS-Line) with period $(K + 1)(M - 1) + 1$, which satisfies constraints (4.4)-(4.6) as well as the PeDe property.*

The proof is presented in Appendix 4.C. We now extend Algorithm 1 to tree networks by applying it to individual lines (t-lines) within the tree. The resulting algorithm for generating scheduling policies in tree networks is given in Algorithm 4. For clarity, we first

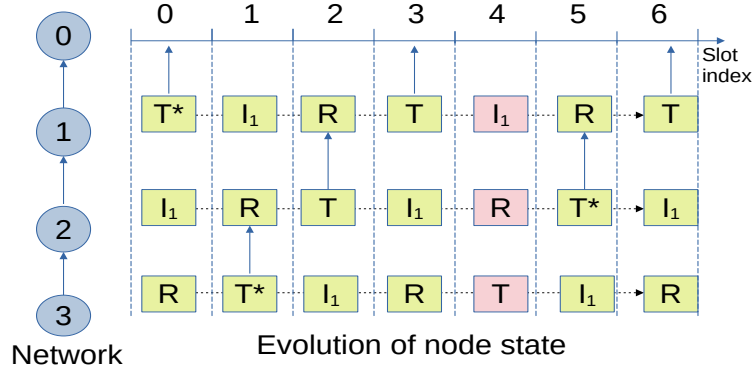


Figure 4.7: Example schedule for a line network with $M = 3$ and $K = 2$ -hop interference constraint. The states of each node are T , R , and I_1 . The T^* state also denotes packet generation. The arrows show packet transmission. Algorithm 1 also produces the activation set in slot 4. In Improved Algorithm 1 this activation set is skipped, since there are no actual transmissions in this slot, in order to reduce the period from 7 to 6.

discuss the extension for two specific tree topologies: the multiline topology and a tree with a single top-subtree. In this discussion, the parent of a node v is denoted by $p(v)$.

4.4.2 Sampling and Scheduling Algorithm for Multiline Networks

Initial Idea: A straightforward extension to multiline networks proceeds as follows. We first select a t-line, typically the one with the largest number of packets to be transmitted, and apply Algorithm 1 to generate the sequence of activation sets for sampling and forwarding packets along that t-line. No other t-line is scheduled until all packets from the selected t-line have reached node 0. Once all packets from the current t-line are delivered, another t-line with the largest remaining number of packets is selected, and the same procedure is repeated. The sequence of activation sets generated for each t-line is appended to the sequences from previously scheduled t-lines. This approach produces a policy that transmits one packet from each source per period while satisfying the K -hop interference constraints.

Improved Policy: It is intuitive that the policy described above can be *improved* by allowing concurrent transmissions on different t-lines while still satisfying the K -hop constraints. (Note that within a single t-line, the activation sets already satisfy the K -hop constraints by evolving node states according to Figure 4.6.) Consider a pair of t-lines whose nodes have states in T, R, I_k . We say that a pair of t-lines is *compatible* in a given slot if the (T, R) transmissions along both t-lines satisfy the K -hop constraints. Compatible t-lines can then be scheduled simultaneously. In Algorithm 4, this is implemented as

follows. In each slot, we first select a t-line (or top-subtree) to serve as the *base*, and then identify any other t-lines (or top-subtrees) that are compatible with it.

Finding compatible t-lines: Compatible t-lines that can be scheduled along with the base are identified using the CanAddToBase function (Algorithm 2). The inputs to this function are base and b , where b is another t-line. Since the node states in both t-lines are initialized using the same procedure as in Algorithm 1 and evolve deterministically, compatibility between base and b can be determined by examining the states of their top-subtree roots. Using the transition scheme in Figure 4.6, we can identify at which hop distance the nodes on both top-subtrees would be in state T . If the link-hop distance between these nodes is less than or equal to K , concurrent transmissions are not allowed, and the two top-subtree roots are considered incompatible; otherwise, they are compatible. For example, with $K = 2$, the compatibility of two t-lines depends on the states of their top-subtree roots. If the top-subtree root of base is in state R , then b 's top-subtree root can be in state R or I_1 for the two t-lines to be compatible. If base's root is in state T , b is compatible only when its root is in state I_1 . Finally, if base's root is in state I_1 , b is compatible if its root is in state R , T , or I_1 . Determining the set of compatible t-lines requires $\mathcal{O}(K^2)$ time, since each possible state of base's root must be checked against every possible state of b 's root. If CanAddToBase returns True, b can be scheduled concurrently with base.

Algorithm 2 Function - CanAddToBase

```

function CANADDTOBASE(base, b)
2:    $s \leftarrow$  base's top-subtree-root.state
   if  $b$ 's top-subtree-root.state  $\notin$  compatible-set( $s$ ) then
4:     Return False
   else
6:     Return True
   end if
8: end function

```

4.4.3 Sampling and Scheduling Algorithm for Tree Networks with a Single Top-Subtree

We have discussed how t-lines in a multiline network can be scheduled. A single top-subtree, however, consists of multiple t-lines, raising the question of how these t-lines should be scheduled within the subtree. To illustrate how Algorithm 1 can be extended to this case, we consider an example of a single top-subtree.

Initial Idea: Consider a top-subtree with two t-lines: T1, consisting of nodes $0, 1, \dots, c, c+1, \dots, c+n_1$, and T2, consisting of nodes $0, 1, \dots, c, c+n_1+1, \dots, c+n_1+n_2$. The t-lines share the nodes $0, 1, \dots, c$ as common nodes, and the total number of nodes is $M = c+n_1+n_2$. A simple heuristic is to first generate a sequence of activation sets to transmit all packets from T1 using Algorithm 1. In the slot when the last packet from T1 reaches node 0, the state of node 1 would be T , while the states of nodes $1, 2, \dots, c, c+n_1+1, \dots, c+n_1+n_2$ would match the initial states for T2 under Algorithm 1. A sequence of activation sets for T2 can then be generated starting from that slot. Under this heuristic, all packets from one t-line are transmitted completely before scheduling the next t-line, with the next t-line being selected in the final slot of transmission for the previous t-line. However, during the phase when Algorithm 1 is applied to T2, the common nodes $1, 2, \dots, c$ do not generate new packets, even though the link activations include packet-generation decisions for these nodes. It is therefore intuitive that the heuristic can be improved by *activating* the next t-line, T2, before all packets from T1 are transmitted. We discuss this improvement next.

Improved Policy: A t-line is said to be *active* or *activated* during a time slot if at least one link (i, j) in that t-line, satisfying $i.\text{state} = T$ and $j.\text{state} = R$, is included in the activation set for that slot. Subsequently, the states of all nodes in the t-line evolve according to the state-transition diagram shown in Figure 4.6.

To improve the policy, we use a heuristic based on Gandham et al. [115]. The heuristic waits only until the packets from nodes $c+1, c+2, \dots, c+n_1$ reach node c (or nodes closer to node 0), and the states of nodes $1, 2, \dots, c$ match the initial states of the nodes for T2 under SS-Line, before activating T2. From the proof of Proposition 4.8, SS-Line requires at most $n_1(K+1)$ slots to satisfy this condition. Rather than counting the number of slots, we instead count the number of packets forwarded from the subtree root to node 0. Suppose that the number of forwarded packets becomes strictly greater than n_1 , and consider the first slot at which this occurs. Under SS-Line, this requires approximately $1 + n_1(K+1)$ slots, by which time all packets from nodes $c+1, c+2, \dots, c+n_1$ would have reached node c or beyond, and the states of nodes $1, 2, \dots, c$ would match the initial node states required for T2. To implement this idea in SS-Tree, we first initialize an activation threshold for each t-line. This threshold denotes the number of packets that must be forwarded from the t-line's top-subtree root before the t-line is activated. Subsequently, while constructing the activation sets, a t-line is activated whenever the number of forwarded packets exceeds its activation threshold. The activation threshold initialization procedure is described in Algorithm 3, where the required number of packets to be forwarded is computed recursively. In the above example, T2 must wait for n_1 packets, which is equal to the number of sources in

T1 that are descendants of the common parent. This value serves as the activation threshold for all nodes after c in T2. If the t-lines rooted at a subtree root are arranged in some order (for example, by node id), then the activation threshold for a given t-line is equal to the total number of descendants in all other t-lines that appear earlier in the ordering. The function `InitializeActiveThreshold` described in Algorithm 3, recursively computes the number of descendants required to set the `node.activethreshold`, starting from the top-subtree root.

Algorithm 3 `InitializeActiveThreshold`

```

function INITIALIZEACTIVETHRESHOLD(node, num_packets)
2:   node.activethreshold = num_packets
   increment  $\leftarrow$  0
4:   for  $c$  in children of node do
       InitializeActiveThreshold( $c$ , num_packets + increment)
6:   increment  $\leftarrow$  increment + number of descendants of  $c$ 
   end for
8: end function

```

4.4.4 Sampling and Scheduling Algorithm for Tree Networks

The Algorithm 4 outlines the procedure used to generate the sequence of activation sets. The procedure is divided into two phases: the first phase performs the initialization, and then the second phase generates the periodic activation sequence $U(n)$.

In the initialization stage, each node is assigned its initial state, the packet generation indicator flag (`pflag`) is set to false, and the activation threshold value is initialized. The initial node states are determined by the function `InitializeState`, which assigns each node to either the transmit state T , the receive state R , or one of the $K - 1$ idle states, namely $I_1, I_2, \dots, I_{K-1}$ for $K \geq 2$. This initialization follows the same state-assignment principle used in SS-Line and is based on the hop distance of the node from the root (see lines 2-3 of Algorithm 1). The `pflag` variable is used to ensure that each node generates at most one packet within a single period. For a top-subtree b , $b.nb$ denotes the total number of nodes in b , while $b.pb$ is used to keep track of the number of packets that have not yet been transmitted.

In the next phase, the activation sets are generated. The algorithm continues to execute as long as there are untransmitted packets remaining in the tree (line 7). At the beginning of each slot, all top-subtrees are initially marked as inactive (line 8). As discussed earlier, a *base* top-subtree is selected. The same base top-subtree is retained from the previous slot until all packets in that top-subtree have been transmitted. If no base top-subtree has been assigned, then the algorithm selects the top-subtree containing the largest number of

untransmitted packets as given in lines 10-12 (Algorithm 4). Thereafter, additional top-subtrees are activated. To achieve this, we propose two activation methods. In the first method, termed SS-Tree, the other top-subtrees b in decreasing order of $b.pb$ (i.e., their untransmitted packet count), and those that are compatible are included in the activation set (lines 14-18). In the second method, referred to as SS-Tree with *Continue-Service* denoted as CS, priority is given to the top-subtrees that were activated in previous slots (see lines 19-24), followed by the consideration of the remaining top-subtrees (lines 25-29). The top-subtrees that have been activated during the previous $K + 1$ slots are maintained in the set `PreviouslyScheduledSubtrees`. Finally, in lines 31-35, the activation sets are determined, and the states of all active top-subtrees are updated using the function `ScheduleAndStateUpdate` (see Algorithm 5). In `ScheduleAndStateUpdate` (Algorithm 5), the top-subtree root is considered first. If its state is T , it either generates a packet, after checking through `pflag`, that no packet has already been generated in the current period, or forwards a packet if one is available in its buffer (see lines 4-13). The corresponding link transmission is then added to the activation set for slot $U(t)$ in line 13. Next, each descendant of node v is examined to determine whether it can be activated based on its activation threshold. If a descendant node d is active and its state is T , it can either generate a packet or forward a buffered packet. Likewise, the corresponding link transmission is included in the activation set for slot $U(t)$ in line 25. The states of all active nodes are then evolved by one step according to the state-transition diagram as in Figure 4.6.

Discussion - SS-Tree and CS satisfy PeDe with a finite period: As long as there are untransmitted packets in any top-subtree, Algorithm 4 always selects some top-subtree as the base (see lines 8-12). In the worst case, only the base top-subtree remains active when no other top-subtree is compatible with it. Within the base, the t-lines are activated sequentially so that packets are forwarded from all nodes to the top-subtree root (see the discussion on the activation threshold). An upper bound on the number of slots required to forward all packets from a top-subtree b is $(K + 1)n_b$. Therefore, in the worst case, the total number of slots required is bounded above by $(K + 1)M$. We use this bound below to characterize the algorithmic complexity of Algorithm 4.

Proposition 4.9. *The time complexity of Algorithm 4 is $\mathcal{O}(K^2 M^2 + K M^2 \log(M))$.*

Proof is given in Appendix 4.D.

Remark 4.1. We note that Algorithm 1 can be further improved, as discussed in P(4.3)-Improvement. Similar improvements are implemented into Algorithm 4, as described below.

Algorithm 4 Algorithms to generate $\mathcal{U}_N$ for a tree network

Initialization:

2: $\forall v \in \mathcal{V}$, InitializeState(v), $v.buffer \leftarrow 0$, $v.pflag \leftarrow \text{False}$
 $\forall b \in \mathcal{R}_{top}$, $b.pb$ = number of sources in b 's subtree, $b.nb = b.pb$

4: $\forall b \in \mathcal{R}_{top}$, InitializeActiveThreshold(b , 0)
 $t \leftarrow 0$, $base \leftarrow \phi$, PreviouslyScheduledSubtrees $\leftarrow []$

6: Initialize policy $\mathcal{U} = []$

Sampling and Scheduling:

8: **while** any $b.pb > 0$ for $b \in \mathcal{R}_{top}$ **do**
 $\forall b$, $b.active = \text{False}$

10: **if** $base = \phi$ **then**
 $base = \arg \max_b \{b.pb\}$

12: **end if**
 $base.active = \text{True}$

14: **if** Algorithm is SS-Tree **then**
 for $b \in \mathcal{R}_{top} \setminus \{base\}$, ordered decreasingly by $b.pb$ **do**

16: **if** CanAddBase($base$, b) and $b.pb > 0$ **then**
 $b.active = \text{True}$

18: **end if**

20: **end for**

20: **else** ▷ Algorithm is CS
 for $b \in \text{PreviouslyScheduledSubtrees} \setminus \{base\}$, ordered decreasingly by $b.pb$ **do**

22: **if** CanAddBase($base$, b) and $b.pb > 0$ **then**
 $b.active = \text{True}$

24: **end if**

26: **end for**

26: **for** $b \in \mathcal{R}_{top} \setminus \{\text{PreviouslyScheduledSubtrees}\}$, ordered decreasingly by $b.pb$ **do**

28: **if** CanAddBase($base$, b) and $b.pb > 0$ **then**
 $b.active = \text{True}$

30: **end if**

32: **end for**

32: **for** $b \in \mathcal{R}_{top}$ **do**
 if $b.active = \text{True}$ **then**

34: $U_b(t) \leftarrow \text{ScheduleAndStateupdate}(b, t)$
 Append $U_b(t)$ to $\mathcal{U}$

36: **end if**

38: **end for**

40: **if** $base.pb = 0$ **then**
 $base = \phi$

42: **end if**
 Update PreviouslyScheduledSubtrees
 $t \leftarrow t + 1$

end while

Algorithm 5 ScheduleAndStateupdate(b, t)

```
function SCHEDULEANDSTATEUPDATE( $b, t$ )
2:   Initialization:
    $v \leftarrow$  top-subtree root of  $b$ 
4:   Initialize activation set  $U(t) = []$ 
   if  $v.state = T$  and  $v.pflag = \text{False}$  and  $v.bufferlength = 0$  then
6:      $v$  generates a packet
      $v.pflag = \text{True}$ 
8:      $v.bufferlength = v.bufferlength + 1$ 
   end if
10:  if  $v.state = T$  and  $v.buffer > 0$  then
    $v$  forwards the head-of-queue packet  $pk$  in its buffer to  $p(v) = 0$ 
12:   $v.bufferlength = v.bufferlength - 1$ 
    $v.pb = v.pb - 1$ 
14:  Let  $s$  be the source for  $pk$ . Append  $(pk, (v, p(v)))$  to  $U(t)$ 
   end if
16:  for each descendant  $d$  of  $v$  do
   if  $v.nb - v.pb > d.ActivityThreshold$  then
18:    if  $d.state = T$  and  $d.pflag = \text{False}$  and  $d.bufferlength = 0$  then
      $d$  generates a packet
20:     $d.pflag = \text{True}$ 
      $d.bufferlength = d.bufferlength + 1$ 
22:    end if
     if  $d.state = T$  and  $d.bufferlength > 0$  then
24:        $d$  forwards the head-of-queue packet  $pk$  in its buffer to its parent  $p(d)$ 
        $d.bufferlength = d.bufferlength - 1$ 
26:       Let  $s$  be the source for  $pk$ . Append  $(pk, (d, p(d)))$  to  $U(t)$ 
     end if
28:    end if
   end for
30:   $v.state = \text{updateState}(v.state)$ 
   for each descendant  $d$  of  $v$  do
32:    if  $v.nb - v.pb > d.ActivityThreshold$  then
      $d.state = \text{updateState}(d.state)$ 
34:    end if
   end for
36:  Return  $U(t)$ 
end function
```

Specifically, P(4.3)-Improvement is applied before line 13 of Algorithm 4. At this stage, we check whether the current sequence of node states in the base top-subtree allows at least one packet transmission. If no transmission is possible, multiple state-transition steps are applied to every node, following the transition diagram in Figure 4.6, until a sequence of node states permitting at least one transmission is obtained. Since the value of pb for the base top-subtree is positive, we note that such a state sequence must always exist. We also apply P(4.3)-Improvement to the other t-lines b before they are activated. First, we note in Algorithm 4, the lines 16-18, 22-24, and 27-29 are same. These lines implement the addition of t-lines other than the base. To implement P(4.3)-Improvement, each of these are replaced by the following logic:

```

1: if CanAddBase(base,  $b$ ) and  $b.pb > 0$  then
2:   if  $b$  has a packet transmission then  $b.active = \text{True}$ 
3:   else
4:     Update state for every node in  $b$ 
5:     if sequence of states has repeated then
6:       Stop
7:     else Goto step 1
8:     end if
9:   end if
10: end if

```

We first check whether a compatible t-line b to be added has at least one packet transmission, as in line 2 above. If this condition is satisfied, the t-line is set to be active.

Otherwise, the node states are updated iteratively (line 4) until one of the following conditions is met: (a) the sequence of node states repeats, indicating that no state sequence exists in b that is both compatible with the base and allows at least one packet transmission; in this case, the procedure stops (line 6), and another t-line b is considered, or (b) a sequence of node states is found that is compatible and permits at least one packet transmission.

4.5 Simulations and Performance Analysis

In this section, we present the A_{avg} performance of the proposed heuristic policies. The improved versions of these heuristic policies (see Remark 4.1) are considered throughout the analysis. The obtained average age of incorrect information is compared against the numerical lower bound, denoted by $A_{\text{glpk-lb}}$, as well as the analytical lower bound derived in Proposition 4.7. In addition, the A_{avg} performance of the proposed periodic determinis-

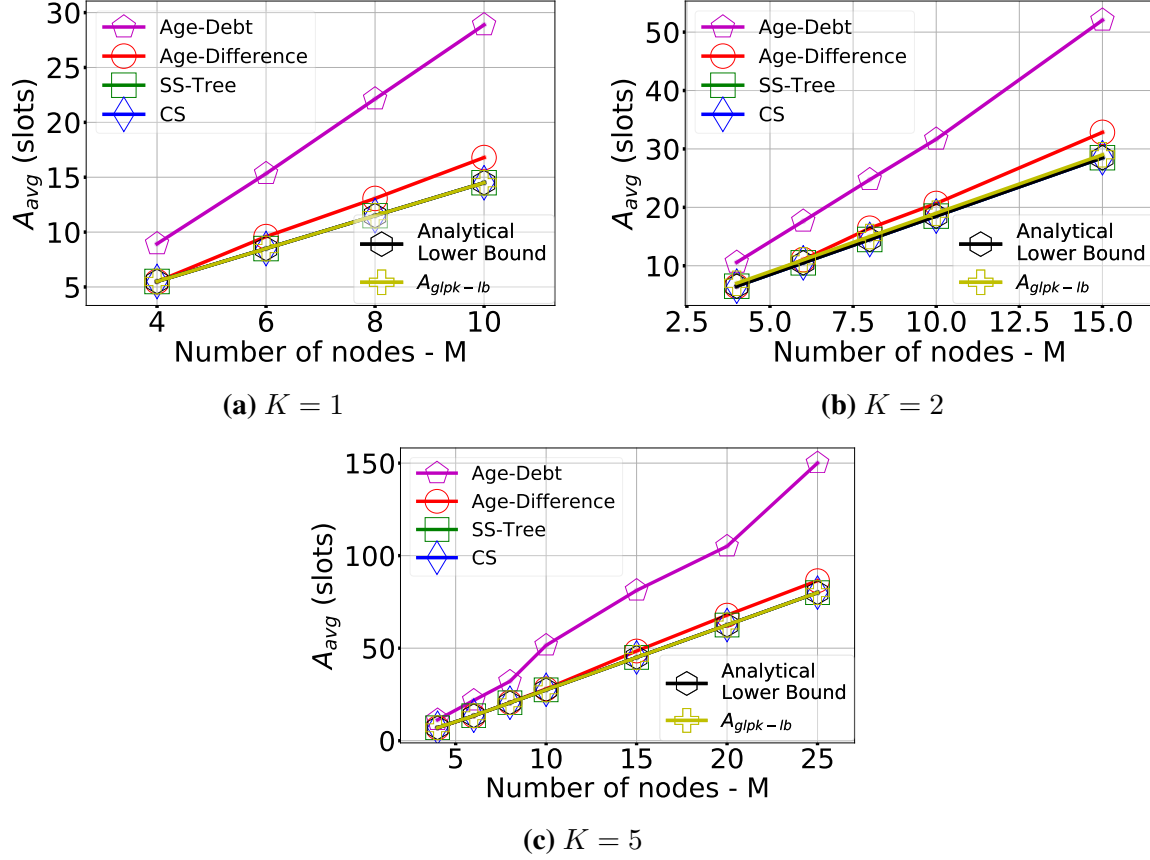


Figure 4.8: Simulated AAOI for line networks under SS-Tree, CS, Age-Difference, and Age-Debt policies. Results are compared with the analytical lower bound and $A_{glpk-lb}$ for varying M : (a) $K = 1$, (b) $K = 2$, and (c) $K = 5$.

tic policies is compared with existing policies from prior work, namely the Age-Difference policy and the Age-Debt policy proposed in [9]. The simulations and performance analysis are carried out for different network topologies, such as line, multiline, and tree networks, as well as for different values of K in the K -hop interference model and varying total numbers of nodes M . The periodic policies SS-Tree and CS are simulated over one complete period to obtain $d_s, \forall s$, and the period length N , based on which the AAOI is computed. The resulting simulated values of A_{avg} over one period provide easily computable upper bounds on A_{avg}^* for periodic deterministic policies. The A_{avg} values for the Age-Difference and Age-Debt policies are obtained through simulations over 5000 slots. We note that Age-Difference and Age-Debt are generally non-periodic or non-deterministic policies¹¹.

¹¹Certain implementations of the Age-Difference and Age-Debt policies are non-deterministic, where the activation set is selected randomly from the collection of sets that minimize the scheduling metric. Even in deterministic implementations without such randomization, our simulations indicate that, in many cases, no periodic pattern can be identified in the resulting scheduling decisions. Although periodic behavior is

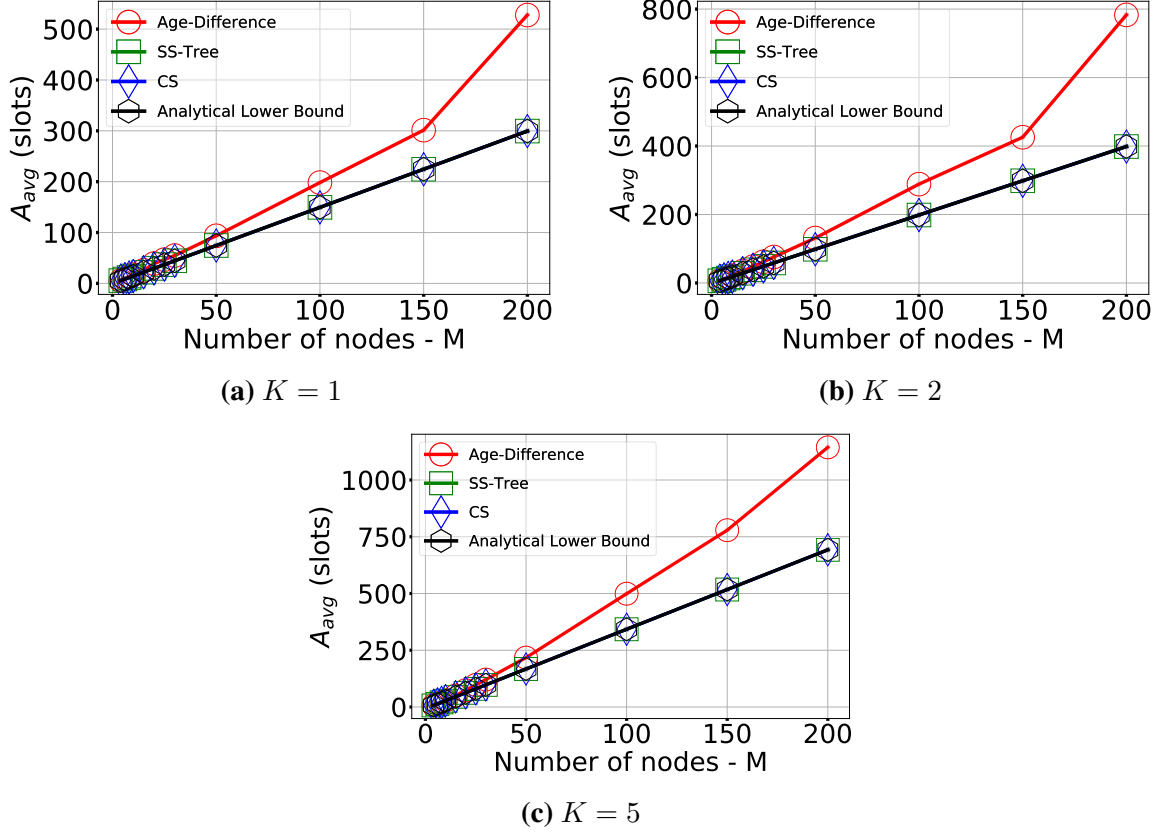


Figure 4.9: Simulated AAOI for line networks under SS-Tree, CS, and Age-Difference policies for larger values of M : (a) $K = 1$, (b) $K = 2$, and (c) $K = 5$.

In Figures 4.8(a)-(c) we compare A_{avg} of SS-Tree and CS with Age-Difference and Age-Debt as well as the analytical lower bound in Proposition 4.5 and GLPK lower bound ($A_{\text{glpk-lb}}$ from Section 4.3.1) for line networks as a function of M for $K \in \{1, 2, 5\}$. We note that, in all cases, the performance of SS-Tree and CS is the same, as expected, since there is only a single line, and they closely match the numerical and analytical lower bounds, which implies that they are close to optimal. For all cases, the periodic policies have a smaller A_{avg} compared to Age-Difference and Age-Debt. The percentage difference between Age-Difference and SS-Tree (or CS) increases with increasing M . Age-Debt is observed to achieve a larger AAOI compared to Age-Difference policies in every case considered. The Age-Debt policy considered here is the non-learning variant that employs flow control with parameters $\alpha_{\text{max}} = 5M$ and $V = 183$ (see [9]). We also examined learning-based approaches for selecting appropriate α values for each source node based

observed in certain scenarios, these policies are, in general, treated as non-periodic because there is no identifiable period.

on the corresponding age values obtained from the Age-Difference policy. In addition, we considered assigning identical α values to all source nodes, chosen as the average age under the Age-Difference policy. However, neither of these learning-based approaches resulted in improved performance of the Age-Debt policy compared to the results reported in [9]. In contrast, the parameter choice $\alpha_{\max} = 5M$ and $V = 183$ successfully reproduced the results presented in [9]. In Figures 4.8(a)-(b), smaller values of M are considered, since obtaining the numerical lower bound $A_{glpk-lb}$ and enumerating the complete set of activation sets required for the Age-Difference and Age-Debt policies become computationally challenging. In Figures 4.9(a)-(c), we compare the performance of the Age-Difference, SS-Tree, and CS policies against the analytical lower bound for larger values of M . The Age-Difference policy selects an activation set in each slot according to [9, Eq. 13], given by $\arg \max_{m \in \mathbb{U}} (\text{age-difference}(m))$, where $\text{age-difference}(m)$ denotes a metric that depends on the activation set m . For larger values of M , instead of searching over the full set $\mathbb{U}$, we restrict the choice of m to the set of activation sets generated by the SS-Tree or CS policies. This significantly reduces the computational burden associated with enumerating the complete activation set space. Again, we observe that the AAoI performance of the Age-Difference policy is worse compared to that of SS-Tree and CS. For $M = 200$, the percentage increase in AAoI relative to SS-Tree (or CS) is 76.2%, 96.5%, and 65.2% for $K = 1$, $K = 2$, and $K = 5$, respectively, as shown in Figures 4.9(a)-(c). The optimality is expected as the state-transition based policy (see Fig. 4.6) would achieve the minimum N and since our policy schedules packets contiguously, the delivery delay is the same as hop-distance.

Next, we consider multiline network topologies, as shown in Figures 4.10 and 4.11. In our simulations, random multiline networks are generated as follows. First, a fixed value of M is chosen. For each value of M , the nodes are assigned to B subtrees by sampling the subtree sizes n_b from a multinomial distribution with uniform probabilities, subject to the constraint that the total number of nodes equals M . In this study, we set $B = 2$. For each value of M , we randomly generate 100 network instances. For every such network and for each policy under consideration, we compute A_{avg} . A scatter plot of the points (M, A_{avg}) is then used to illustrate the variation of A_{avg} across different networks for each policy. In addition, we plot the mean value of A_{avg} , averaged over the 100 network instances, as a function of M for each policy. These mean values are shown as thick lines, with markers indicating the corresponding average age at each value of M . As before, smaller values of M are considered in Figures 4.10(a)-(c). For all values of K , we observe that the proposed analytical lower bounds closely match $A_{glpk-lb}$. However, we observe that the proposed

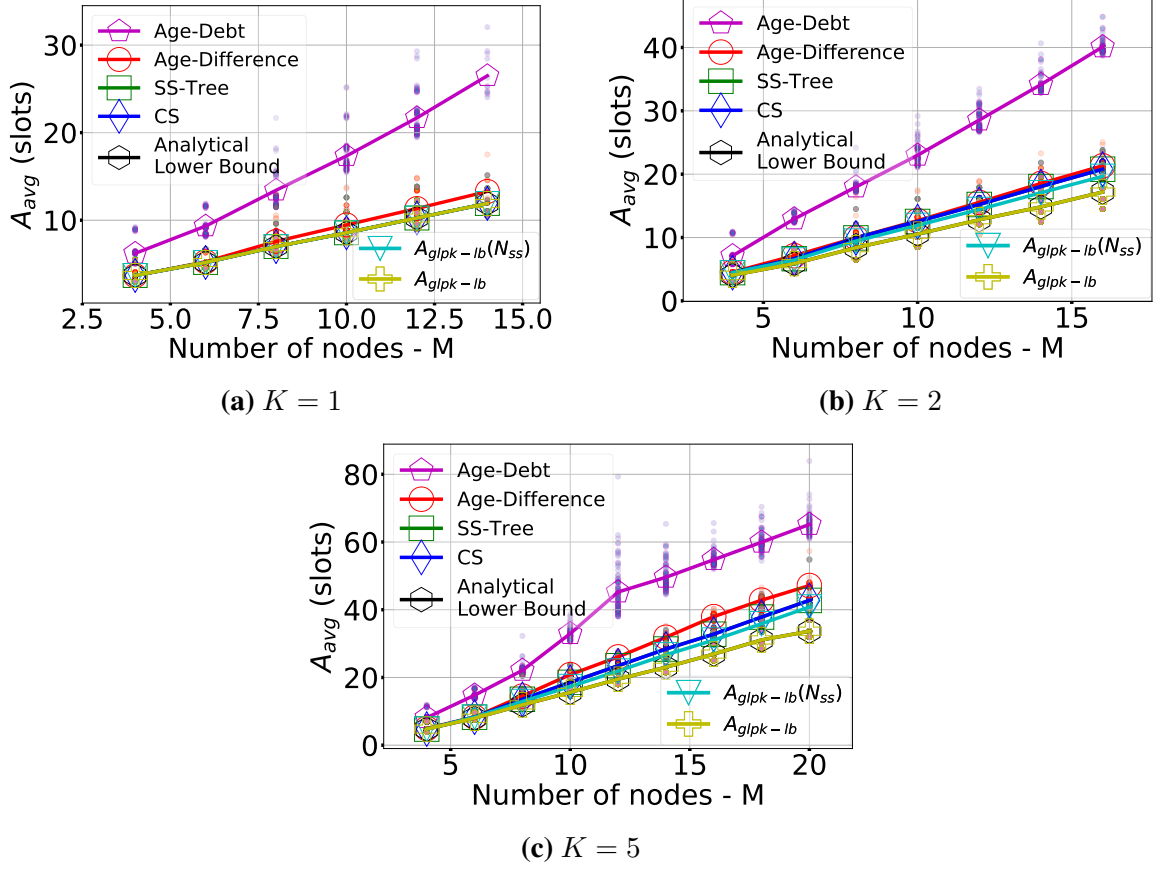


Figure 4.10: Simulated AAoI for multiline networks under SS-Tree, CS, Age-Difference, and Age-Debt policies. Results are compared with analytical lower bounds $A_{glpk-lb}(N_{ss})$ and $A_{glpk-lb}$ for varying M : (a) $K = 1$, (b) $K = 2$, and (c) $K = 5$.

policies, SS-Tree and CS, yield a larger value of A_{avg} compared to $A_{glpk-lb}$. This gap is as high as 27.8% for $K = 5$. The A_{avg} achieved by the Age-Difference policy is further higher when compared to SS-Tree or CS. An interesting observation is that the AAoI achieved by SS-Tree and CS closely matches $A_{glpk}(N_{ss})$, where N_{ss} denotes the period obtained for both policies. This suggests that these policies are near-optimal for the corresponding period. However, the period satisfies $N_{ss} > N_{lb}$. For larger values of M , as shown in Figures 4.11(a)-(c), we again observe that SS-Tree and CS achieve smaller or comparable values of A_{avg} relative to the Age-Difference policy, which is non-periodic and uses the same set of activation vectors as SS-Tree and CS. For $M = 200$, the percentage increase in AAoI of the Age-Difference policy relative to SS-Tree (or CS) is 49.8%, 15.9%, and 13.3% for $K = 1$, $K = 2$, and $K = 5$, respectively, as shown in Figures 4.11(a)-(c). The lower bound uses the analytical lower bound on N . However, we conjecture that the lower bound on N is weak, since if we use the N_{ss} obtained from simulating the SS policy in the lower

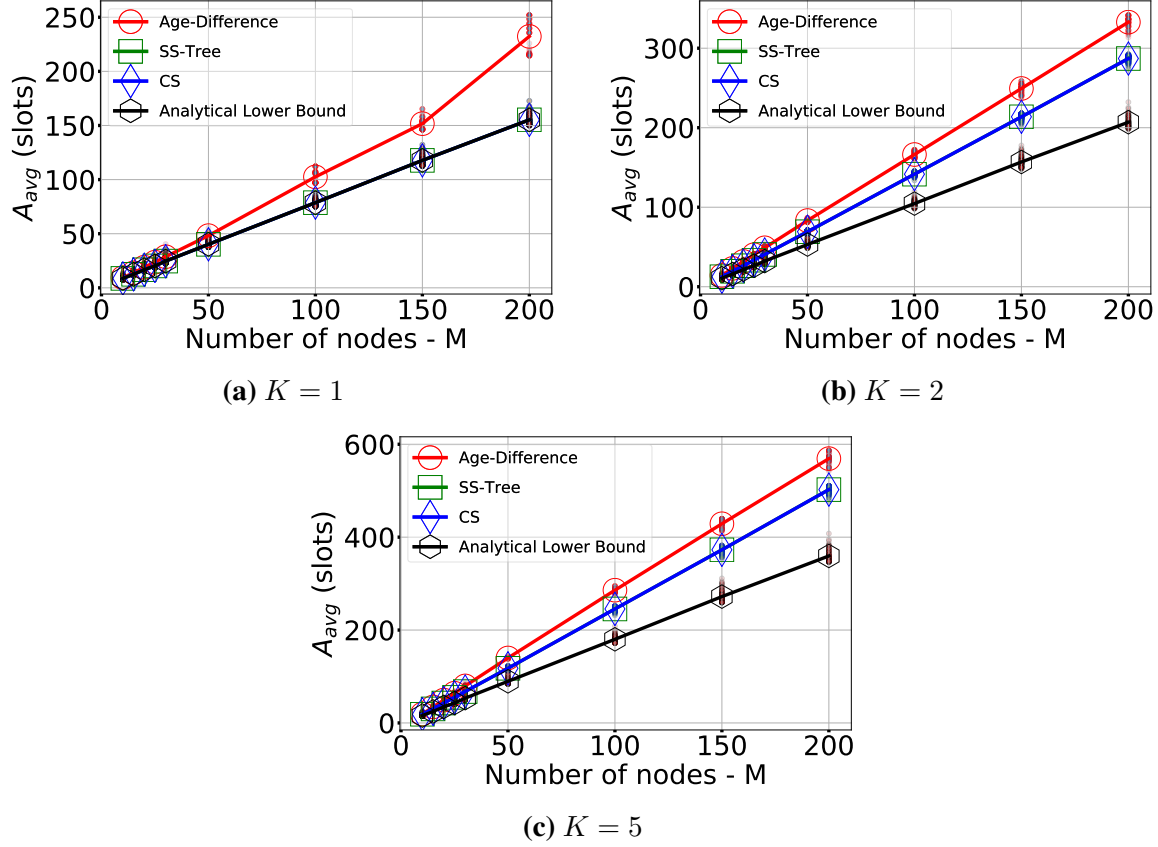


Figure 4.11: Simulated AAoI for multiline networks under SS-Tree, CS, Age-Difference, and Age-Debt policies for larger values of M : (a) $K = 1$, (b) $K = 2$, and (c) $K = 5$. Results are compared with analytical lower bounds.

bound, the gap vanishes. However, we do not have a better lower bound for N , which is a possible direction for future work.

In Figure 4.12, we compare the AAoI achieved by the different policies along with the corresponding lower bounds for tree network topologies. For each value of M , we generate an ensemble of 100 randomly selected tree topologies. Each tree is constructed by uniformly sampling from the set of all tree topologies with M nodes using a Prüfer sequence-based method [93]. The scatter plots and the average values of A_{avg} across the network ensemble are obtained in the same manner as discussed earlier and are illustrated in Figure 4.12. In Figure 4.12(a), smaller values of M are considered. As before, we observe that the analytical lower bound closely matches $A_{\text{glpk-lb}}$. The SS-Tree and CS policies also achieve values of A_{avg} that are close to $A_{\text{glpk-lb}}(N_{ss})$. However, for $M = 30$, SS-Tree and CS exhibit a higher A_{avg} compared to $A_{\text{glpk-lb}}$, with a percentage difference of 24.6%. The AAoI performance for larger values of M is shown in Figure 4.12(b). From

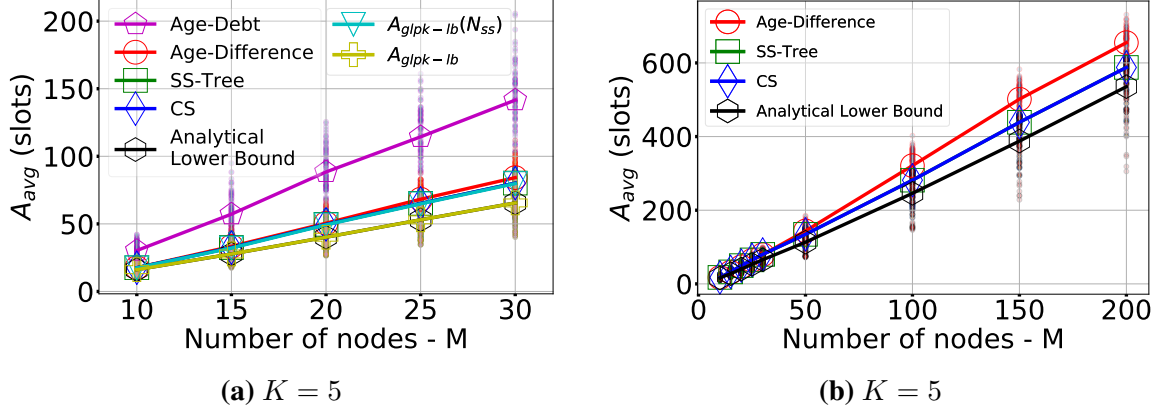


Figure 4.12: Simulated AAoI for tree networks under SS-Tree, CS, Age-Difference, and Age-Debt policies. Results are compared with analytical lower bounds $A_{glpk-lb}(N_{ss})$ and $A_{glpk-lb}$ for varying M with $K = 5$: (a) full comparison; (b) results for larger M under SS-Tree, CS, and Age-Difference policies.

this figure, we note that the percentage gap between SS-Tree (or CS) and the analytical lower bound decreases to 10.9% when M increases to 200. The Age-Difference policy again yields a larger A_{avg} compared to SS-Tree and CS. The percentage increase of Age-Difference relative to SS-Tree (or CS) is 5.4% at $M = 30$ (Figure 4.12(a)) and increases to 11.6% at $M = 200$ (Figure 4.12(b)). Again, the lower bound uses an analytical lower bound on N . Improving the lower bound on N is relegated to future work.

4.6 Extensions

4.6.1 Extension: Networks with Packet Losses

In this section, we present heuristic extensions of our policies to the case of networks with packet losses. The network model that we consider here is similar to that in Section 4.2 except for packet losses. We assume that each packet transmission on a link fails with a probability p . This failure event is independent and identically distributed (IID) across slots as well as across links¹². We assume that successful transmissions are acknowledged at the link level, so that when a transmission $(s, (i, j))$ fails, i is aware of the failure. We consider the following modification to SS-Tree and CS policies to operate in networks with packet losses. First, we obtain the periodic deterministic schedule (with period N_{ss}) for SS-Tree and CS from Algorithm 4 under the assumption that the network does not have any packet

¹²The discussion can be extended to the case of links with different failure probabilities.

losses. We operate the network under this schedule. Suppose the transmission $(s, (i, j)) \in U(t_f)$ fails in the t_f^{th} slot in a period. The s^{th} source's packet is discarded and in that period any subsequent link transmission of the form $(s, (i, j)) \in U(t)$, where $t_f < t \leq N$ and for any $(i, j) \in \mathcal{E}$ is not done. For SS-Tree and CS policies, the packet would be generated again in the next period and delivery to node 0 is attempted. For comparison, we assume that for Age-Difference and Age-Debt policies, the packet is discarded if there is a packet loss, and subsequent packet generation and forwarding is done in accordance with the age and virtual queue evolution respectively. We note that for Age-Difference and Age-Debt an alternative would be to buffer the s^{th} source's packet at i if the transmission $(s, (i, j))$ fails and then re-try transmission. We denote this alternative as Age-Difference-Retry (or Age-Debt-Retry), while the former policies are denoted with a suffix *-Discard*.

Lower Bounds on A_{avg} for Periodic Deterministic Policies: We note that the lower bounds derived in Proposition 4.6 can be extended to any periodic deterministic packet-discard policy. The extension of these lower bounds is based on the following characterization of the A_{avg} . Consider a source s that is at a hop distance of h_s from 0. We note that the probability of successfully transmitting a packet from s to 0 is $p_s = (1 - p)^{h_s}$ since all link transmissions from s to 0 should be successful. Let $p_{l,s} = 1 - p_s$. We then have the following characterization of the A_{avg} .

Proposition 4.10. *The average AAoI for any periodic deterministic policy with period N employing packet discards is $A_{avg} = \frac{1}{M} \sum_{s=1}^M \left(d_s + \frac{(N-1)(2-p_s)}{2p_s} \right)$, where d_s is the delay in delivering a packet from s to 0 under the policy.*

Proof. We characterize A_{avg} using Renewal Reward Theorem (RRT) [92] by identifying a renewal process in the evolution of $A_0^s(t)$. The renewal instants are the slots just after successful delivery of a packet; $A_0^s(t)$ at the start of a renewal cycle is d_s . The renewal cycle is the random number of periods taken for the successful delivery of a packet from s . The number of such periods is Geometric with success probability p_s and is IID; we use R to denote the number of periods in a renewal cycle. The number of slots in a renewal cycle is NR . The cumulative age in a renewal cycle is $d_s NR + NR(NR - 1)/2$ since $A_0^s(t)$ is linearly increasing in a renewal cycle. Using RRT A_{avg} is the ratio of the expected cumulative age to the expected number of slots in a renewal cycle; this ratio is $\left(d_s + \frac{(N-1)(2-p_s)}{2p_s} \right)$. The average over all sources gives us the above result. $\square$

From Proposition 4.6 we have that each $d_s \geq h_s$. Also, for any periodic deterministic policy we have that $N \geq N_{lb}$. Thus, we have the following corollary.

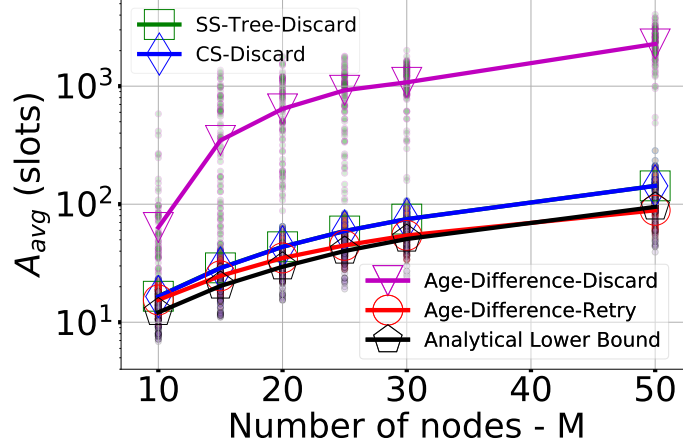


Figure 4.13: Comparison of A_{avg} for SS-Tree-Discard, CS-Discard, Age-Difference-Discard, and Age-Difference-Retry policies, along with an analytical lower bound. Results are shown for tree networks with packet losses ($p = 0.05$) under 2-hop interference.

Corollary 4.1. *The A_{avg} for any periodic deterministic policy employing packet discards is bounded from below as $A_{avg} \geq \frac{1}{M} \sum_{s=1}^M \left(h_s + \frac{(N_{lb}-1)}{2} \frac{(1+p_l)}{(1-p_l)} \right)$.*

Simulation Results: In Figure 4.13, we compare the AAoI for SS-Tree-Discard, CS-Discard, Age-Difference-Discard, Age-Difference-Retry, as well as the lower bound from Corollary 4.1. Tree networks with $p = 0.05$ and 2-hop interference are considered. For every M , we consider 100 randomly chosen tree topologies (generated using Prufer sequences as described in Section 4.5). Age-Debt-Discard as well as Age-Debt-Retry are excluded as their A_{avg} are much worse compared to the other policies. We also note that in simulation of A_{avg} for Age-Difference-Discard, some nodes do not successfully transmit packets to the sink node. For example, the average percentage of nodes without packet deliveries to 0 are 0.04%, 0.07%, and 2.08% in 100 graphs for $M = \{25, 30, 50\}$ respectively. These nodes are excluded from the A_{avg} calculation. We observe that SS-Tree-Discard as well as CS-Discard perform better than Age-Difference-Discard, while Age-Difference-Retry (which is non-periodic) has the best performance¹³

4.6.2 General Functions of AoI

In this section, we extend the discussion to the case where the performance metrics are defined as general functions of the AoI.

¹³The lower bounds are for periodic policies while Age-Difference policies are non-periodic and can have a AAoI smaller than the lower bounds.

4.6.2.1 General Function of AoI for a Source

In many applications [11], the performance metric of interest is a nonlinear, non-decreasing function $f(A_0^s(t))$ of the AoI corresponding to source s .

For a periodic deterministic policy with period N satisfying PeDe, we denote the time average of $f(A_0^s(t))$ for source s by $\text{AFAoI}(s)$. Since the evolution of AoI is periodic, as illustrated in Figure 4.2, the AoI values assumed by $A_0^s(t)$ over a period are $d_s, d_s + 1, \dots, d_s + N - 1$. Therefore, for source s , we have

$$\text{AFAoI}(s) = \frac{1}{N} \sum_{i=0}^{N-1} f(d_s + i). \quad (4.11)$$

As a system level performance metric, we define the average across all sources as $\text{AFAoI} = \frac{1}{M} \sum_{s=1}^M \text{AFAoI}(s)$.

Since $f(\cdot)$ is non-decreasing, $f(d_s + i)$ is also a non-decreasing function of d_s . Furthermore, for a fixed value of d_s , it can be shown that $\text{AFAoI}(s)$ is non-decreasing in the period N . Therefore, a possible approach for minimizing AFAoI is to design a policy that jointly minimizes both d_s and N . As from the discussions regarding P(4.3) and P(4.3)-Improvement in Section 4.4, the SS-Tree and CS policies are designed to heuristically minimize d_s for all s , as well as the period N . Hence, we propose using SS-Tree and CS without modification for any general non-decreasing function $f(\cdot)$. To evaluate the performance of SS-Tree and CS, we next derive numerical and analytical lower bounds on AFAoI.

Proposition 4.11. *Let $\text{AFAoI}_{lb}(s) = \frac{1}{N_{lb}} \sum_{i=0}^{N_{lb}-1} f(h_s + i)$, where h_s is the hop-distance of source s and N_{lb} has been obtained from Proposition 4.6.*

We define $\text{AFAoI}_{lb} = \frac{1}{M} \sum_{m=1}^M \text{AFAoI}_{lb}(s)$. Consider a tree network with K -hop interference and a non-decreasing function $f(\cdot)$ of the AoI. For any policy satisfying PeDe we have that $\text{AFAoI} \geq \text{AFAoI}_{lb}$.

Proof. Consider $\text{AFAoI}(s)$ for any non-decreasing $f(\cdot)$. Then, $f(d_s + i) \geq f(h_s + i)$ where h_s is the hop distance of source s . For any N , it can be shown that $\frac{1}{N} \sum_{i=0}^{N-1} f(h_s + i) \leq \frac{1}{N+1} \sum_{i=0}^N f(h_s + i)$. Any periodic deterministic policy that satisfies PeDe has $N \geq N_{lb}$ (from Proposition 4.6). Therefore, a lower bound on $\text{AFAoI}(s)$ is $\text{AFAoI}_{lb}(s) = \frac{1}{N_{lb}} \sum_{i=0}^{N_{lb}-1} f(h_s + i)$. We define $\text{AFAoI}_{lb} = \frac{1}{M} \sum_{s=1}^M \text{AFAoI}_{lb}(s)$, so that $\text{AFAoI} \geq \text{AFAoI}_{lb}$. $\square$

For functions $f(\cdot)$ which are also convex, we have the following numerical as well as

analytical lower bound.

Proposition 4.12. *For a tree network with K -hop interference and a non-decreasing convex function $f(\cdot)$, we define $AFAoI_{c-lb}(N)$ as the optimal value of the following problem.*

$$\begin{aligned} \min_{(d_s, g_{s,n}, a_{l,n}^s, \forall s \in [M], n \in [N], l \in \mathcal{E})} \quad & AFAoI, \\ \text{subject to} \quad & (4.2), (4.3), (4.4), (4.5), (4.6), (4.7). \end{aligned} \quad (4.12)$$

Here we consider $d_s, g_{s,n}, a_{l,n}^s \forall s \in [M], n \in [N], l \in \mathcal{E}$ as non-negative real numbers. Let $AFAoI_{c-lb} = \min_N AFAoI_{c-lb}(N)$ for feasible N . For any policy satisfying PeDe we have that $AFAoI \geq AFAoI_{c-lb}$.

Proof. The objective is AFAoI which is a function of $d_s, \forall s$ for a fixed N . Suppose $f(\cdot)$ is a convex function of AoI. We then have that $f(d_s + i)$ is a convex function of d_s for a fixed i . Since $AFAoI(s)$ for a fixed N is a sum of convex functions of d_s (see (4.11)), it is also a convex function. The objective AFAoI is the average of $AFAoI(s)$ and hence is convex. The relationships between $d_s, g_{s,n}$, and $a_{l,n}^s$ are captured in the linear constraint (4.7). All other constraints are also linear. By relaxing the integer constraints on $g_{s,n}, a_{l,n}^s$ and d_s we obtain a convex optimization problem that can be solved numerically using standard solvers such as Gurobi [121]. The optimal value of this problem which is a function of N is denoted as $AFAoI_{c-lb}(N)$. We define $AFAoI_{c-lb} = \min_N AFAoI_{c-lb}(N)$, which is the lower bound for convex $f(\cdot)$. $\square$

As in Proposition 4.3, we can show that $AFAoI_{c-lb}$ is $\min_{N \in \{N_{min}, \dots, N_{max}\}} AFAoI_{c-lb}(N)$ where $N_{min} = N_{lb}$ and $N_{max} = \sum_{s \in [M]} h_s$.

Simulations and Performance Comparison: In this section, we compare the AFAoI achieved by the SS-Tree and CS policies with the corresponding lower bounds. We consider two representative choices for $f(\cdot)$: (a) $f(a) = a^2$, and (b) $f(a) = \mathbb{I}\{a > 10\}$. The simulations and performance comparisons are carried out for different network topologies, values of M ¹⁴, and different values of K . In Figure 4.14, we consider the case $f(a) = a^2$ and compare the AFAoI of SS-Tree and CS with the numerical lower bound from Proposition 4.12 and the analytical lower bound from Proposition 4.11. For line networks, the AFAoI achieved by SS-Tree and CS matches both the numerical lower bound $AFAoI_{c-lb}$

¹⁴Smaller values of M are considered primarily due to the computational complexity involved in solving the convex optimization problem. However, the proposed algorithms as well as the analytical lower bounds can be extended to larger values of M , as demonstrated in Section 4.5.

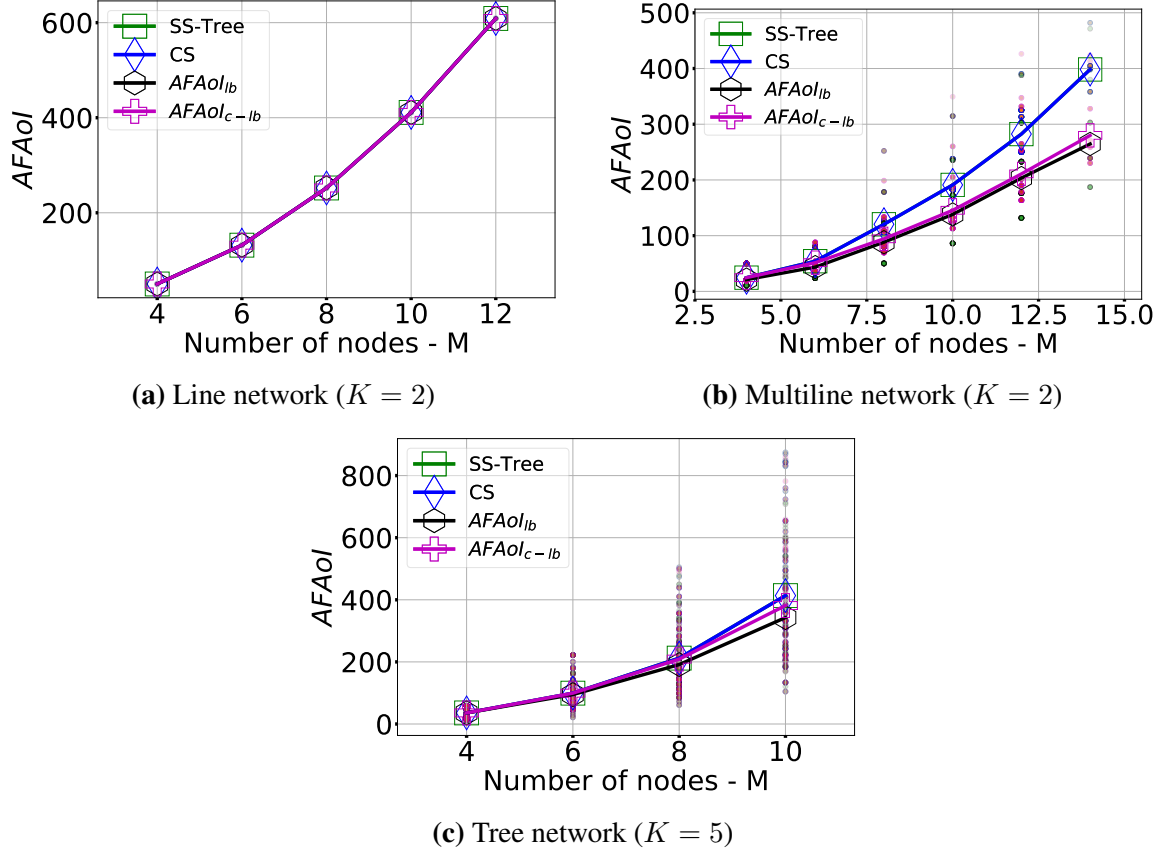


Figure 4.14: Simulated AFAoI for line, multiline, and tree networks under SS-Tree and CS policies. Results are compared with the general lower bound $AFAoI_{lb}$ and the Gurobi-based bound $AFAoI_{c-lb}$ for varying M and K , using $f(a) = a^2$: (a) line ($K = 2$), (b) multiline ($K = 2$), and (c) tree ($K = 5$).

and the analytical lower bound $AFAoI_{lb}$. For multiline and tree network topologies, however, SS-Tree and CS exhibit larger AFAoI values compared to the corresponding lower bounds. Further improvement in the performance of SS-Tree and CS for these topologies remains an interesting direction for future work.

In Figure 4.15, we consider the case $f(a) = \mathbb{I}\{a > 10\}$, which is non-convex and corresponds to the age violation probability [122]. We compare the AFAoI achieved by SS-Tree and CS with the numerical lower bound given in Proposition 4.11. For line, multiline, and tree network topologies, the AFAoI values obtained using SS-Tree and CS closely match the analytical lower bound, with percentage differences of 0%, 0.05%, and 0.02%, respectively, as M increases to 200. We note that the optimization of packet generation rates for minimizing the age violation probability has been studied in [122]. In our setting, however, the packet generation rate is determined by the period N .

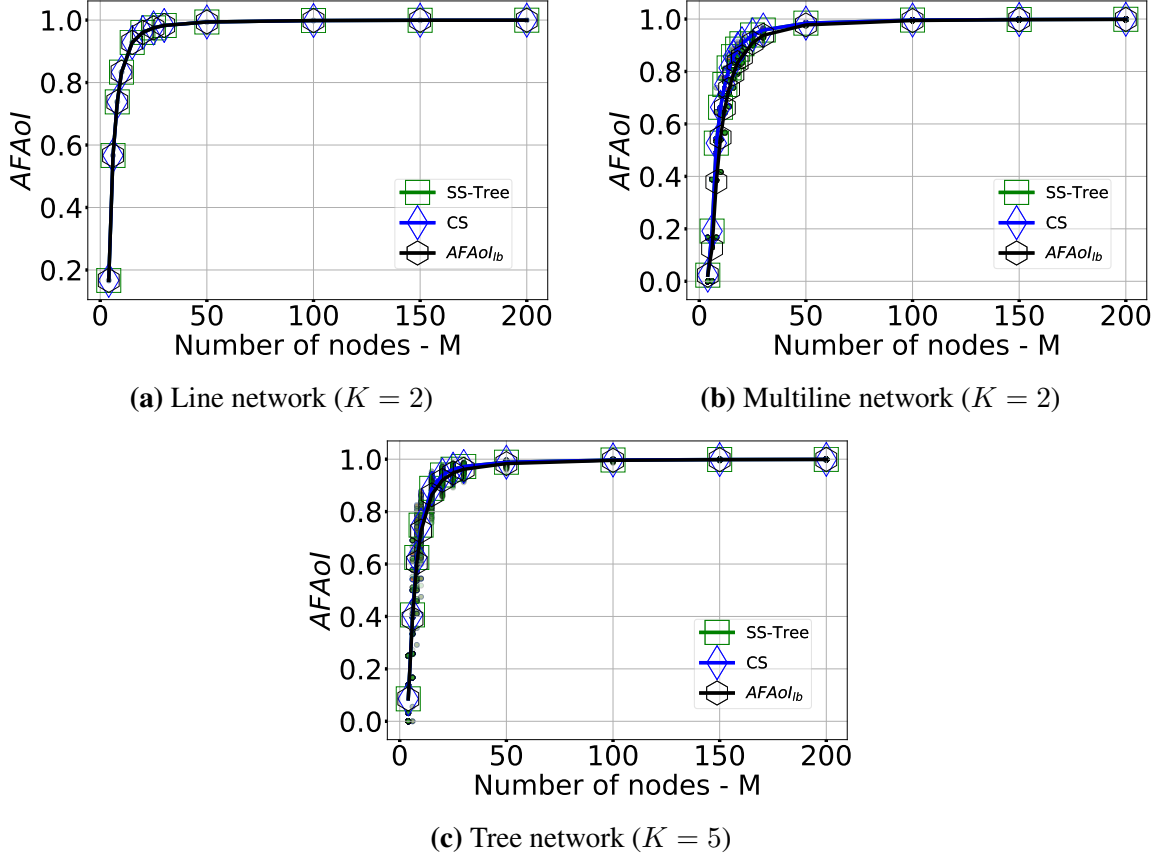


Figure 4.15: Simulated AFAoI for line, multiline, and tree networks under SS-Tree and CS policies. Results are compared with the analytical lower bound $AFAoI_{lb}$ for varying M and K , using $f(a) = \mathbb{I}\{a > 10\}$: (a) line ($K = 2$), (b) multiline ($K = 2$), and (c) tree ($K = 5$).

4.6.2.2 AoI Metrics for the System

So far, we have considered the average of $AAoI(s)$ or $AFAoI(s)$ as the performance metric for the system. We now extend the discussion to a weighted $AAoI$ metric¹⁵, defined as $WAAoI = \frac{1}{M} \sum_{s=1}^M w_s AAoI(s)$, where $w_s \geq 0$ are weights assigned to every source in the network $s \in [M]$, these weights can be utilized to prioritize the packets from different sources. For periodic deterministic policies satisfying PeDe, the lower bounds derived earlier for $AAoI$ can be naturally extended to $WAAoI$. In particular, the numerical lower bound obtained from the linear program, $A_{glpk-lb}(N)$, extends directly since the corresponding objective function becomes $\frac{1}{M} \sum_{s=1}^M w_s (d_s + (N - 1)/2)$ which is still linear. Therefore, the lower bound $A_{glpk-lb}$ can also be extended to the case where the

¹⁵This discussion can also be extended to weighted AFAoI metrics.

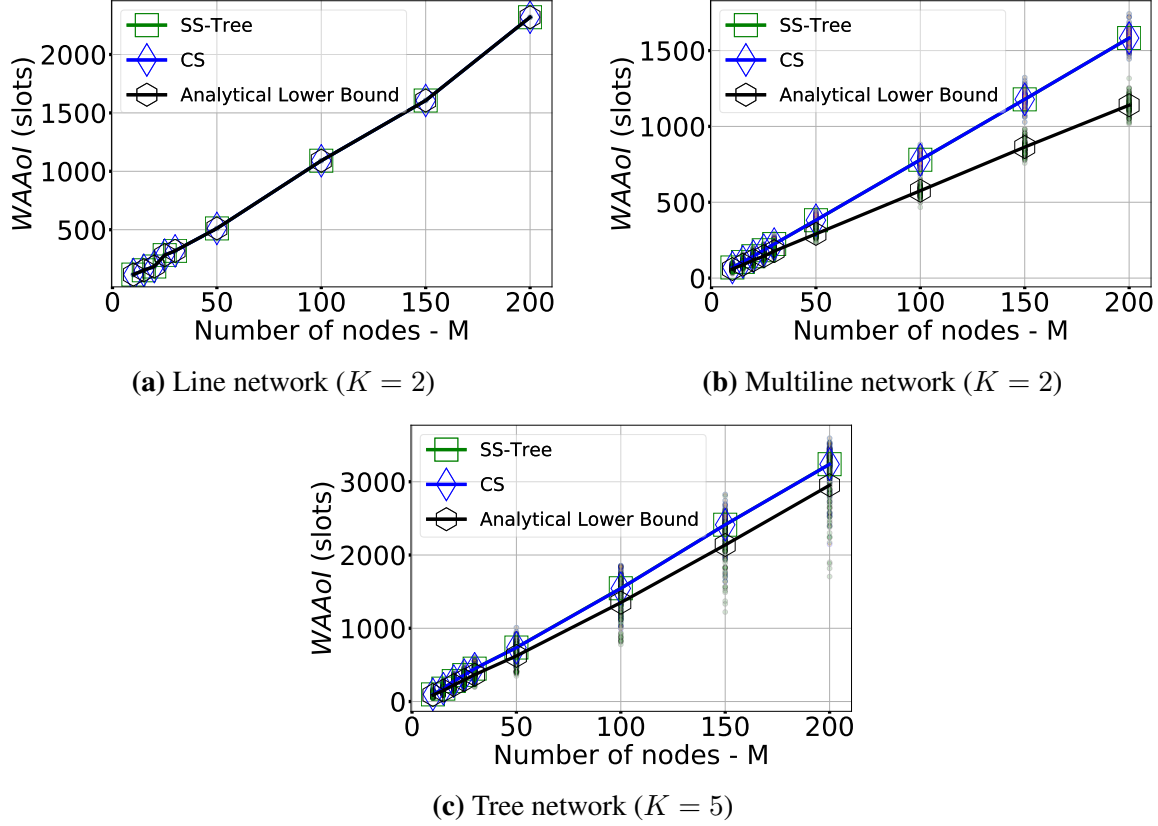


Figure 4.16: Simulated WAAoI for line, multiline, and tree networks under SS-Tree and CS policies. Results are compared with the analytical lower bound for varying M and K , with source weights randomly selected from $[1, 10]$: (a) line ($K = 2$), (b) multiline ($K = 2$), and (c) tree ($K = 5$).

metric is WAAoI. Similarly, the analytical lower bound from Proposition 4.7 extends to $\text{WAAoI} \geq \frac{1}{M} \sum_{s=1}^M w_s \left(h_s + \frac{N_{lb}-1}{2} \right)$, where h_s and N_{lb} are as before. We propose to use SS-Tree and CS algorithms without any modification, since for every source the AoI metric is an increasing function of the AoI (i.e., $w_s A_0^s(t)$).

Simulations and Performance Comparison: In Figure 4.16, we compare the WAAoI achieved by the SS-Tree and CS policies with the analytical lower bound $\frac{1}{M} \sum_{s=1}^M w_s \left(h_s + \frac{N_{lb}-1}{2} \right)$. The weights w_s corresponding to each source $s \in M$ are randomly selected from the range $[1, 10]$. We observe that, for line networks under 2-hop interference, the WAAoI achieved by the SS-Tree and CS policies matches the analytical lower bound. However, for multiline and tree network topologies, the percentage gap between SS-Tree (or CS) and the analytical lower bound increases with M .

4.6.2.3 Other Metrics for Sources and System

Another widely used information freshness metric is the Peak Age of Information (PAoI) [44]. For periodic deterministic policies, the PAoI corresponding to source s is given by $\text{PAoI}(s) = d_s + (N - 1)/2$, where N denotes the policy period. The average or weighted PAoI across all sources can be defined as $\frac{1}{M} \sum_s w_s \text{PAoI}(s)$. It is evident that the previous discussions on AAoI and WAAoI extend directly to this case, since the corresponding expressions are same. As discussed under the basic principle P(4.3), the SS-Tree and CS policies heuristically aim to jointly minimize d_s for all s and the period N . For periodic policies, this also implies that they seek to minimize the peak AoI, i.e., $\max_t A_0^s(t), \forall s$.

Another relevant system-level performance metric is the Age of Stalest Information (AoSI) [123]. For the system considered here, the time-average AoSI is

$$\lim_{T \rightarrow \infty} \left[\frac{1}{T} \sum_{t=1}^T \max_s A_0^s(t) \right]$$

. We note that $\frac{1}{T} \sum_{t=1}^T \max_s A_0^s(t) \geq \max_s \frac{1}{T} \sum_{t=1}^T A_0^s(t)$. Hence, the time-average AoSI is lower bounded by $\max_s \text{AAoI}(s)$. Since the proposed policies (through P(4.3)) heuristically aim to jointly minimize d_s for all s and the period N , they also minimize $\max_s \text{AAoI}(s)$, and consequently provide a lower bound on the AoSI. Furthermore, as the proposed periodic policies also heuristically reduce the peak AoI for each source s , evaluating their performance for metrics such as AoSI is relevant and will be considered in future work.

For applications such as remote estimation, performance metrics like mean-squared error (MSE) have been shown to be related to the AAoI. The relationship between MSE and AAoI has been analyzed for Wiener processes in [1] and for Ornstein-Uhlenbeck processes in [73, 124]. Since proposed periodic policies minimize the AAoI, they also have utility in such applications.

4.7 Conclusions

Motivated by practical industrial network protocols, we studied the design of periodic deterministic policies for minimizing the AAoI in tree networks under K -hop interference constraints. The problem of obtaining policies that achieve minimum AAoI was formulated as an integer programming problem. A linear programming relaxation of this problem was employed to obtain numerical lower bounds on the minimum AAoI. Furthermore, analytical lower bounds on the minimum AAoI were derived and utilized in the performance

evaluation. For line networks, we proposed the SS-Line policy, inspired by Gandham et al. [115], which leverages the concept of node states along with a special state-transition scheme to generate a schedule of link transmissions that satisfies the K -hop interference constraints. SS-Line was subsequently extended to two heuristic policies for tree networks, namely SS-Tree and CS. The AAoI performance of these proposed policies was evaluated through extensive simulations conducted across a variety of network topologies, including line, multiline, and tree networks. In all cases, the proposed analytical lower bounds were found to match the numerical lower bound $A_{glpk-lb}$. For line networks, across all considered values of K , the proposed periodic policies achieve an AAoI that matches both the analytical lower bounds and the numerical lower bound $A_{glpk-lb}$. For multiline and tree network topologies, for all considered values of K and sufficiently large M , the periodic policies achieve an AAoI that matches with or is smaller than the Age-Difference and Age-Debt policies. Additionally, we observe that the proposed policies attain an AAoI that closely aligns with the lower bound $A_{glpk-lb}(N_{ss})$, where N_{ss} is the period realized by these policies. We then extended the discussion to scheduling for general functions of AAoI. Analytical lower bounds, as well as a numerical lower bound based on a convex optimization formulation, were presented.

Appendix

4.A Computational Difficulties Associated with ILP

The complexity of solving Integer Linear Programs (ILP) can vary depending on the algorithm used. Lenstra's algorithm [125] solves ILP with a run-time complexity of $2^{O(n^3)} \cdot (m \cdot \log V)^{O(1)}$ where n represents the number of variables, m denotes the number of constraints, V is the largest coefficient in the ILP formulation. The term $2^{O(n^3)}$ captures the exponential dependence on the number of variables. The polynomial factor $(m \cdot \log V)^{O(1)}$ reflects the impact of the number of constraints and the magnitude of the coefficients on the overall run-time. So the complexity is exponential in the number of variables.

For our problem (4.8) which needs to be solved for a fixed period N , we formulate an optimization problem with several variables and constraints. The total number of variables and constraints are as follows.

I) Total number of variables:

- For $g_{s,n}$, s ranges from 1 to M , and n ranges from 1 to N , resulting in a total of $M \times N$ variables.
- For $a_{l,n}^s$, s ranges from 1 to M , n ranges from 1 to N , and l ranges from 1 to L , resulting in a total of $M \times N \times L$ variables.

Thus, the total number of variables is $M \times N + M \times N \times L$ (which is n in the above discussion).

II) Total number of constraints:

- Constraint (4.2): $\sum_{n=1}^N g_{s,n} = 1, \forall s \in [M]$. This gives a total of M constraints.
- Constraint (4.3): $\sum_{l \in E^-(0)} \sum_{n=1}^N a_{l,n}^s = 1, \forall s \in [M]$. This also results in M constraints.
- Constraint (4.4): $\sum_{s=1}^M \left(\sum_{l \in E^+(v)} a_{l,n}^s + \sum_{l \in E^-(v)} a_{l,n}^s \right) \leq 1$. This constraint gives a total of $N \times (M + 1)$ constraints.
- Constraint (4.5): $\sum_{s=1}^M a_{l,n}^s + \sum_{s=1}^M a_{l',n}^s \leq 1$. The total number of these constraints is $L \times |N_l| \times N$.
- Constraint (4.6): $\sum_{l \in E^+(i)} a_{l,n}^s \leq \sum_{l \in E^-(i)} \sum_{n' \leq n-1} a_{l,n'}^s + \delta(i-s) \sum_{n' \leq n} g_{s,n'} - \sum_{l \in E^+(i)} \sum_{n' \leq n-1} a_{l,n'}^s$. There is one such constraint for each i , each s , and each n , resulting in $M \times M \times N$ constraints.

Therefore, the total number of constraints is $M + M + N \times (M + 1) + L \times |N_l| \times N + M \times M \times N$. We note that this is m in the above discussion.

4.B Proof of Proposition 4.6

Proof. We consider relaxation of the constraints in (4.8). We remove the constraint (4.6) since our lower bound is based on counting the number of slots required to collect packets at 0 without any consideration to the ordering. We assume that (4.2) and (4.3) hold. However, (4.4) and (4.5) are assumed to hold only for links of the form $(i, 0)$, where i is a top-subtree root. The proof then proceeds similarly as in the case of the multiline network. We first obtain the exclusive slots for every subtree. Consider a node i in a subtree b which is at a hop distance h , where $h \in [\lceil K/2 \rceil]$. If i transmits a packet (either its own or one of its descendants), then due to K -hop interference, there cannot be any transmissions from any node j in any top-subtree, where j 's hop distance is within $\lceil K/2 \rceil$. We note that this corresponds to slots in which links of the form $(i, 0)$ are scheduled or constrained to be not scheduled (corresponding to variables $a_{(i,0),n}^s$ being 1 or 0). In a tree, let $n_b(h)$ denotes the number of nodes in top-subtree b which are within a hop-distance of h from 0, with $n_b^h(0) = 0$. Since the total number of sources in b with hop distances at most h is $n_b(h)$, i has to transmit $n_b - n_b(h - 1)$ packets using the same number of slots. And these slots are *exclusive* since no other node j as discussed above can transmit. By considering all nodes i within a hop distance of $\lceil K/2 \rceil$ for b , we define the number of exclusive slots e_b for a top-subtree b as $e_b = \sum_{h=1}^{\lceil K/2 \rceil} (n_b - n_b(h - 1))$. The number of slots required to satisfy PeDe is therefore bounded below by $\sum_b e_b$.

We now consider the transmissions from nodes which are at a hop distance more than $\lceil K/2 \rceil$. Of these transmissions in a top-subtree b , we consider those from nodes i with $h \in \{\lceil K/2 \rceil + 1, \dots, K - 1\}$. Such transmissions cannot be concurrent with the exclusive slots for b or amongst themselves due to K -hop interference constraint. However, some of these transmissions can be concurrent with the exclusive slots for any other top-subtree and will be considered as such in the following¹⁶. Transmissions from nodes i with $h > K - 1$ can be concurrent with the exclusive slots and are disregarded for the lower bound. For a top-subtree b , the total number of transmissions from nodes i with $h \in \{\lceil K/2 \rceil + 1, \dots, K - 1\}$ is bounded below by $f_b = \sum_{h=\lceil K/2 \rceil + 1}^{\min(l_b, K-1)} f_b(h)$, where we

¹⁶We note that only some of these transmissions can be concurrent, e.g., a node at a hop distance of $\lceil K/2 \rceil + 1$ cannot transmit concurrently with another top-subtree root. For a lower bound, we consider all transmissions to be concurrent.

consider just the nodes in the longest t-line of b with a length of l_b . Here $f_b(h)$ is the total number of packets to be forwarded from a node at a hop distance of h in the longest t-line; this is the total number of descendants of that node in addition to itself. Since these f_b transmissions can be concurrent with the exclusive slots of any other top-subtree, we obtain $(f_b - e_{b,o})^+$ which is the extra number of slots required for transmissions from b . Here $e_{b,o}$ is the sum of exclusive slots for all top-subtrees except b . Then, over all top-subtrees, the maximum number of extra slots required is $e_x = \max_{b \in \mathcal{R}_{top}} (f_b - e_{b,o})^+$. Again, this corresponds to additional slots and corresponding variables $a_{(i,0),n}^s$ required to satisfy the relaxed constraints. A lower bound on the period is therefore $e_x + \sum_{b \in \mathcal{R}_{top}} e_b$. $\square$

4.C Proof of Proposition 4.8

Proof. By construction (see line 6) we obtain a periodic deterministic schedule $\mathcal{U}_N$ with period $(K + 1)(M - 1) + 1$. As discussed in P1-(4.4), (4.5) with the initialization of the states (using m_i in line 3) and the node-state evolution from Figure 4.6, in every slot the transmitting links satisfy constraints (4.4)-(4.5). Consider the last node M in the line. Independent of the initial value of M .state, there will be one visit to the state T in the node evolution. So, M generates and transmits a packet in $K + 1$ slots. Now consider the node $M - 1$. If the initial state of this node is T , then it generates and transmits a packet in the first slot. Then, within a further $K + 1$ slots it would receive and forward the M^{th} node's packet (since initially M .state = I_1). If the initial state of the $M - 1^{th}$ node is not T , then it receives and forwards M^{th} nodes packet within $K + 1$ slots. Then, within a further $K + 1$ slots it would generate and transmit its own packet. Similarly, we can show that for the $(M - n)^{th}$ node, where $n \in [M - 1]$, the node receives and forwards packets from its descendants as well as generates and forwards its packet within $1 + n(K + 1)$ slots if the initial state is T , or $(n + 1)(K + 1)$ slots if the initial state is not T . For node 1, the initial state is T , so that the total number of slots required is $(M - 1)(K + 1) + 1$ to forward its own as well as the packets of all other nodes $2, \dots, M$. For any other node, the total number of slots required is at most $(M - 1)(K + 1)$ which is $\leq (M - 1)(K + 1) + 1$. Thus, the schedule generated by Algorithm 1 transmits one packet from every node within $(K + 1)(M - 1) + 1$ slots and satisfies PeDe (constraints (4.2)-(4.3). The algorithm also satisfies (4.6) as discussed in P1-(4.6). $\square$

4.D Proof of Proposition 4.9

Proof. We first obtain the time complexity of the initialization step for Algorithm 4. We note that the initialization of state as well as other node parameters require $\mathcal{O}(M)$ time. Since we visit each node in order to initialize its activation threshold, that also requires $\mathcal{O}(M)$ time. Initializing the compatible states set for CanAddBase function requires $\mathcal{O}(K^2)$ time. We now characterize the time complexity of a single iteration (see lines 9-40) of Algorithm 4. We describe only those steps requiring $\mathcal{O}(M)$ or higher in the following (steps such as line 9 or line 13 which require $\mathcal{O}(1)$ time are not discussed). Finding out the $\arg \max$ in line 11 requires $\mathcal{O}(M)$ time. Sorting of the top-subtrees in terms of $b.pb$ requires $\mathcal{O}(M \log M)$ in lines 13, or 21 and 26. In function CanAddBase we have to search a worst-case of $\mathcal{O}(K)$ elements (since the compatible set for some top-subtree root's state can have K elements) and this needs to be done for at most M top-subtree roots; so here the complexity is $\mathcal{O}(KM)$. Then, for each top-subtree we do ScheduleAndStateUpdate. We note that even though ScheduleAndStateupdate is done in a loop separately for each top-subtree and then again for every node in that top-subtree which is active, essentially we do $\mathcal{O}(1)$ operations for all nodes in the worst case. Thus, the lines 32-36 require $\mathcal{O}(M)$ time. Thus, the overall complexity is $\mathcal{O}(KM + M \log(M))$. Since in worst case, we have $(K + 1)M$ iterations, the overall complexity is $\mathcal{O}(K^2M^2 + KM^2 \log(M))$ including that for initialization. $\square$

Chapter 5

Periodic Deterministic Policies for Minimizing Average Age of Incorrect Information in Time-Slotted Multi-Hop Networks

5.1 Introduction and Prior Work

Modern industrial wireless networks need to support remote estimation of physical processes, which is essential for applications such as industrial process control, autonomous vehicles, and video surveillance. These applications rely on deterministic scheduling, as provided by industrial standards like WirelessHART [5], ISA100.11a [29], and IEEE 802.15.4e TSCH [4]. Wireless networks can be designed and optimized for such applications using information-freshness metrics, such as the age of information (AoI) [2] and the age of incorrect information (AoII) [3]. In this chapter, we study the remote estimation of source processes over a multihop wireless network with K -hop interference constraints, employing periodic deterministic sampling and scheduling policies. While prior work on multihop network scheduling has primarily focused on AoI, we consider AoII, which captures both the freshness of information as well as correctness of the remote estimate and the source process.

We derive an analytical characterization of the average AoII (AAoII) under periodic policies, expressed as a function of the policy's period and the packet delivery delays for

*This chapter is based on the conference paper by Nibin Raj, Vineeth B.S., "On Average Age of Incorrect Information in multihop wireless networks with interference for periodic sampling and scheduling policies," in Proceedings of National Conference on Communications (NCC), 2026.

each source. This characterization is further used to establish a novel analytical lower bound on the AAoII achievable by any periodic policy. Additionally, an upper bound on the AAoII is obtained using a periodic policy from our prior work (as discussed in Chapter 4). Together, these results provide fundamental performance limits for periodic policies and offer a quantitative understanding of how the scheduling period, delivery delays, and source process parameters impact information freshness.

5.1.1 Prior Work

The concept of AoI was first introduced in [10]. Since then, several studies [32, 11, 126, 127] have investigated AoI in single-hop networks. In particular, these works considered queueing models in which all undelivered packets, including stale or outdated updates, are retained in the buffer, and analyzed the resulting long-term average age. These studies demonstrated that the sampling rate that minimizes the average AoI (AAoI) generally does not coincide with the rates that optimize throughput or delay [32, 11, 126]. For a comprehensive review of the AoI literature, the reader is referred to [33]; an overview of earlier studies on single-source, single-destination settings is provided in [34]. Most of the studies addressed problem on AoI in multi-source, single-hop networks [38], [39], [40], [41]. There has been a limited number of prior work on scheduling for multihop networks with the objective to optimize information freshness metrics. The minimization of average AoI for a multihop network with general interference constraints was considered in [42] and interference-free multihop networks in [43]. Bounds on the AAoI for a ring topology with multiple sources and destinations were considered in [44]. Farazi et al. [47] studied scheduling strategies for minimizing AoI in multi-hop networks with multiple sources, where information is disseminated through all-to-all broadcast communications in a time-slotted setting. Periodic deterministic policies were considered in [44, 47] for a complete interference model and a multi-source, multi-sink scenario.

Another information-freshness metric AoII, was introduced in [3]. The suitability of AoII as a goal-oriented performance metric was further explored in [99]. The design of a decentralized policy for minimizing the average AoII in a slotted ALOHA network was investigated in [57]. The work in [58] investigated the AoII performance of a slotted ALOHA network without feedback, where each source was modeled as a two-state Markov chain. In another work, the authors proposed a heuristic scheduling policy for minimizing AoII in a setting with multiple sources transmitting status updates to a central node [102]. In contrast to existing work on AAoII in single-hop networks, we consider the problem of

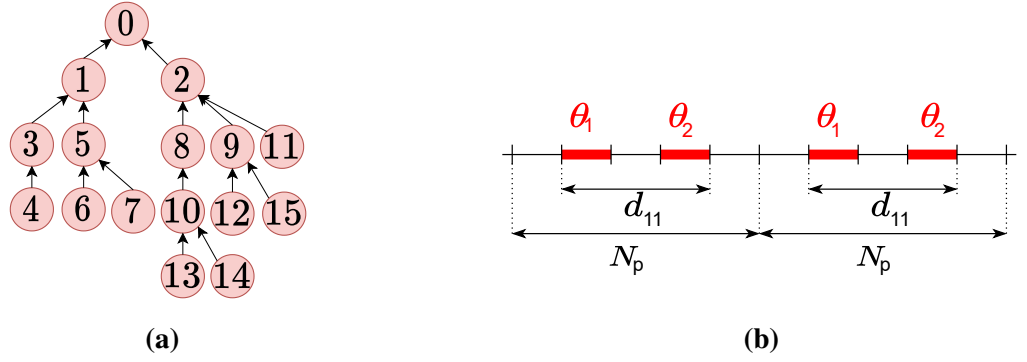


Figure 5.1: (a) Multihop tree network with $M = 15$ sources. (b) Link activation schedule for source 11, where $\theta_1 = (11, (11, 2))$ and $\theta_2 = (11, (2, 0))$. The delay d_{11} denotes the time for packets from source 11 to reach node 0, and N_p is the scheduling period. The schedule repeats every N_p .

AAoII minimization in a multihop network setting.

Prior work on multihop scheduling for optimizing AoI, as well as general functions of AoI, is available in [44, 9]. The design of non-periodic scheduling algorithms for optimizing AAoII was considered in Chapter 3. In contrast, this chapter focuses on deterministic periodic scheduling policies, which are commonly employed in practical communication protocols. Periodic scheduling for AoI was studied in Chapter 4, and the present chapter can be viewed as an extension of Chapter 4 to the AoII metric. As summarized above, we provide a novel characterization of AoII under a periodic policy and employ a lower bound on the scheduler period, together with the policy proposed in Chapter 4, to obtain an upper bound.

5.1.2 Contributions

1. In this chapter, we derive a novel analytical characterization of the AAoII for random piecewise-constant source processes under periodic multihop sampling and scheduling policies (see Proposition 5.3). This characterization is expressed in terms of the policy period, the packet delivery delays associated with each source, and the parameters of the underlying source processes.
2. Next, we develop a lower bound on the AAoII by leveraging lower bounds on the period of any periodic policy that generates and successfully delivers one packet from every source within each period (Section 5.3.1).

3. In addition, an upper bound on the AAoII is presented in Section 5.3.2 using a periodic policy adopted from Chapter 4.
4. Simulation results and related discussions are provided in Section 5.4. Through these simulations, we demonstrate the near-optimality of the periodic policy for networks operating under complete interference.
5. Furthermore, we compare the performance of a non-periodic AAoII-minimizing scheduling policy against the derived bounds and show that, under both complete-interference and K -hop interference models, periodic policies achieve lesser AAoII.

Notations: The notations used in this chapter are given in Table 5.1.

Table 5.1: List of symbols and their descriptions used in Chapter 5

Symbol	Description
$[M]$	Set of integers $\{1, 2, \dots, M\}$.
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Directed tree representing the multi-hop network, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of directed edges.
$\mathcal{S}$	Set of source nodes (sensor nodes) that generate packets for the root node.
$ U $	Cardinality of the node set U .
$(k, l) \in \mathcal{E}$	Communication link in the network, representing a transmission from node k to node l .
$(s, (k, l))$	Link activation indicating that a packet from source s is transmitted from node k to node l in slot t .
$(s, (s, l))$	Special case where the packet is generated at the source node s and transmitted directly to node l .
Θ_t	Activation set at time slot t , where each element $\theta_k \in \Theta_t$ represents a link (k, l) transmitting a packet from a source s .
N_p	Period of the scheduler.
π_P	Periodic scheduling policy.
K	Parameter defining the K -hop interference model, with $K \in \{1, 2, \dots\}$.
$X^s(t)$	State of the observed process for source $s \in \mathcal{S}$ at time slot t .

p_c^s (or q_c^s)	Probability that the process $X^s(t)$ transitions (or does not transition) to a new state at the beginning of slot t , where $q_c^s = 1 - p_c^s$.
$f_X^s(x)$	Probability density function used to sample a new state for $X^s(t)$ when a transition occurs.
$\hat{X}^s(t)$	Remote estimate of the process state $X^s(t)$ at the root node 0 for source s .
$A^s(t)$	AoII of source s at the root node 0 at time slot t .
$V^s(t)$	Latest slot (including t) at which the remote estimate $\hat{X}^s(t)$ of source s was correct.
$\overline{A}^{\pi_P}$	Average AoII for the system under periodic policy π_P .
N_p	Slotframe length or period of the scheduling policy.
d_s	Delivery delay for source s , $\forall s \in \mathcal{S}$.
w_s	Waiting period for source s , i.e., the number of slots from the beginning of the slotframe until the packet generation instant: $w_s = N_p - d_s$.
n	Index of the slotframe, where $n \in \mathbb{Z}^+$.
A_n^s	Embedded AoII of source s at the beginning of the n^{th} slotframe.
C_n^s	Embedded AoII of source s at the end of Phase I within the n^{th} slotframe.
$\mathcal{R}_{\text{top}}$	Set of all top-subtree root nodes.
B	Number of top-subtree roots, i.e., $B = \mathcal{R}_{\text{top}} $.
$\mathcal{T}_b$	Subtree rooted at node $b \in \mathcal{R}_{\text{top}}$.
n_b	Number of nodes in subtree $\mathcal{T}_b$, i.e., $n_b = \mathcal{T}_b $, with $\sum_{b=1}^B n_b = M$.
N_{lb}	Analytical lower bound on the period N .
$\overline{A}^{\pi_P, \text{LB}}$	Lower bound on the average AoII under the periodic policy π_P .
$\overline{A}^{\pi_P, \text{UB}}$	Upper bound on the average AoII under the periodic policy π_P .

5.2 System Model and Problem Statement

We consider a directed tree topology represented by the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ denotes the set of all nodes in the network and $\mathcal{E}$ denotes the set of all links in the multihop network. An example of such a tree network is shown in Fig. 5.1(a). All other assumptions regarding the network are the same as those presented in Chapter 4.

In this chapter, we consider both a complete interference model, in which only a sin-

gle packet transmission is allowed in a given timeslot, and a K -hop interference model, where $K \in \mathbb{Z}_+$. Under the K -hop interference model, concurrent transmissions on two links separated by at least K hops do not interfere with each other and can therefore occur simultaneously. However, if the separation between the two links is less than K hops, interference occurs, preventing successful packet reception at the corresponding receivers.

We study multihop scheduling policies that determine the set of links scheduled for transmission in each timeslot. Furthermore, for every active link, the scheduler must also decide which source packet is to be forwarded. To formally represent the scheduling decision at slot t , we define an *activation set*, denoted by Θ_t , as the largest set of links that can transmit simultaneously while satisfying the underlying interference constraints. The activation set Θ_t is expressed as $\{\theta_1, \theta_2, \dots, \theta_{|\Theta_t|}\}$, where each element corresponds to a link (k, l) transmitting a packet associated with source s . Accordingly, each element $\theta_k \in \Theta_t$ is denoted by $(s, (k, l))$, which signifies that a packet corresponding to the s^{th} source is transmitted from node k to node l during slot t . When $k = s$, the packet is newly generated at source node s and transmitted to node l . Otherwise, the packet, previously received at node k , is forwarded from node k to node l . At every intermediate node, generated or received packets are stored in a buffer before being relayed to the corresponding parent node, and this forwarding process continues until the packet reaches the sink node 0. Further, whenever a newer packet from the same source arrives at an intermediate node, it is assumed to replace any older packet of that source currently stored in the buffer. The system evolves according to a multihop scheduling policy. A *scheduling policy* is defined as the sequence of activation sets $(\Theta_t, t \in \mathbb{Z}_+)$. In this chapter, we focus on centralized scheduling policies, where the schedules are generated based on complete knowledge of the network topology.

This chapter focuses on the design of deterministic periodic scheduling policies, motivated by practical communication protocols supported by industrial standards such as WirelessHART [5], ISA100.11a [29], and IEEE 802.15.4e TSCH [4], which employ periodic schedulers with a period equal to a *slotframe*. We assume that a periodic scheduler, denoted by π_P , satisfies the following properties:

1. The scheduler generates a deterministic schedule $(\Theta(0), \Theta(1), \dots, \Theta(N_p - 1))$. We define $\Theta_t = \Theta(t \bmod N_p)$ for all $t \in \mathbb{Z}_+$. The period of the scheduler is N_p .
2. In every period, the scheduler generates one packet from every source s .
3. Once a packet is generated a source's packet is delivered to the destination in d_s slots which is the delivery delay.

Properties (2) and (3) are required to ensure that the AoII remains bounded under periodic deterministic schedulers (also recall that links were assumed to be lossless).

A natural question that arises here is whether there exist periodic schedulers that satisfy the above properties. The SS-Tree policy, introduced in Chapter 4, is one such scheduler that satisfies these properties. Note that $(N_p, (d_s, \forall s))$ represent the parameters of a given periodic policy π_P . An example of a periodic schedule with period N_p is illustrated in Fig. 5.1(b), along with the corresponding d_s for node 11.

The source model considered here is the same piecewise-constant process used in Chapter 3 (refer to Section 3.2). Specifically, at the beginning of time slot t , the process $X^s(t)$ transitions to a new state with probability p_c^s , where the new state is drawn independently according to the probability density function $f_X^s(x)$. Otherwise, it remains in the same state with probability $q_c^s = 1 - p_c^s$.

The AoII of source s at node 0 is represented by the function $A^s(t)$, defined as follows:

$$A^s(t) = t - V^s(t), \quad (5.1)$$

Here, $V^s(t)$ denotes the most recent slot up to time t (including slot t) at which the remote estimate $\hat{X}^s(t)$ of the source process $X^s(t)$ was correct. Mathematically, it is given by, $V^s(t) = \max \left\{ t' : t' \leq t \text{ and } X^s(t') = \hat{X}^s(t') \right\}$. We note that both $\hat{X}^s(t)$ and $A^s(t)$ depend on the policy π . We define the average AoII for source s under policy π_p as follows:

$$\overline{A}^{\pi_p}(s) = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[A^s(t) | A^s(0) = 0], \quad (5.2)$$

where we assume that initial AoII is 0.

We define the average AoII of the system under policy π_P as

$$\overline{A}^{\pi_p} = \frac{1}{M} \sum_{s=1}^M \overline{A}^{\pi_p}(s) \quad (5.3)$$

The objective of this chapter is to characterize the AAoII in multihop networks under deterministic periodic sampling and scheduling policies. In the following sections, we derive analytical lower and upper bounds for the AAoII.

Remark 5.1. The AoII $A^s(t)$ for the source process $X^s(t)$ becomes 0 when the remote estimate and $X^s(t)$ become equal. For other source process models such as discrete finite state Markov chains, $A^s(t)$ may become 0 after being positive if $X^s(t)$ first changes and

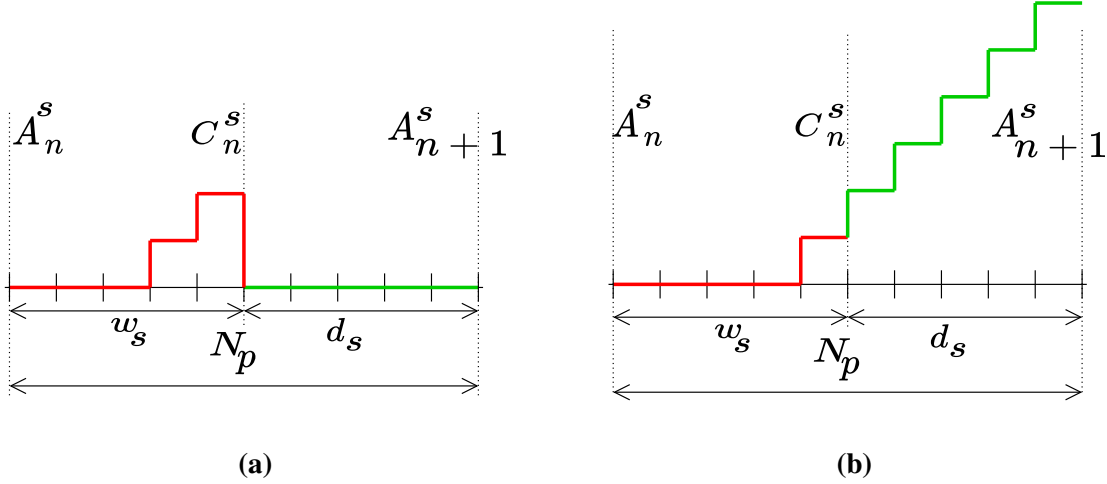


Figure 5.2: Illustration of the AoII evolution, $A^s(t)$, in the n^{th} slotframe, from A_n^s to C_n^s during Phase I (shown in red), and from C_n^s to A_{n+1}^s during Phase II (shown in green), for the case $C_n^s > 0$: (a) $A_{n+1}^s = 0$ with probability $q_c^{d_s}$, and (b) $A_{n+1}^s = C_n^s + d_s$ with probability $1 - q_c^{d_s}$.

then returns to a state which is equal to $\hat{X}^s(t)$. For such source processes, since characterizing AoII may be complex, [105] introduced the AoII+ metric which measures the time that has elapsed since the first change. The AoII characterization that we obtain for our simplified source model could be extended to AoII+ for finite state Markov chains (with same probability of leaving any state).

5.3 Analytical Characterization and Bounds

In this section, we consider a periodic scheduling policy π_P that satisfies the assumptions stated above. We characterize the AAoII, $\bar{A}^{\pi_P}$, as a function of the slotframe length N_P and the delay d_s for all s . To derive $\bar{A}^{\pi_P}$, we analyze the evolution of $A^s(t)$ for a tagged source s over one slotframe period of length N_P . Here, note that the slotframe is redefined with respect to the tagged source s so that it consists of two phases. Phase I corresponds to the waiting period before the packet from source s is generated. Let $w_s = N_P - d_s$ denote the time from the beginning of the slotframe to the packet generation instant. Phase II, defined over the interval $(w_s \leq t < N_P)$, represents the transmission and reception phase, during which the generated packet is delivered to the sink node 0 after a delay of d_s . We note that there are $w_s = N_P - d_s$ slots corresponding to Phase I and d_s slots corresponding to Phase II.

Consider the n^{th} slotframe, where $n \in \mathbb{Z}_+$. Let A_n^s denote the embedded AoII $A^s(t)$

at the beginning of the slotframe, and let C_n^s denote the embedded AoII $A^s(t)$ at the end of Phase I within the same slotframe. The evolution of $A^s(t)$ from A_n^s to C_n^s during Phase I, and from C_n^s to A_{n+1}^s during Phase II, is summarized in Table 5.2. The evolution of the AoII $A^s(t)$ across Phase I and Phase II within a slotframe is illustrated in Fig. 5.2. We note that the evolution of (A_n^s) forms a Markov chain with $A_n^s \in \mathbb{Z}_+$. This Markov chain is irreducible.¹ The state 0 is also aperiodic. Furthermore, it can be shown that the expected return time to state 0 is finite. Therefore, the Markov chain is irreducible, aperiodic, and positive recurrent, with a stationary distribution. In the following propositions, we consider the stationary version of the embedded AoII process A_n^s , denoted by A^s , and characterize the stationary probability $\mathbb{P}\{A^s = 0\}$, denoted by π_0^s , as well as the stationary expectation $\mathbb{E}[A^s]$.

Proposition 5.1. *The stationary probability $\pi_0^s = q_c^{d_s}$.*

The result follows from solving the balance equations for the A_n^s Markov chain (using transition probabilities from Table 5.2) and noting that from any state the probability of reaching 0 is $q_c^{d_s}$. The detailed proof is provided in Appendix 5.A. The π_0^s is used to obtain the following result.

Proposition 5.2. *The stationary expectation $\mathbb{E}[A^s] = N_p (q_c^{-d_s} - 1) + d_s q_c^{N_p} - q_c^{d_s} - (q_c - q_c^{d_s})/p_c$.*

The proof is based on moment matching and is provided in detail in Appendix 5.B. We now characterize the AAoII $\bar{A}^{\pi_p}$ as a function of N_p and d_s for a periodic policy in the following proposition.

Proposition 5.3. *The AAoII $\bar{A}^{\pi_p}$ for a periodic policy π_p with period N_p and delay d_s is $\frac{1}{M} \sum_{s=1}^M \bar{A}^{\pi_p}(s)$, where*

$$\begin{aligned} \bar{A}^{\pi_p}(s) = & \frac{d_s^2 p_c^2 + d_s p_c (p_c (2\mathbb{E}[A^s] + 2q_c^{d_s} + 2w_s - 1) - 2q_c^{d_s})}{2N_p p_c^2} \\ & + \frac{p_c^2 w_s (2\mathbb{E}[A^s] + w_s - 1) - 2q_c^{d_s+1} (p_c w_s + q_c^{N_p} - 1)}{2N_p p_c^2}, \end{aligned}$$

and $\mathbb{E}[A^s]$ is the embedded AoII from Proposition 5.2.

¹Starting from state 0, any other AoII value can be reached. For instance, the states $\{0, 1, \dots, N_p\}$ can be reached in one step, while the states $\{0, 1, \dots, 2N_p\}$ can be reached in two steps of the embedded chain, and so on. Moreover, state 0 can be reached from any AoII value with positive probability in a single step.

Table 5.2: Evolution of the embedded AoII A_n^s to A_{n+1}^s

Phase I: A_n^s to C_n^s		
C_n^s	if $A_n^s = 0$	0, w.p. $q_c^{w_s}$ x , w.p. $q_c^{w_s-x} p_c$, where $x \in \{1, 2, \dots, w_s\}$
	if $A_n^s > 0$	$A_n^s + w_s$ w.p. 1
Phase II: C_n^s to A_{n+1}^s		
A_{n+1}^s	if $C_n^s = 0$	0, w.p. $q_c^{d_s}$ x , w.p. $q_c^{d_s-x} p_c$, where $x \in \{1, 2, \dots, d_s\}$
	if $C_n^s > 0$	0 w.p. $q_c^{d_s}$ $C_n^s + d_s$ w.p. $1 - q_c^{d_s}$

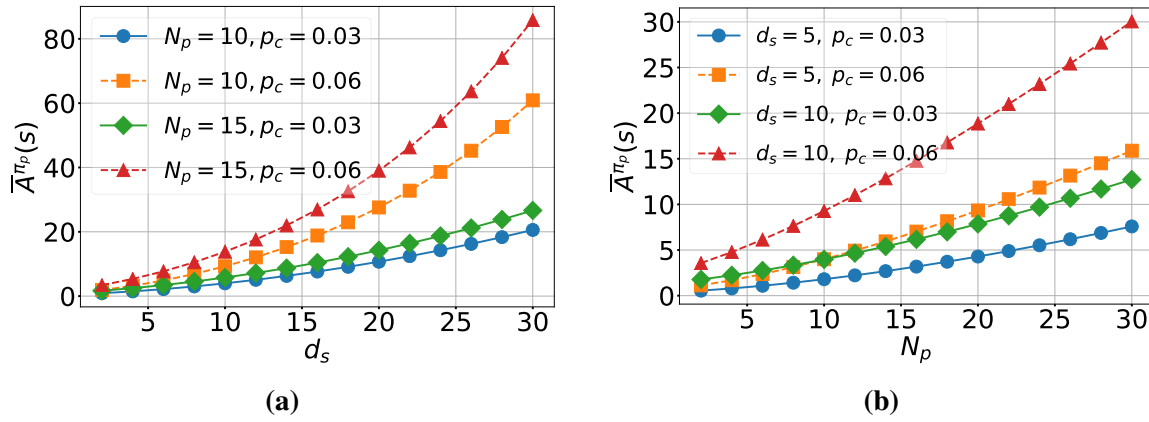


Figure 5.3: Illustration of the monotone non-decreasing property of $\bar{A}^{\pi_p}(s)$: (a) as a function of $d_s \in \{2, 4, \dots, 30\}$ for $N_p \in \{10, 15\}$ and $p_c \in \{0.03, 0.06\}$; (b) as a function of $N_p \in \{2, 4, \dots, 30\}$ for $d_s \in \{5, 10\}$ and $p_c \in \{0.03, 0.06\}$.

The above result is obtained using the Markov renewal reward theorem by expressing the AoII within a slotframe as the reward. The detailed proof is provided in Appendix 5.C. The parameters p_c and q_c in the characterization of $\bar{A}^{\pi_p}(s)$ may, in general, depend on the source s . We also note that, for a given policy π_P , the parameters $(N_P, d_s, \forall s)$ are not independent. For example, in a network with complete interference, $N_P \geq \sum_{s=1}^M d_s$. However, in order to derive upper and lower bounds on $\bar{A}^{\pi_P}$, we treat N_P and d_s as independent variables. Fig. 5.3 illustrates the monotone non-decreasing behavior of $\bar{A}^{\pi_P}(s)$ as a function of d_s for a fixed N_P , and as a function of N_P for a fixed d_s . A larger N_P reduces the update frequency, while a larger d_s increases the delay in packet reception at the sink node 0. In both cases, the probability that the source state changes (with probability p_c) before the packet is delivered increases, thereby leading to a higher AoII. This monotonic behavior is used in the following sections to derive bounds on $\bar{A}^{\pi_P}$ based on bounds on N_P and d_s .

5.3.1 Analytical Lower Bounds on AAoII

The following definitions are recalled from Chapter 4. In a tree network, a *top-subtree* refers to a subgraph rooted at a child node of node 0. A subtree rooted at $b \in \mathcal{R}_{top}$ is denoted by $\mathcal{T}_b$. Further, we denote the number of source nodes in $\mathcal{T}_b$ by $|\mathcal{T}_b| = n_b$, such that $\sum_{b=1}^B n_b = M$. An example with $B = 2$ and $n_1 = 6$ is shown in Fig. 5.1(a). A line network is defined as a linear graph with nodes indexed from 0 to M , where each node i forwards its packet to node $i - 1$, for $i \in \{1, 2, \dots, M\}$. A multiline network is a tree topology in which every top-subtree forms a line. We define a *t-line* as a linear subgraph of the tree consisting of the path from a leaf node in $\mathcal{T}_b$ to its root node $b \in \mathcal{R}_{top}$, and subsequently to node 0. For example, in Fig. 5.1(a), the path $4 \rightarrow 3 \rightarrow 1 \rightarrow 0$ forms a t-line.

Bounds on N_{lb} from Chapter 4: A lower bound, denoted by N_{lb} , on the period of any deterministic policy satisfying assumptions (1), (2), and (3) stated above was derived in Chapter 4 through Propositions 4.5 and 4.6. These bounds were derived by considering the minimum number of slots required to collect packets from all sources at node 0 under interference constraints. The first bound applies to multiline networks, where each subtree b contains n_b nodes. For a network under K -hop interference constraints, it has been shown that N_P is lower bounded by N_{lb} , where

$$N_{lb} = e_x + \sum_{b \in \mathcal{R}_{top}} e_b \quad (5.4)$$

where $e_b = \sum_{k=1}^{\lceil K/2 \rceil} (n_b - k + 1)$ and $e_x = \max_{b \in \mathcal{R}_{top}} (f_b - e_{b,o})^+$. Here, $f_b = t_b - e_b$ for all $b \in \mathcal{R}_{top}$, $t_b = \sum_{j=1}^{K+1} (n_b - j + 1)$, and $e_{b,o} = \sum_{j \neq b} e_j$.

Similarly, for a tree network under K -hop interference constraints, the period N_P has been shown to satisfy the lower bound,

$$N_{lb} = e_x + \sum_{b \in \mathcal{R}_{top}} e_b \quad (5.5)$$

with $e_b = \sum_{h=1}^{\lceil K/2 \rceil} (n_b - n_b(h - 1))$, $e_x = \max_{b \in \mathcal{R}_{top}} (f_b - e_{b,o})^+$, $f_b = \sum_{h=\lceil K/2 \rceil+1}^{\min(l_b, K)} f_b(h)$, and $e_{b,o} = \sum_{j \neq b} e_j$. Here, $n_b(h)$ denotes the number of nodes in subtree b that are within a hop distance of h from node 0, with $n_b(0) = 0$. Further, l_b denotes the length of the longest t-line in subtree b , and $f_b(h)$ denotes the number of packets forwarded by the node located at hop distance h from node 0 within subtree b .

Lower Bound $\bar{A}^{\pi_P^{LB}}$: For any periodic deterministic policy, the period satisfies $N_P \geq N_{lb}$ and the delivery delay satisfies $d_s \geq h_s$, where h_s denotes the hop distance of source s from

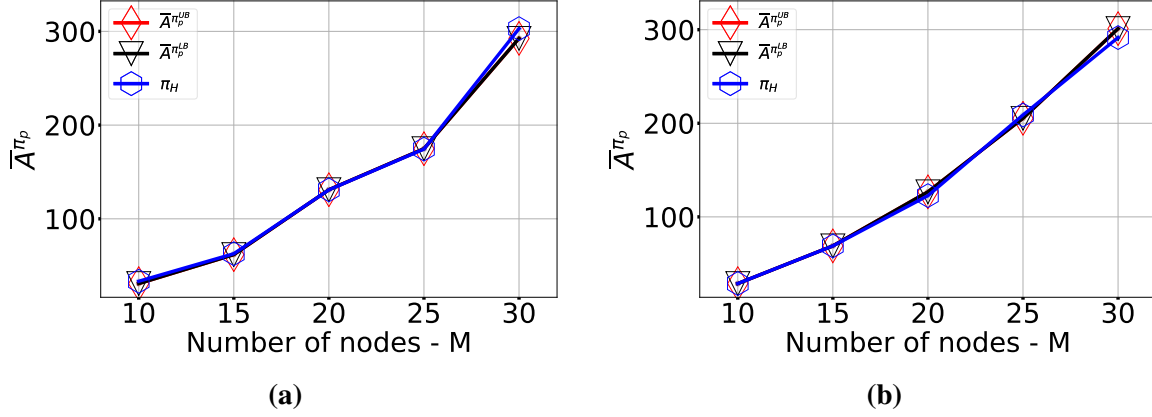


Figure 5.4: Lower bound ($\bar{A}^{\pi_p^{LB}}$) and upper bound ($\bar{A}^{\pi_p^{UB}}$) as a function of M for line networks with complete interference: (a) homogeneous and (b) heterogeneous cases. AoII under π_H is also shown for comparison.

node 0. The lower bound is then obtained by substituting N_{lb} for N_P and h_s for d_s , for all s , in the expression for $\bar{A}^{\pi_P}$ given in Proposition 5.3.

5.3.2 A Scheduling Policy and an Upper Bound on N_p

In this section, we consider the periodic deterministic sampling and scheduling policy SS-Tree (Continue-Service (CS) variant) presented in Algorithm 4 of Chapter 4. The algorithm generates a sequence of activation sets, $\Theta = (\theta(0), \theta(1), \dots, \theta(N_{ss}-1))$ denotes the period. This sequence is then repeated cyclically with period N_{ss} . The period N_{ss} and the delivery delays d_s , for all $s \in \{1, \dots, M\}$, are determined by simulating the SS-Tree algorithm over a single period.

Upper Bound $\bar{A}^{\pi_p^{UB}}$: The upper bound is computed by substituting the period N_{ss} and the delivery delays d_s , for all s , obtained from the SS-Tree algorithm into $\bar{A}^{\pi_P}$ as given in Proposition 5.3.

5.4 Simulations and Performance Comparisons

In this section, we compare the upper bound $\bar{A}^{\pi_p^{UB}}$ (corresponding to the AAoII of SS-Tree) with the analytical lower bound $\bar{A}^{\pi_p^{LB}}$ across various network topologies and source parameters. We also evaluate the AAoII performance of SS-Tree against a baseline non-periodic scheduling policy, denoted as π_H from Chapter 3, to compare periodic and non-periodic approaches. The comparisons are conducted over three network topologies: *line*,

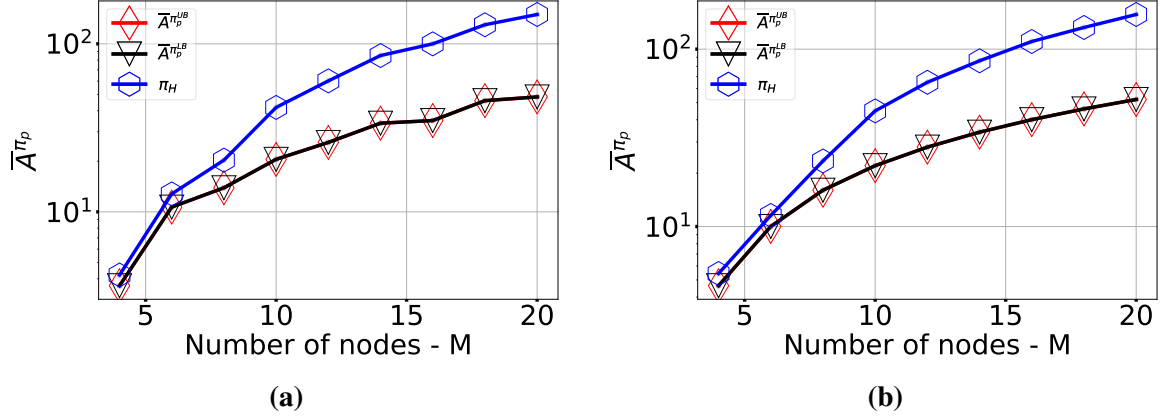


Figure 5.5: Lower bound ($\bar{A}_p^{\pi_p^{LB}}$) and upper bound ($\bar{A}_p^{\pi_p^{UB}}$) as a function of M for line networks with 5-hop interference: (a) homogeneous and (b) heterogeneous cases. AoII under π_H is also shown for comparison.

multiline, and *tree* networks, each with varying network sizes M . Furthermore, we consider two types of networks: *homogeneous* networks, where all sources $s \in \{1, 2, \dots, M\}$ share the same state change probability $p_c \in (0, 1)$, and *heterogeneous* networks, where each source s has a distinct state change probability $p_c^s \in (0, 1)$. Consider a network with $h_m = \max_s h^s$, representing the maximum hop distance from any source to the sink node 0. In the homogeneous case, all sources share a single state change probability p_c chosen from the interval $\left[\frac{1}{3h_m}, \frac{1}{2h_m}\right]$. For heterogeneous networks, p_c^s is assigned based on the hop distance h^s : sources with $h^s < h_m/2$ are assigned $p_c^s = \frac{1}{3h_m}$, while sources with $h^s \geq h_m/2$ are assigned $p_c^s = \frac{1}{2h_m}$. For each network, the analytical lower bound $\bar{A}_p^{\pi_p^{LB}}$ is computed using Proposition 5.3. The upper bound $\bar{A}_p^{\pi_p^{UB}}$ is obtained by simulating the SS-Tree policy over one period to determine N_{ss} and d_s for all sources. The AAoII for the non-periodic policy π_H is obtained through direct simulation.

For line networks, the average AoII as a function of M is shown in Figs. 5.4 and 5.5, corresponding to networks with complete interference and 5-hop interference, respectively. We observe that the periodic SS-Tree policy is near-optimal for both homogeneous and heterogeneous cases in the complete-interference scenario. Moreover, the performance of the non-periodic policy π_H is comparable to that of SS-Tree. In networks with complete interference, only a single link can be active in each slot. The SS-Tree algorithm is designed to generate and deliver packets in contiguous slots, ensuring $h_s = d_s$, which explains its near-optimal performance. For the 5-hop interference scenario, periodic policies still achieve lower AAoII compared with π_H , and SS-Tree remains near-optimal. The match is expected as the N_{ss} used in the upper bound is the same as the minimum N from the lower

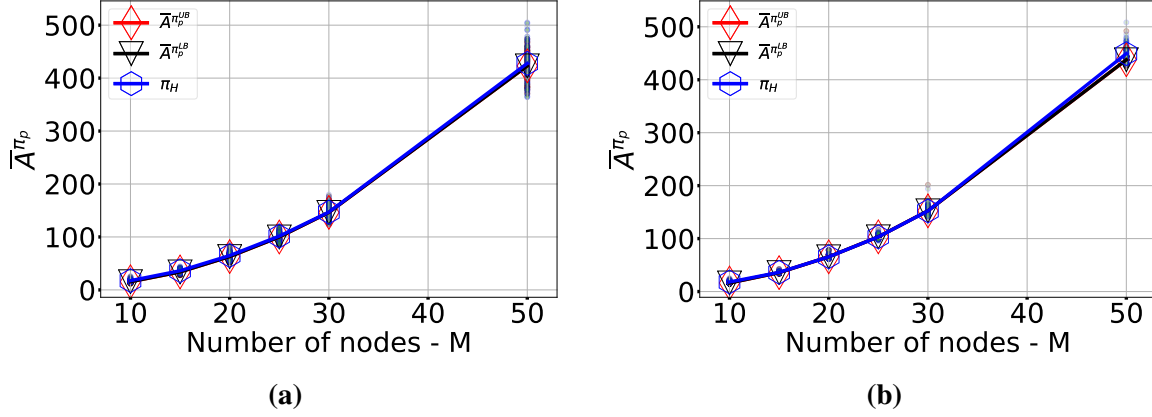


Figure 5.6: Lower bound ($\overline{A}_p^{\pi_p^{LB}}$) and upper bound ($\overline{A}_p^{\pi_p^{UB}}$) as a function of M for multiline networks with complete interference: (a) homogeneous and (b) heterogeneous cases. AoII under π_H is also shown for comparison.

bound.

We now consider multiline networks. Let n_i denote the number of nodes in the i -th child of the root node, for $i \in \{1, 2, \dots, b\}$. The values of n_i are drawn from a multinomial distribution with M total trials, b mutually exclusive outcomes, and equal probability for each outcome, ensuring that $\sum_{i=1}^b n_i = M$. In this study, we set $b = 2$ and generate 100 random graphs for each value of M . The AoII is visualized using scatter plots in Figs. 5.6 and 5.7, for networks with complete interference and 5-hop interference, respectively. Each point in the plots corresponds to the AAoII (upper or lower bound) for a specific network instance. For complete-interference networks, SS-Tree is observed to be near-optimal, while the non-periodic policy π_H exhibits performance comparable to the periodic policies.

Finally, the network topology considered is a tree network, generated by uniformly sampling from all possible trees with M nodes. Tree network is generated using a Prufer sequence-based method [93], and 100 random tree instances are generated for each network size M . The bounds are shown in Figs. 5.8 and 5.9 for networks with complete and 5-hop interference, respectively. For complete interference, SS-Tree again demonstrates near-optimal performance. Under 5-hop interference, periodic policies achieve lower AAoII than non-periodic policies. However, for multiline and tree topologies with K -hop interference, SS-Tree is no longer near-optimal. Since only one link can be scheduled at a time, the minimum N is always the sum of hop distances. This is used in both upper and lower bounds. We again conjecture that the lower bound on N is weak and leave its improvement for future work.

Figure 5.10 illustrates the AAoII bounds along with $\overline{A}^{\pi_H}$ for a homogeneous network

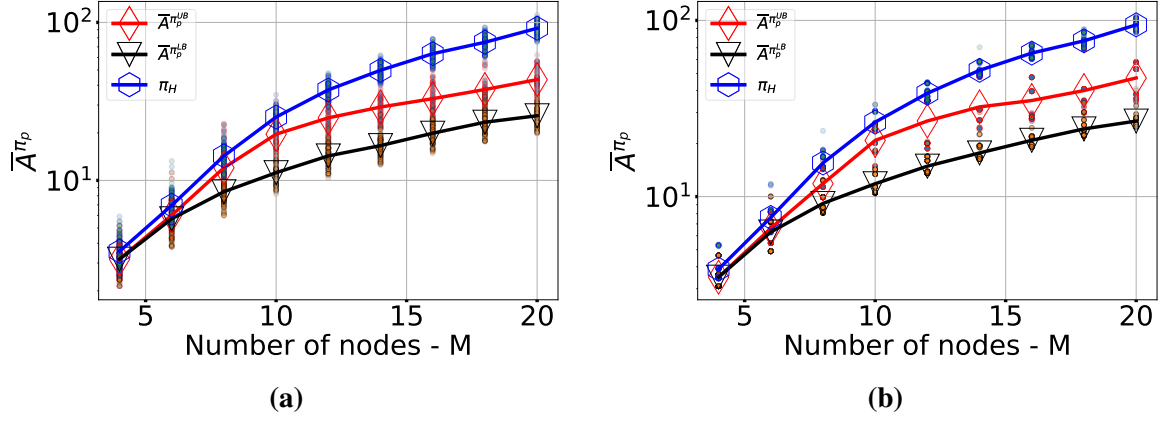


Figure 5.7: Lower bound ($\bar{A}_p^{\pi_p, \text{LB}}$) and upper bound ($\bar{A}_p^{\pi_p, \text{UB}}$) as a function of M for multiline networks with 5-hop interference: (a) homogeneous and (b) heterogeneous cases. AoII under π_H is also shown for comparison.

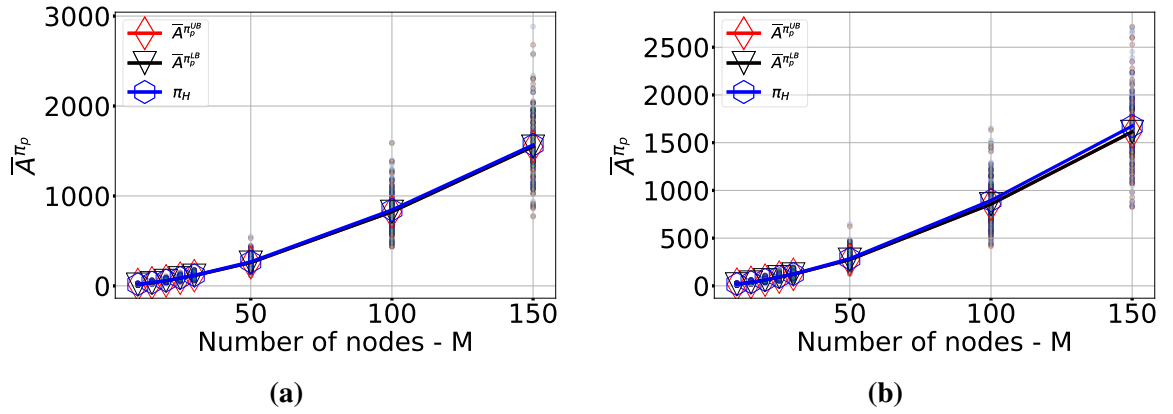


Figure 5.8: Lower bound ($\bar{A}_p^{\pi_p, \text{LB}}$) and upper bound ($\bar{A}_p^{\pi_p, \text{UB}}$) as a function of M for tree networks with complete interference: (a) homogeneous and (b) heterogeneous cases. AoII under π_H is also shown for comparison.

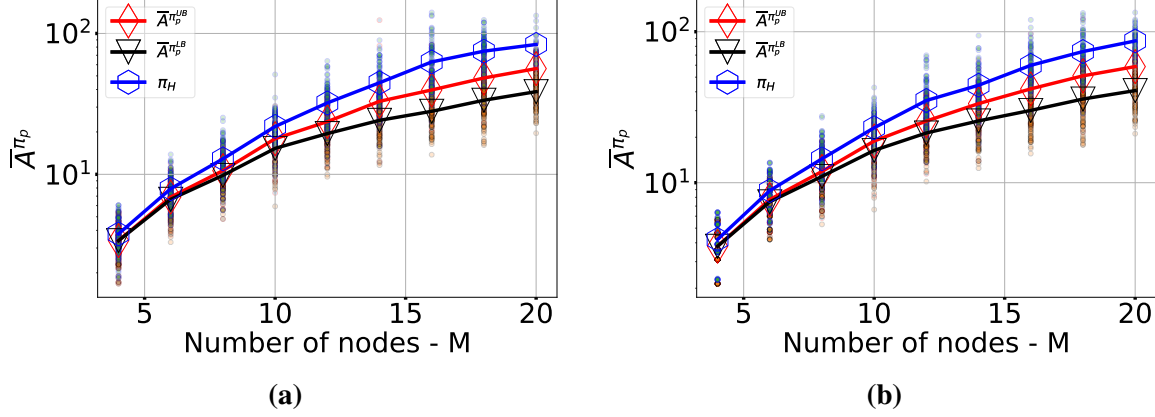


Figure 5.9: Lower bound ($\bar{A}^{\pi_p, \text{LB}}$) and upper bound ($\bar{A}^{\pi_p, \text{UB}}$) as a function of M for tree networks with 5-hop interference: (a) homogeneous and (b) heterogeneous cases. AoII under π_H is also shown for comparison.

with $K \in \{3, 5\}$, plotted as a function of p_c . In this scenario, SS-Tree is not near-optimal, although the periodic policy still outperforms π_H . We also analyzed the behavior of the normalized AAoII, $\bar{A}^{\pi_p}/M$, as the network size M increases, using the lower bound (see Fig. 5.11). For line and multiline networks, the normalized AAoII remains roughly constant for large M , indicating that AAoII grows linearly with network size. In contrast, this linear trend is not observed for tree topologies.

5.5 Discussion: Extension to the case of lossy links

In this section, we discuss extensions of our bounds to the case of networks with packet losses. The network model that we consider here is similar to that in Section 5.2 except that each packet transmission on a link fails with a probability p_f . Link failures are modelled as independent and identically distributed (IID) across slots as well as across links². We assume that when node i transmits to j using an activation $(s, (i, j))$ then i is aware of whether the transmission is a success or failure (say using link level acknowledgements). We assume that as in Section 4.6.1 if in a period a packet transmission fails then that packet is discarded. A new packet is generated for that source in the next period only. This is the extension of our earlier policies for handling link losses. The earlier upper bounds can be obtained under this assumption.

We now discuss how the lower bounds on AAoII can be extended. We note that the

²Although not discussed here, this can be extended to the case of links with heterogeneous failure probabilities.

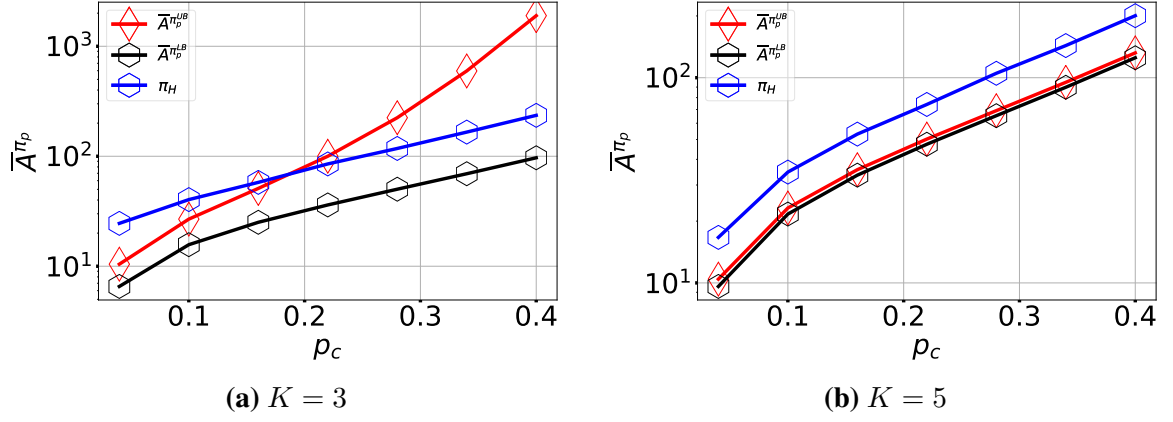


Figure 5.10: Lower bound ($\bar{A}^{\pi_p^{LB}}$) and upper bound ($\bar{A}^{\pi_p^{UB}}$) as a function of p_c for homogeneous tree networks with M sources under (a) 3-hop and (b) 5-hop interference. AoII under π_H is also shown for comparison.

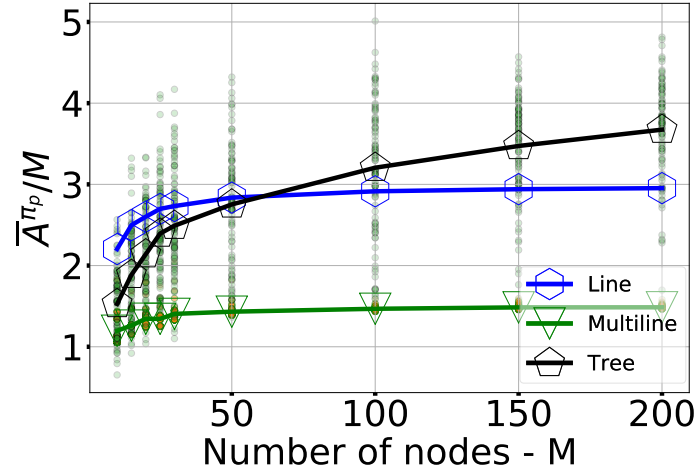


Figure 5.11: Normalized AAoII $\bar{A}^{\pi_p}/M$ for heterogeneous line, multiline, and tree networks as a function of M .

Table 5.3: Evolution of the embedded AoII A_n^s to A_{n+1}^s for a network with link losses

Phase I: A_n^s to C_n^s		
C_n^s	if $A_n^s = 0$	0, w.p. $q_c^{w_s}$ x , w.p. $q_c^{w_s-x}p_c$, where $x \in \{1, 2, \dots, w_s\}$
	if $A_n^s > 0$	$A_n^s + w_s$ w.p. 1
Phase II: C_n^s to A_{n+1}^s		
A_{n+1}^s	if $C_n^s = 0$	0, w.p. $q_c^{d_s}$ x , w.p. $q_c^{d_s-x}p_c$, where $x \in \{1, 2, \dots, d_s\}$
	if $C_n^s > 0$	0 w.p. $q_c^{d_s}(1-p_f)^{h_s}$ $C_n^s + d_s$ w.p. $1 - q_c^{d_s}(1-p_f)^{h_s}$

evolution of the AoII $A^s(t)$ for a tagged source, with the embedded AoII A_n^s follows the transitions in Table 5.3, with the other notations as before. We note that the only difference is in the last row, i.e., in the evolution from C_n^s to A_{n+1}^s . If $C_n^s > 0$ then the AoII $A_{n+1}^s = 0$ only if the packet is received successfully across h_s hops (i.e., with probability $(1-p_f)^{h_s}$) and the source does not change for d_s slots (with probability $q_c^{d_s}$). We now discuss the steps for deriving π_0 for a network with link losses, which is similar to Proposition 5.1.

As in the proof of Proposition 5.1 we consider the case when the system reaches stationarity. Suppose we consider the probability generating function $G(Z) = \mathbb{E}[Z^A]$ of the stationary AoII, which is assumed to be well-defined. Then $\mathbb{E}[Z^{A_{n+1}}] = \mathbb{E}[Z^{A_n}]$. By conditioning on the events $\{A_n = 0\}$ and $\{A_n > 0\}$, we have

$$G(Z) = \pi_0 \mathbb{E}[Z^{A_{n+1}} | A_n = 0] + (1 - \pi_0) \mathbb{E}[Z^{A_{n+1}} | A_n > 0]. \quad (5.6)$$

When $A_n = 0$, the intermediate AoII C_n at the packet generation instant $w_s = N_p - d_s$ satisfies $\mathbb{P}\{C_n = 0 | A_n = 0\} = q_c^{w_s}$, $\mathbb{P}\{C_n = x | A_n = 0\} = q_c^{w_s-x}p_c$, $x = 1, \dots, w_s$. Given $C_n = 0$, the AoII at the end of the cycle remains zero if no change occurs during the last d_s slots. Therefore,

$$\mathbb{E}[Z^{A_{n+1}} | C_n = 0] = q_c^{d_s} + \sum_{y=1}^{d_s} Z^y q_c^{d_s-y} p_c. \quad (5.7)$$

For $C_n = x > 0$, the packet generated at $t = w_s$ is already outdated by x , hence

$$\mathbb{E}[Z^{A_{n+1}} | C_n = x] = q_c^{d_s}(1-p_f)^{h_s} Z^0 + (1 - q_c^{d_s}(1-p_f)^{h_s}) Z^{x+d_s}. \quad (5.8)$$

Combining these terms we have $\mathbb{E}[Z^{A_{n+1}} | A_n = 0]$. When $A_n = a > 0$, the intermediate

AoII is deterministic with $C_n = a + w_s$, and

$$\mathbb{E}[Z^{A_{n+1}} | A_n = a] = q_c^{d_s}(1 - p_f)^{h_s} Z^0 + (1 - q_c^{d_s}(1 - p_f)^{h_s}) Z^{a+w_s+d_s}. \quad (5.9)$$

Setting $Z = 0$ gives the stationary probability $\pi_0 = \mathbb{P}\{A = 0\} = G(0)$. From the above conditional expectations, we obtain

$$\begin{aligned} \mathbb{E}[0^{A_{n+1}} | A_n = 0] &= \mathbb{P}\{A_{n+1} = 0 | A_n = 0\} = q_c^{d_s} \left(q_c^{w_s} + (1 - p_f)^{h_s} \sum_{x=1}^{w_s} q_c^{w_s-x} p_c \right) \\ &= q_c^{d_s} (q_c^{w_s} + (1 - p_f)^{h_s} (1 - q_c^{w_s})), \end{aligned} \quad (5.10)$$

and for $A_n > 0$,

$$\mathbb{E}[0^{A_{n+1}} | A_n = a] = \mathbb{P}\{A_{n+1} = 0 | A_n = a\} = q_c^{d_s}(1 - p_f)^{h_s}. \quad (5.11)$$

Substituting these results into the total expectation gives

$$\pi_0 = \pi_0 q_c^{d_s} (q_c^{w_s} + (1 - p_f)^{h_s} (1 - q_c^{w_s})) + (1 - \pi_0) q_c^{d_s} (1 - p_f)^{h_s}. \quad (5.12)$$

Hence, the stationary probability that the AoII at the beginning of a cycle is zero is

$$\pi_0 = \frac{q_c^{d_s}(1 - p_f)^{h_s}}{1 - q_c^{N_p}(1 - (1 - p_f)^{h_s})}. \quad (5.13)$$

The above expression reduces to $q_c^{d_s}$ as in Proposition 5.1 if $p_f = 0$. As illustrated above, we can similarly extend the derivations of Propositions 5.2, 5.3 with link losses to obtain fundamental lower bounds on AAoII.

5.6 Conclusions

In this chapter, we analyzed the AAoII for periodic deterministic sampling and scheduling policies in multihop wireless networks under K -hop interference constraints. A novel analytical characterization of the AAoII was developed, which enabled the derivation of lower and upper bounds for such periodic policies. Simulation results demonstrated the near-optimality of a policy from prior work for networks with complete interference and highlighted the advantages of periodic policies over a non-periodic scheduling policy across different scenarios.

Appendix

5.A Proof of Proposition 5.1

In this section, we derive the stationary probability π_0 corresponding to the event that the AoII is 0 at the beginning of a cycle.

Proof. Under stationarity, the evolution of the AoII satisfies $\mathbb{E}[A_{n+1}] = \mathbb{E}[A_n]$, or equivalently $\mathbb{E}[Z^{A_{n+1}}] = \mathbb{E}[Z^{A_n}] \forall Z$.

Let $G(Z) = \mathbb{E}[Z^A]$ denote the probability generating function (PGF) of the stationary AoII. By conditioning on the events $\{A_n = 0\}$ and $\{A_n > 0\}$, we have

$$G(Z) = \pi_0 \mathbb{E}[Z^{A_{n+1}} | A_n = 0] + (1 - \pi_0) \mathbb{E}[Z^{A_{n+1}} | A_n > 0]. \quad (5.14)$$

When $A_n = 0$, the intermediate AoII C_n at the packet generation instant $w_s = N_p - d_s$ satisfies $\mathbb{P}\{C_n = 0 | A_n = 0\} = q_c^{w_s}$, $\mathbb{P}\{C_n = x | A_n = 0\} = q_c^{w_s-x} p_c$, $x = 1, \dots, w_s$. Given $C_n = 0$, the AoII at the end of the cycle remains zero if no change occurs during the last d_s slots. Therefore,

$$\mathbb{E}[Z^{A_{n+1}} | C_n = 0] = q_c^{d_s} + \sum_{y=1}^{d_s} Z^y q_c^{d_s-y} p_c. \quad (5.15)$$

For $C_n = x > 0$, the packet generated at $t = w_s$ is already outdated by x , hence

$$\mathbb{E}[Z^{A_{n+1}} | C_n = x] = q_c^{d_s} Z^0 + (1 - q_c^{d_s}) Z^{x+d_s}. \quad (5.16)$$

Combining these terms yields $\mathbb{E}[Z^{A_{n+1}} | A_n = 0]$.

When $A_n = a > 0$, the intermediate AoII is deterministic with $C_n = a + w_s$, and

$$\mathbb{E}[Z^{A_{n+1}} | A_n = a] = q_c^{d_s} Z^0 + (1 - q_c^{d_s}) Z^{a+w_s+d_s}. \quad (5.17)$$

Setting $Z = 0$ gives the stationary probability $\pi_0 = \mathbb{P}\{A = 0\} = G(0)$. From the above conditional expectations, we obtain

$$\begin{aligned} \mathbb{E}[0^{A_{n+1}} | A_n = 0] &= \mathbb{P}\{A_{n+1} = 0 | A_n = 0\} = q_c^{d_s} \left(q_c^{w_s} + \sum_{x=1}^{w_s} q_c^{w_s-x} p_c \right) \\ &= q_c^{d_s} (q_c^{w_s} + (1 - q_c^{w_s})) = q_c^{d_s}, \end{aligned} \quad (5.18)$$

and for $A_n > 0$,

$$\mathbb{E}[0^{A_{n+1}} \mid A_n = a] = \mathbb{P}\{A_{n+1} = 0 \mid A_n = a\} = q_c^{d_s}. \quad (5.19)$$

Substituting these results into the total expectation gives

$$\pi_0 = \pi_0 q_c^{d_s} + (1 - \pi_0) q_c^{d_s} = q_c^{d_s}. \quad (5.20)$$

Hence, the stationary probability that the AoII at the beginning of a cycle is zero is

$$\pi_0 = q_c^{d_s}. \quad (5.21)$$

□

5.B Proof of Proposition 5.2

This appendix presents the derivation of the embedded AoII and its simplification into a closed-form expression under the periodic scheduling policy.

Proof. The embedded expectation is computed using the law of total expectation:

$$\mathbb{E}[A_{n+1}] = \mathbb{E}[\mathbb{E}[A_{n+1} \mid A_n]] = \sum_{a \geq 0} \pi_a \mathbb{E}[A_{n+1} \mid A_n = a].$$

Conditioning further on the intermediate AoII C_n at the packet-generation instant $w_s = N_p - d_s$, we have

$$\mathbb{E}[A_{n+1} \mid A_n = a] = \sum_c \mathbb{P}\{C_n = c \mid A_n = a\} \mathbb{E}[A_{n+1} \mid C_n = c].$$

We evaluate the two cases separately.

Case (a): $A_n = 0$. When $A_n = 0$, the distribution of C_n is $\mathbb{P}\{C_n = 0 \mid A_n = 0\} = q_c^{w_s}$, $\mathbb{P}\{C_n = x \mid A_n = 0\} = q_c^{w_s-x} p_c$, $x = 1, \dots, w_s$.

If $C_n = 0$, then after the remaining d_s slots, the final AoII equals $y \in \{0, 1, \dots, d_s\}$. The contribution to the expectation from the nonzero outcomes is

$$\mathbb{E}[A_{n+1} \mid C_n = 0] = \sum_{y=1}^{d_s} y q_c^{d_s-y} p_c.$$

If $C_n = x > 0$, then the final AoII equals 0 with probability $q_c^{d_s}$ and equals $x + d_s$ with probability $1 - q_c^{d_s}$. Therefore,

$$\mathbb{E}[A_{n+1} \mid C_n = x] = (1 - q_c^{d_s})(x + d_s).$$

Hence,

$$\begin{aligned} \mathbb{E}[A_{n+1} \mid A_n = 0] &= q_c^{w_s} \sum_{y=1}^{d_s} y q_c^{d_s-y} p_c \\ &+ \sum_{x=1}^{w_s} q_c^{w_s-x} p_c (1 - q_c^{d_s})(x + d_s). \end{aligned} \quad (5.22)$$

Case (b): $A_n = a > 0$. When $A_n = a > 0$, the intermediate AoII is deterministic:

$$C_n = a + w_s \quad (\text{with probability } 1).$$

Thus,

$$\mathbb{E}[A_{n+1} \mid A_n = a] = (1 - q_c^{d_s})(a + w_s + d_s) = (1 - q_c^{d_s})(a + N_p).$$

Averaging over π_a for $a > 0$ gives

$$\begin{aligned} \sum_{a>0} \pi_a \mathbb{E}[A_{n+1} \mid A_n = a] &= (1 - q_c^{d_s}) \sum_{a>0} \pi_a (a + N_p) \\ &= (1 - q_c^{d_s}) \left(\sum_{a \geq 0} \pi_a a + N_p \sum_{a>0} \pi_a \right) \\ &= (1 - q_c^{d_s}) (\mathbb{E}[A_n] + N_p(1 - \pi_0)). \end{aligned} \quad (5.23)$$

Under stationarity, $\mathbb{E}[A_n] = \mathbb{E}[A_{n+1}]$. Using (5.22) and (5.23), we obtain

$$\mathbb{E}[A_{n+1}] = \pi_0 \mathbb{E}[A_{n+1} \mid A_n = 0] + \sum_{a>0} \pi_a \mathbb{E}[A_{n+1} \mid A_n = a] \quad (5.24)$$

$$\begin{aligned} \mathbb{E}[A_{n+1}] &= \pi_0 \left[q_c^{w_s} \sum_{y=1}^{d_s} y q_c^{d_s-y} p_c \right. \\ &\quad \left. + \sum_{x=1}^{w_s} q_c^{w_s-x} p_c (1 - q_c^{d_s})(x + d_s) \right] \\ &\quad + (1 - q_c^{d_s}) (\mathbb{E}[A_{n+1}] + N_p(1 - \pi_0)). \end{aligned} \quad (5.25)$$

From the stationary result, $\pi_0 = q_c^{d_s}$ and $w_s = N_p - d_s$. Substituting these into (5.25)

and simplifying the resulting geometric sums gives

$$\mathbb{E}[A_{n+1}] = N_p(q_c^{-d_s} - 1) + d_s q_c^{N_p} - q_c^{d_s} - \frac{q_c - q_c^{d_s}}{p_c}. \quad (5.26)$$

Thus, the embedded expectation simplifies to the closed-form expression in (5.26). $\square$

5.C Proof of Proposition 5.3

In this section, we derive the average Age of Incorrect Information (AAoII) $\bar{A}^{\pi_p}$ for a user s under the periodic scheduling policy with period N_p , delay d_s , and stationary probability π_0 .

Proof. Consider a periodic scheduler with period N_p . For a given source s , let d_s denote the delay (in slots) from packet generation to delivery at the destination node 0, and define $w_s = N_p - d_s$ as the time from the beginning of the cycle to the packet generation instant within that cycle. Let A_n denote the AoII at the beginning of the n -th cycle, and C_n denote the intermediate AoII at time w_s (i.e., immediately after packet generation). Let p_c represent the state change probability.

Assume that the beginning of cycle AoII A_n has stationary distribution $\{\pi_a\}_{a \geq 0}$. The conditional transition probability of the AoII satisfies

$$\begin{aligned} \mathbb{P}\{A_{n+1} = a' \mid A_n = a\} &= \sum_c \mathbb{P}\{A_{n+1} = a', C_n = c \mid A_n = a\} \\ &= \sum_c \mathbb{P}\{A_{n+1} = a', C_n = c\} \\ &\quad \mathbb{P}\{C_n = c \mid A_n = a\}. \end{aligned} \quad (5.27)$$

The long-run average AoII is obtained by applying the renewal reward theorem over periodic cycles of length N_p . Since each period is identical, the time-average AoII equals the expected accumulated area (reward) within one cycle divided by N_p .

Assume a cycle that begins with $A_n = a$. Each cycle consists of two phases: Phase I, from the beginning of the cycle up to the packet generation time $w_s = N_p - d_s$; and Phase II, from w_s (immediately after generation) to the end of the cycle at time N_p , lasting d_s slots.

Conditioned on the intermediate AoII $C_n = c$ at time w_s , the total accumulated area under the AoII curve during one cycle is denoted as the *reward* R . Applying the law of

total expectation, the expected reward given the initial AoII $A_n = a$ is expressed as

$$\mathbb{E}[R \mid A_n = a] = \sum_{c \geq 0} \mathbb{P}\{C_n = c \mid A_n = a\} \mathbb{E}[R \mid A_n = a, C_n = c]. \quad (5.28)$$

We now evaluate the conditional cycle reward in each phase. During Phase I, if the cycle begins with AoII a and reaches c at time w_s , the cumulative area accumulated during the w_s slots is

$$R_1(a, c) = a w_s + \frac{(c - a)(c - a - 1)}{2}. \quad (5.29)$$

During Phase II, for $c > 0$, if $c > 0$ at the generation instant, the AoII at time w_s equals c . Over the remaining d_s slots until the end of the cycle, the AoII increases deterministically, and the accumulated area is

$$R_2(c) = c d_s + \frac{d_s(d_s - 1)}{2}. \quad (5.30)$$

During Phase II, for $c = 0$, if $c = 0$ at time w_s , the AoII resets to zero but may increase during the remaining d_s slots due to state changes in the monitored process. Let $\tilde{a}$ denote the AoII during this period, where $\mathbb{P}\{\tilde{a} = x\} = (1 - p_c)^{d_s - x} p_c$, $x = 1, \dots, d_s$. The expected accumulated area is then

$$R_3 = \mathbb{E}\left[\frac{\tilde{a}(\tilde{a} - 1)}{2}\right] = \sum_{x=1}^{d_s} (1 - p_c)^{d_s - x} p_c \frac{x(x - 1)}{2}. \quad (5.31)$$

Total expected cycle reward is obtained by combining both phases, the expected area within a cycle conditioned on $A_n = a$ is given by $\sum_a \pi_a \sum_c \mathbb{P}\{C_n = c \mid A_n = a\} R_1(a, c) + \sum_a \pi_a \sum_{c > 0} \mathbb{P}\{C_n = c \mid A_n = a\} R_2(c) + \mathbb{P}\{C_n = 0 \mid A_n = a\} R_3$. Averaging over the stationary distribution π_a of A_n and dividing by the cycle length N_p , the time-average AoII is obtained as

$$\begin{aligned} \overline{A}^{\pi_p} = \frac{1}{N_p} & \left[\sum_a \pi_a \sum_c \mathbb{P}\{C_n = c \mid A_n = a\} R_1(a, c) \right. \\ & + \sum_a \pi_a \left(\sum_{c > 0} \mathbb{P}\{C_n = c \mid A_n = a\} R_2(c) \right. \\ & \left. \left. + \mathbb{P}\{C_n = 0 \mid A_n = a\} R_3 \right) \right] \end{aligned} \quad (5.32)$$

Now, we derive each term in this expression. The first term is by substituting $R_1(a, c)$, we get

$$\begin{aligned} T_1 &= \sum_{a=0}^{\infty} \pi_a \sum_{c=0}^{w_s} \mathbb{P}\{C_n = c \mid A_n = a\} \left(a w_s + \frac{(c-a)(c-a-1)}{2} \right) \\ &= \sum_{a=0}^{\infty} \pi_a a w_s \sum_{c=0}^{w_s} \mathbb{P}\{C_n = c \mid A_n = a\} \\ &\quad + \sum_{a=0}^{\infty} \pi_a \sum_{c=0}^{w_s} \mathbb{P}\{C_n = c \mid A_n = a\} \frac{(c-a)(c-a-1)}{2}. \end{aligned}$$

Since $\sum_{c=0}^{w_s} \mathbb{P}\{C_n = c \mid A_n = a\} = 1$, the first term simplifies to

$$T_1 = \mathbb{E}[A_{n+1}] w_s + \sum_{a=0}^{\infty} \pi_a \sum_{c=0}^{w_s} \mathbb{P}\{C_n = c \mid A_n = a\} \frac{(c-a)(c-a-1)}{2}.$$

We split the remaining summation into two parts: $a = 0$ and $a > 0$.

$$\begin{aligned} S &= \sum_{a=0}^{\infty} \pi_a \sum_{c=0}^{w_s} \mathbb{P}\{C_n = c \mid A_n = a\} \frac{(c-a)(c-a-1)}{2} \\ &= \pi_0 \sum_{c=0}^{w_s} \mathbb{P}\{C_n = c \mid A_n = 0\} \frac{c(c-1)}{2} + \sum_{a>0} \pi_a \frac{w_s(w_s-1)}{2}. \end{aligned}$$

Here, for $a > 0$, we used the assumption that $C_n = a + w_s$ with probability 1, so $(c-a) = w_s$.

Defining $q_c = 1 - p_c$, the conditional distribution for $A_n = 0$ yields

$$\pi_0 \sum_{c=1}^{w_s} \mathbb{P}\{C_n = c \mid A_n = 0\} \frac{c(c-1)}{2} = \pi_0 \sum_{x=1}^{w_s} q_c^{w_s-x} p_c \frac{x(x-1)}{2}.$$

Therefore, the first term simplifies to

$$\begin{aligned} T_1 &= \mathbb{E}[A_{n+1}] w_s + \pi_0 \sum_{x=1}^{w_s} q_c^{w_s-x} p_c \frac{x(x-1)}{2} + (1 - \pi_0) \frac{w_s(w_s-1)}{2} \\ &= \frac{p_c^2 w_s (2\mathbb{E}[A_{n+1}] + w_s - 1) - 2\pi_0 q_c (p_c w_s + q_c^{w_s} - 1)}{2p_c^2} \end{aligned}$$

Similarly, the second term can be written as

$$T_2 = \sum_a \pi_a \left(\sum_{c>0} \mathbb{P}\{C_n = c \mid A_n = a\} R_2(c) \right).$$

Substituting the expression for $R_2(c)$, we get

$$\begin{aligned} T_2 &= \pi_0 \sum_{c=1}^{w_s} \mathbb{P}\{C_n = c \mid A_n = 0\} \left(c d_s + \frac{d_s(d_s - 1)}{2} \right) \\ &\quad + \sum_{a=1}^{\infty} \pi_a \left((a + w_s) d_s + \frac{d_s(d_s - 1)}{2} \right) \\ &= \pi_0 d_s \sum_{c=1}^{w_s} c \mathbb{P}\{C_n = c \mid A_n = 0\} \\ &\quad + \pi_0 \frac{d_s(d_s - 1)}{2} (1 - \mathbb{P}\{C_n = 0 \mid A_n = 0\}) \\ &\quad + \mathbb{E}[A_{n+1}] d_s + (1 - \pi_0) \left(d_s w_s + \frac{d_s(d_s - 1)}{2} \right). \end{aligned}$$

Using the conditional distribution for $A_n = 0$,

$$\sum_{c=1}^{w_s} c \mathbb{P}\{C_n = c \mid A_n = 0\} = \sum_{x=1}^{w_s} x q_c^{w_s-x} p_c,$$

and $1 - \mathbb{P}\{C_n = 0 \mid A_n = 0\} = 1 - q_c^{w_s}$, where $q_c = 1 - p_c$. Substituting these expressions, we obtain

$$\begin{aligned} T_2 &= \pi_0 d_s \sum_{x=1}^{w_s} x q_c^{w_s-x} p_c + \pi_0 \frac{d_s(d_s - 1)}{2} (1 - q_c^{w_s}) + \mathbb{E}[A_{n+1}] d_s + (1 - \pi_0) \left(d_s w_s + \frac{d_s(d_s - 1)}{2} \right) \\ &= \frac{d_s \left[p_c \left(2\mathbb{E}[A_{n+1}] - d_s (\pi_0 q_c^{w_s} - 1) - \pi_0 q_c^{w_s} + 2\pi_0 + 2w_s - 1 \right) \right]}{2p_c} + \frac{d_s \left[2\pi_0 (q_c^{w_s} - 1) \right]}{2p_c} \end{aligned}$$

Similarly, the third term is $T_3 = \sum_{a=0}^{\infty} \pi_a \mathbb{P}\{C_n = 0 \mid A_n = a\} R_3$. Substituting the expression for R_3 , note that $C_n = 0$ occurs only when $A_n = 0$ with probability $q_c^{w_s}$. For $a > 0$, $\mathbb{P}\{C_n = 0 \mid A_n = a\} = 0$. Therefore, the third term becomes $T_3 = \pi_0 q_c^{w_s} R_3 = \pi_0 q_c^{w_s} \sum_{x=1}^{d_s} q_c^{d_s-x} p_c \frac{x(x-1)}{2}$

$$T_3 = \frac{\pi_0 q_c^{w_s} \left(d_s p_c^2 (d_s + 1) - 2p_c (d_s - q_c^{d_s} + 1) - 2q_c^{d_s} + 2 \right)}{2p_c^2}, .$$

Substituting T_1, T_2, T_3 , and π_0 from (5.21) into (5.32), the time-average AoII is obtained as given in (5.33).

$$\begin{aligned} \overline{A}^{\pi_p} = & \frac{d_s^2 p_c^2 + d_s p_c (p_c (2\mathbb{E}[A_{n+1}] + 2q_c^{d_s} + 2w_s - 1) - 2q_c^{d_s})}{2N_p p_c^2} \\ & + \frac{p_c^2 w_s (2\mathbb{E}[A_{n+1}] + w_s - 1) - 2q_c^{d_s+1} (p_c w_s + q_c^{N_p} - 1)}{2N_p p_c^2} \end{aligned} \quad (5.33)$$

This expression characterizes the complete AAoII under the periodic scheduling policy. $\square$

Chapter 6

Summary and Scope for Future Work

This thesis investigated the design of sampling and scheduling algorithms for optimizing information freshness in multi-hop wireless sensor networks used for remote estimation applications. Information freshness was quantified using two metrics: the AoI which measures the staleness of received updates, and the AoII which additionally captures the correctness of the estimated process at the receiver. The study focused on time-slotted wireless networks with interference constraints, packet losses, and centralized scheduling at the sink node. The main contributions of this thesis are summarized as follows.

In Chapters 2 and 3, this thesis addressed the problem of minimizing the information freshness metrics AAOI and AAoII, respectively, in multi-hop wireless networks with interference and random packet losses using non-periodic policies. These chapters, focused on the design of centralized scheduling policies, motivated by practical protocols that employ centralized schedulers, such as IEEE 802.15.4e TSCH [4]. For the minimization of average AoI in Chapter 2, a priority index-based centralized sampling and scheduling policy is proposed, motivated by a RMAB formulation for a policy class that maintains at most one packet per source in the network at any given time. Based on this RMAB formulation, we derive an analytically tractable Priority Index-based policy that is centrally implementable, as it requires only the AoI information of each source at the sink node. Simulation results demonstrate that the proposed approach significantly outperforms existing policies in minimizing AAOI, with the performance gains becoming more pronounced as the network size increases or the link success probability decreases.

In Chapter 3, the study was extended to AoII, which captures both the freshness and correctness of the remote estimate. Centralized sampling and scheduling policies were developed to minimize average AoII in multi-hop networks with interference constraints and lossy links. Due to the limited information available at the sink, the scheduling problem naturally leads to a POMDP. To obtain tractable solutions, RMAB-based formulations were

again proposed for a policy class that maintains only one packet per source in the network at a time. A priority-index policy was characterized to jointly determine which sources to sample and which links to activate in each time slot. Using this index, an analytically tractable centralized scheduling policy was derived that can be practically implemented. Simulation results demonstrated that these policies achieve significant performance improvements compared to heuristic approaches.

In Chapters 4 and 5, we investigated the design of periodic deterministic sampling and scheduling policies for minimizing the information freshness metrics AAoI and AAoII, respectively, in multi-hop networks under interference constraints. The idea of periodic scheduling is motivated by practical industrial communication protocols such as WirelessHART [5], ISA100.11a [29], and IEEE 802.15.4e TSCH [4]. For the AAoI minimization problem, the optimization problem is formulated as an integer programming problem, and a linear programming relaxation is employed to obtain numerical lower bounds. Analytical lower bounds on the minimum achievable AAoI were also derived. For line networks, a periodic policy called SS-Line was proposed based on a node-state transition scheme that guarantees schedules satisfying K -hop interference constraints. This policy was further extended to two heuristic algorithms, SS-Tree and CS, for general tree networks. Extensive simulation results demonstrated that the proposed policies achieve AAoI values that closely match the derived analytical and numerical lower bounds, and outperform existing multi-hop AoI-aware policies such as Age-Difference and Age-Debt in a wide range of scenarios. We also extended the proposed policy to general system performance metrics, such as general functions of AoI, and weighted AoI. For the study of AAoII, a novel analytical characterization of AAoII for periodic policies was developed, which was subsequently used to derive lower and upper bounds on AAoII. Simulation results demonstrated the near-optimality of the non-periodic policy proposed in Chapter 3 for networks with complete interference. The results also highlighted that periodic deterministic policies can outperform the non-periodic policy proposed in Chapter 3 across different network scenarios.

Overall, this thesis developed analytical characterization, performance bounds, and practically implementable scheduling policies to enhance information freshness in multi-hop wireless networks. By leveraging MDP frameworks, RMAB formulations, and implementable scheduling strategies, this work advances the design of multi-hop networks for reliable and timely remote estimation, thereby supporting information freshness in next-generation industrial and cyber-physical systems. Although this thesis addresses several important challenges in the design of freshness-aware multi-hop networks, a number of

interesting directions remain open for future research. Some of the key directions are summarized below. One promising direction for future research is the use of reinforcement learning (RL) for learning adaptive scheduling policies in multi-hop wireless networks with unknown parameters, dynamic channel conditions, or time-varying network topologies. In particular, the scheduling problem can be formulated as a MDP, where the global network state consists of the AoI values of all sources at intermediate nodes, and the action space corresponds to feasible link activation sets under the given interference constraints. To explore this direction, we evaluated a deep Q-network (DQN)-based scheduling policy for a simple line network with three source nodes and $p_s = 1$. As illustrated in Fig. 6.1, the AAoI, computed as a moving average over 100 time slots, is shown for a DQN-based policy trained using a discounted cost criterion with a discount factor of 0.99 over a total of 10000 slots. While these preliminary results demonstrate the feasibility of RL-based scheduling, the practical applicability of such approaches remains limited by the rapidly increasing state and action spaces, topology-dependent state representations, and the need for extensive parameter tuning. Therefore, developing scalable and topology-aware RL frameworks, potentially leveraging graph neural networks, remains an important direction for future research.

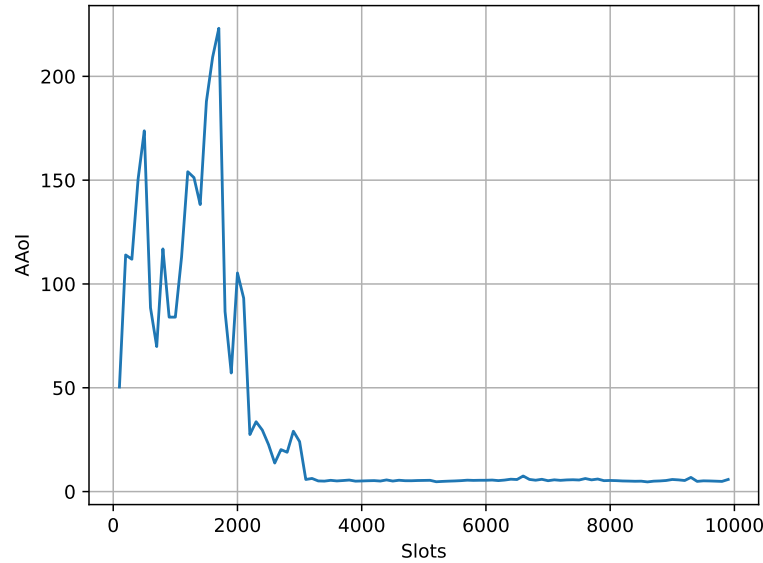


Figure 6.1: AAoI as a function of the time slots for a DQN-based scheduling policy implemented for a line network with number of source nodes $M = 3$ source nodes and reliable links ($p_s = 1$).

In addition, developing distributed scheduling algorithms for multi-hop networks is an important future direction. In large-scale networks, centralized control may become infeasible due to significant communication and computational overhead. Therefore, decentralized policies that rely only on locally available information while maintaining good information freshness performance represent a practical and challenging research problem. Another important direction is the extension to heterogeneous networks, where sources may have different process dynamics, update generation rates, or reliability requirements. Such settings introduce additional complexity into the scheduling problem and require the development of new analytical frameworks. Extending periodic scheduling policies to more general network topologies beyond tree networks, including networks with multiple communication channels or lossy links, also remains an open problem. Furthermore, allocating multiple transmission slots within a period to improve reliability is another interesting extension. Finally, practical implementation and testbed-based validation of the proposed algorithms constitute an important future step, as real-world experiments will help evaluate the feasibility, scalability, and robustness of the proposed policies in industrial IoT and cyber-physical systems.

Bibliography

- [1] Y. Sun, Y. Polyanskiy, and E. Uysal, “Sampling of the Wiener process for remote estimation over a channel with random delay,” *IEEE Trans. Information Theory*, vol. 66, no. 2, pp. 1118–1135, 2020.
- [2] R. D. Yates and S. K. Kaul, “The age of information: Real-time status updating by multiple sources,” *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1807–1827, 2019.
- [3] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, “The age of incorrect information: A new performance metric for status updates,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2215–2228, 2020.
- [4] D. De Guglielmo, S. Brienza, and G. Anastasi, “IEEE 802.15. 4e: A survey,” *Computer Communications*, vol. 88, pp. 1–24, 2016.
- [5] Q. Wang and J. Jiang, “Comparative examination on architecture and protocol of industrial wireless sensor network standards,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2197–2219, 2016.
- [6] E. Delfani and N. Pappas, “Optimizing Information Freshness in IoT Systems with Update Rate Constraints: A Token-Based Approach,” in *IFIP Networking Conference*, 2024, pp. 213–221.
- [7] C. Guo, X. Wang, L. Liang, and G. Y. Li, “Age of information, latency, and reliability in intelligent vehicular networks,” *IEEE Network*, vol. 37, no. 6, pp. 109–116, 2023.
- [8] J. Pan, Y. Li, R. Chai, S. Xia, and L. Zuo, “Age of Information Aware Trajectory Planning of UAV,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 6, pp. 2344–2356, 2024.

- [9] V. Tripathi, R. Talak, and E. Modiano, “Information freshness in multihop wireless networks,” *IEEE/ACM Transactions on Networking*, pp. 1–16, 2022.
- [10] S. Kaul, R. Yates, and M. Gruteser, “Real-time status: How often should one update?” in *Proceedings IEEE INFOCOM*, 2012, pp. 2731–2735.
- [11] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” in *International Conference on Computer Communications*, 2016, pp. 1–9.
- [12] Z. Jiang, S. Zhou, Z. Niu, and C. Yu, “A unified sampling and scheduling approach for status update in multiaccess wireless networks,” in *IEEE Conference on Computer Communications*, 2019, pp. 208–216.
- [13] J. Zhong, R. D. Yates, and E. Soljanin, “Two freshness metrics for local cache refresh,” in *IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 1924–1928.
- [14] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, “Towards an effective age of information: Remote estimation of a markov source,” in *IEEE Conference on Computer Communications Workshops*, 2018, pp. 367–372.
- [15] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, B. Soret *et al.*, “Semantic communications in networked systems: A data significance perspective,” *IEEE Network*, vol. 36, no. 4, pp. 233–240, 2022.
- [16] S. Saha, H. Singh Makkar, V. Bala Sukumaran, and C. R. Murthy, “On the relationship between mean absolute error and age of incorrect information in the estimation of a piecewise linear signal over noisy channels,” *IEEE Communications Letters*, vol. 26, no. 11, pp. 2576–2580, 2022.
- [17] F. Kauer, M. Köstler, and V. Turau, “Reliable wireless multi-hop networks with decentralized slot management: An analysis of IEEE 802.15.4 DSME,” *arXiv preprint arXiv:1806.10521*, 2018.
- [18] D. Van Leemput, J. Hoebeke, and E. De Poorter, “Integrating battery-less energy harvesting devices in multi-hop industrial wireless sensor networks,” *IEEE Communications Magazine*, vol. 62, no. 7, pp. 66–73, 2024.

- [19] G. Tanganelli, A. Virdis, and E. Mingozzi, “Enabling multi-hop forwarding in flowpans through software-defined networking,” in *International Symposium on "A World of Wireless, Mobile and Multimedia Networks"*, 2019, pp. 1–9.
- [20] M. Rakibul Islam, M. Bokhtiar-Al-Zami, B. Paul, R. Palit, J.-C. Grégoire, and S. Islam, “Performance Evaluation of Multi-Hop LoRaWAN,” *IEEE Access*, vol. 11, pp. 50 929–50 945, 2023.
- [21] X. Liu, A. Belogaev, and J. Famaey, “Symbol-Synchronous Communication for Ultra-Low-Power Multi-Hop Ambient IoT Networks,” *arXiv preprint arXiv:2603.23084*, 2026.
- [22] P. Papadimitratos, A. D. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, “Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation,” *IEEE Communications Magazine*, vol. 47, no. 11, pp. 84–95, 2009.
- [23] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2011, pp. 350–358.
- [24] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, “Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.
- [25] V. Addanki and L. Iannone, “Moving a step forward in the quest for deterministic networks (DetNet),” in *IFIP Networking Conference (Networking)*, 2020, pp. 458–466.
- [26] S. Farahani, *ZigBee Wireless Networks and Transceivers*. Newnes, 2008.
- [27] M. R. Palattella, P. Thubert, X. Vilajosana, T. Watteyne, Q. Wang, and T. Engel, “6TiSCH wireless industrial networks: Determinism meets IPv6,” *Internet of things: challenges and opportunities*, pp. 111–141, 2014.
- [28] F. Norman, T. Pascal, V. Balazs, and F. Janos, “RFC 8655: Deterministic networking architecture,” *Internet Engineering Task Force*, 2019.

- [29] X. Jin, F. Kong, L. Kong, W. Liu, and P. Zeng, “Reliability and temporality optimization for multiple coexisting wireless smart networks in industrial environments,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6591–6602, 2017.
- [30] “802.15. 4e-2012—IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),” *IEEE Standard Association: Piscataway, NJ, USA*, 2012.
- [31] C. Kam, S. Kompella, and A. Ephremides, “Age of information under random updates,” in *IEEE International Symposium on Information Theory*, 2013, pp. 66–70.
- [32] L. Huang and E. Modiano, “Optimizing age-of-information in a multi-class queueing system,” in *International Symposium on Information Theory*, 2015, pp. 1681–1685.
- [33] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, “Age of information: An introduction and survey,” vol. 39, no. 5, pp. 1183–1210, 2021.
- [34] A. Kosta, N. Pappas, and V. Angelakis, “Age of information: A new concept, metric, and tool,” *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017.
- [35] A. Karakasidis, P. Vassiliadis, and E. Pitoura, “ETL queues for active data warehousing,” in *Proceedings of the 2nd international workshop on Information quality in information systems*, 2005, pp. 28–39.
- [36] H. Yu, L. Breslau, and S. Shenker, “A scalable web cache consistency architecture,” *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 163–174, 1999.
- [37] S. Chen and K. Nahrstedt, “Distributed QoS routing with imprecise state information,” in *International Conference on Computer Communications and Networks*. IEEE, 1998, pp. 614–621.
- [38] R. D. Yates and S. K. Kaul, “Status updates over unreliable multiaccess channels,” in *International Symposium on Information Theory*, 2017, pp. 331–335.
- [39] R. V. Bhat, R. Vaze, and M. Motani, “Minimization of age of information in fading multiple access channels,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1471–1484, 2021.

- [40] Z. Chen, N. Pappas, E. Björnson, and E. G. Larsson, “Optimizing information freshness in a multiple access channel with heterogeneous devices,” *IEEE Open Journal of the Communications Society*, vol. 2, pp. 456–470, 2021.
- [41] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, “Optimal sampling and scheduling for timely status updates in multi-source networks,” *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 4019–4034, 2021.
- [42] R. Talak, S. Karaman, and E. Modiano, “Minimizing age-of-information in multi-hop wireless networks,” in *Annual Allerton Conference on Communication, Control, and Computing*, 2017, pp. 486–493.
- [43] A. M. Bedewy, Y. Sun, and N. B. Shroff, “The age of information in multihop networks,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, p. 1248–1257, 2019.
- [44] S. Farazi, A. G. Klein, and D. R. Brown, “Fundamental bounds on the age of information in general multi-hop interference networks,” in *IEEE Conference on Computer Communications Workshops*, 2019, pp. 96–101.
- [45] A. Arafa and S. Ulukus, “Age-minimal transmission in energy harvesting two-hop networks,” in *IEEE Global Communications Conference*, 2017, pp. 1–6.
- [46] Q. Liu, H. Zeng, and M. Chen, “Minimizing AoI With Throughput Requirements in Multi-Path Network Communication,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 3, pp. 1203–1216, 2022.
- [47] S. Farazi, A. G. Klein, J. A. McNeill, and D. Richard Brown, “On the age of information in multi-source multi-hop wireless status update networks,” in *International Workshop on Signal Processing Advances in Wireless Communications*, 2018, pp. 1–5.
- [48] M. Bastopcu and S. Ulukus, “Information freshness in cache updating systems,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1861–1874, 2021.
- [49] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger, “Age-optimal constrained cache updating,” in *IEEE International Symposium on Information Theory*, 2017, pp. 141–145.

- [50] R. D. Yates, “The age of gossip in networks,” in *International Symposium on Information Theory*. IEEE, 2021, pp. 2984–2989.
- [51] A. Nikkhah, A. Ephremides, and N. Pappas, “Age of actuation in a wireless power transfer system,” in *IEEE Conference on Computer Communications Workshops*. IEEE, 2023, pp. 1–6.
- [52] C. Kam, S. Kompella, and A. Ephremides, “Age of incorrect information for remote estimation of a binary markov source,” in *IEEE Conference on Computer Communications Workshops*, 2020, pp. 1–6.
- [53] Y. Chen and A. Ephremides, “Analysis of age of incorrect information under generic transmission delay,” in *IEEE Conference on Computer Communications Workshops*, 2023, pp. 1–8.
- [54] S. Kriouile and M. Assaad, “Minimizing the age of incorrect information for unknown markovian source,” *arXiv preprint arXiv:2210.09681*, 2022.
- [55] Y. Chen and A. Ephremides, “Preempting to minimize age of incorrect information under transmission delay,” *IEEE Transactions on Networking*, vol. 34, pp. 668–680, 2026.
- [56] Y. Chen and A. Ephremides, “Minimizing age of incorrect information for unreliable channel with power constraint,” in *IEEE Global Communications Conference*, 2021, pp. 1–6.
- [57] A. Nayak, A. E. Kalør, F. Chiariotti, and P. Popovski, “A decentralized policy for minimization of age of incorrect information in slotted ALOHA systems,” in *IEEE International Conference on Communications*, 2023, pp. 1688–1693.
- [58] A. Munari, “Age of incorrect information in random access channels without feedback,” *arXiv preprint arXiv:2308.06034*, 2023.
- [59] V. Tripathi, N. Jones, and E. Modiano, “Fresh-CSMA: A distributed protocol for minimizing age of information,” *Journal of Communications and Networks*, vol. 25, no. 5, pp. 556–569, 2023.
- [60] A. Zakeri, M. Moltafet, and M. Codreanu, “Semantic-aware sampling and transmission in real-time tracking systems: A POMDP approach,” *IEEE Transactions on Communications*, pp. 1–1, 2024.

- [61] R. Talak, S. Karaman, and E. Modiano, “Distributed scheduling algorithms for optimizing information freshness in wireless networks,” in *IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications*, 2018, pp. 1–5.
- [62] A. Alibotaiken, S. Wang, O. T. Ajayi, and Y. Cheng, “A survey of freshness-aware wireless networking with reinforcement learning,” 2025. [Online]. Available: <https://arxiv.org/abs/2512.21412>
- [63] B. Sombabu, A. Mate, D. Manjunath, and S. Moharir, “Whittle Index for AoI-Aware Scheduling,” in *International Conference on COMMunication Systems & NETWORKS*, 2020, pp. 630–633.
- [64] V. Tripathi and E. Modiano, “A Whittle index approach to minimizing functions of age of information,” *IEEE/ACM Transactions on Networking*, vol. 32, no. 6, pp. 5144–5158, 2024.
- [65] M. Chen, K. Wu, and L. Song, “A Whittle index approach to minimizing age of multi-packet information in IoT network,” *IEEE Access*, vol. 9, pp. 31 467–31 480, 2021.
- [66] Y.-P. Hsu, “Age of information: Whittle index for scheduling stochastic arrivals,” in *International Symposium on Information Theory*, 2018, pp. 2634–2638.
- [67] M. K. C. Shisher and Y. Sun, “How does data freshness affect real-time supervised learning?” in *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2022, pp. 31–40.
- [68] C. Ari, M. K. C. Shisher, E. Uysal, and Y. Sun, “Goal-oriented communications for remote inference under two-way delay with memory,” in *IEEE International Symposium on Information Theory (ISIT)*, 2024, pp. 1179–1184.
- [69] Y. Chen and A. Ephremides, “Scheduling to minimize age of incorrect information with imperfect channel state information,” *Entropy*, vol. 23, no. 12, 2021. [Online]. Available: <https://www.mdpi.com/1099-4300/23/12/1572>
- [70] M. Salimnejad and N. Pappas, “On the age of information in a two-user multiple access setup,” *Entropy*, vol. 24, no. 4, 2022. [Online]. Available: <https://www.mdpi.com/1099-4300/24/4/542>

- [71] V. Tripathi, R. Talak, and E. Modiano, “Age of information for discrete time queues,” *arXiv preprint arXiv:1901.10463*, 2019.
- [72] S. K. Kaul, R. D. Yates, and M. Gruteser, “Status updates through queues,” in *Conference on Information Sciences and Systems*, 2012, pp. 1–6.
- [73] T. Z. Ornee and Y. Sun, “Sampling for remote estimation through queues: Age of information and beyond,” in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2019, pp. 1–8.
- [74] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, “Scheduling policies for minimizing age of information in broadcast wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2637–2650, 2018.
- [75] R. Talak, I. Kadota, S. Karaman, and E. Modiano, “Scheduling policies for age minimization in wireless networks with unknown channel state,” in *International Symposium on Information Theory*, 2018, pp. 2564–2568.
- [76] Y.-P. Hsu, E. Modiano, and L. Duan, “Age of information: Design and analysis of optimal scheduling algorithms,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 561–565.
- [77] R. D. Yates, “Lazy is timely: Status updates by an energy harvesting source,” in *International Symposium on Information Theory*, 2015, pp. 3008–3012.
- [78] Q. Wang, H. Chen, C. Zhao, Y. Li, P. Popovski, and B. Vucetic, “Optimizing information freshness via multiuser scheduling with adaptive noma/oma,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1766–1778, 2022.
- [79] S. Wang, O. T. Ajayi, and Y. Cheng, “An analytical approach for minimizing the age of information in a practical CSMA network,” in *Conference on Computer Communications*, 2024, pp. 1721–1730.
- [80] N. Akar, S. Liyanaarachchi, and S. Ulukus, “Scalable cyclic schedulers for age of information optimization in large-scale status update systems,” in *Conference on Computer Communications Workshops*, 2024, pp. 1–7.
- [81] A. Alabbasi and V. Aggarwal, “Joint information freshness and completion time optimization for vehicular networks,” *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 1118–1129, 2022.

- [82] S. Wang, Y. Cheng, L. X. Cai, and X. Cao, “Minimizing the age of information for monitoring over a WiFi network,” in *Global Communications Conference*, 2022, pp. 383–388.
- [83] S. Wang and Y. Cheng, “Deep learning-augmented SHS model for accurate AoI analysis in heterogeneous unsaturated CSMA networks,” in *Conference on Computer Communications*, 2025, pp. 1–10.
- [84] Z. Liu, Z. Chen, L. Luo, M. Hua, W. Li, and B. Xia, “Age of information-based scheduling for wireless device-to-device communications using deep learning,” in *Wireless Communications and Networking Conference*, 2021, pp. 1–6.
- [85] M. K. Chowdhury Shisher, H. Qin, L. Yang, F. Yan, and Y. Sun, “The age of correlated features in supervised learning based forecasting,” in *Conference on Computer Communications Workshops*, 2021, pp. 1–8.
- [86] X. Zhou, W. Liang, K. Yan, W. Li, K. I.-K. Wang, J. Ma, and Q. Jin, “Edge-enabled two-stage scheduling based on deep reinforcement learning for internet of everything,” *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3295–3304, 2023.
- [87] Z. Huang, C. Guo, J. Chen, and B. Liao, “AoI-aware dynamic user scheduling in vehicular networks based on soft reinforcement learning,” in *Vehicular Technology Conference*, 2023, pp. 1–6.
- [88] A. Elgabli, H. Khan, M. Krouka, and M. Bennis, “Reinforcement learning based scheduling algorithm for optimizing age of information in ultra reliable low latency networks,” in *Symposium on Computers and Communications*, 2019, pp. 1–6.
- [89] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. Alexander, “RPL: IPv6 routing protocol for low-power and lossy networks,” Internet Engineering Task Force (IETF), Fremont, CA, USA, RFC 6550, 2012.
- [90] A. R. Urke, Ø. Kure, and K. Øvsthus, “A survey of 802.15.4 TSCH schedulers for a standardized industrial internet of things,” *Sensors*, vol. 22, no. 1, p. 15, 2021.
- [91] R. R. Weber and G. Weiss, “On an index policy for restless bandits,” *Journal of Applied Probability*, vol. 27, no. 3, pp. 637–648, 1990.
- [92] A. Kumar, D. Manjunath, and J. Kuri, *Wireless networking*. Elsevier, 2008.

- [93] E. W. Weisstein, “Prüfer code,” From MathWorld—A Wolfram Web Resource. <https://mathworld.wolfram.com/PrueferCode.html>.
- [94] Z. Tang, Z. Sun, N. Yang, and X. Zhou, “Whittle index-based scheduling policy for minimizing the cost of age of information,” *IEEE Communications Letters*, vol. 26, no. 1, pp. 54–58, 2022.
- [95] K. Huang, W. Liu, Y. Li, and B. Vucetic, “To Retransmit or Not: Real-Time Remote Estimation in Wireless Networked Control,” in *International Conference on Communications*, 2019, pp. 1–7.
- [96] E. Zhibankova, A. Khakimov, E. Markova, and Y. Gaidamaka, “The age of information in wireless cellular systems: Gaps, open problems, and research challenges,” *Sensors*, vol. 23, no. 19, p. 8238, 2023.
- [97] A. Tonini, P. Redweik, M. Painho, and M. Castelli, “Remote estimation of target height from unmanned aerial vehicle (UAV) images,” *Remote Sensing*, vol. 12, no. 21, p. 3602, 2020.
- [98] J. Lu, Y. Miao, Y. Huang, W. Shi, X. Hu, X. Wang, and J. Wan, “Evaluating an unmanned aerial vehicle-based remote sensing system for estimation of rice nitrogen status,” in *International Conference on Agro-Geoinformatics*, 2015, pp. 198–203.
- [99] A. Maatouk, M. Assaad, and A. Ephremides, “The age of incorrect information: An enabler of semantics-empowered communication,” *IEEE Transactions on Wireless Communications*, vol. 22, no. 4, pp. 2621–2635, 2022.
- [100] B. Buyukates, A. Soysal, and S. Ulukus, “Age of information in multihop multicast networks,” *Journal of Communications and Networks*, vol. 21, no. 3, pp. 256–267, 2019.
- [101] Y. Chen and A. Ephremides, “Scheduling to minimize age of incorrect information with imperfect channel state information,” *Entropy*, vol. 23, no. 12, p. 1572, 2021.
- [102] S. Kriouile and M. Assaad, “Minimizing the age of incorrect information for real-time tracking of Markov remote sources,” in *IEEE International Symposium on Information Theory*, 2021, pp. 2978–2983.
- [103] M. Ayik, E. T. Ceran, and E. Uysal, “Optimization of AoII and QAoII in multi-user links,” in *IEEE Conference on Computer Communications Workshops*, 2023, pp. 1–6.

- [104] J. Niño-Mora, “Dynamic priority allocation via restless bandit marginal productivity indices,” *Transactions in Operations Research*, vol. 15, pp. 161–198, 2007.
- [105] Q. Liu, C. Li, Y. T. Hou, W. Lou, J. H. Reed, and S. Kompella, “Wireless scheduling to optimize age of information based on earliest update time,” *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6352–6366, 2023.
- [106] C. Papadimitriou and J. Tsitsiklis, “The complexity of optimal queueing network control,” in *Proceedings of the IEEE 9th Annual Conference on Structure in Complexity Theory*, 1994, pp. 318–322.
- [107] H. C. Tijms, *A first course in stochastic models*. John Wiley and sons, 2003.
- [108] N. Akbarzadeh and A. Mahajan, “Conditions for indexability of restless bandits and an $O(K^3)$ algorithm to compute whittle index,” *Advances in Applied Probability*, vol. 54, no. 4, pp. 1164–1192, 2022.
- [109] A. Maatouk, S. Kriouile, M. Assad, and A. Ephremides, “On the optimality of the Whittle’s index policy for minimizing the age of information,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1263–1277, 2021.
- [110] R. Boppana and M. M. Halldórsson, “Approximating maximum independent sets by excluding subgraphs,” *BIT Numerical Mathematics*, vol. 32, no. 2, pp. 180–196, 1992.
- [111] R. R. Weber and G. Weiss, “On an index policy for restless bandits,” *Journal of applied probability*, vol. 27, no. 3, pp. 637–648, 1990.
- [112] K. Liu, R. Weber, and Q. Zhao, “Indexability and whittle index for restless bandit problems involving reset processes,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 7690–7696.
- [113] M. Chen, Y. Xiao, Q. Li, and K.-c. Chen, “Minimizing age-of-information for fog computing-supported vehicular networks with deep Q-learning,” in *International Conference on Communications*, 2020, pp. 1–6.
- [114] J. Li, J. Tang, and Z. Liu, “Minimizing age of information for mobile edge computing empowered industrial internet of things,” in *10th Int. Workshop on Signal Design and Its Appl. in Commun.*, 2022, pp. 1–5.

- [115] S. Gandham, Y. Zhang, and Q. Huang, “Distributed time-optimal scheduling for convergecast in wireless sensor networks,” *Computer Networks*, vol. 52, no. 3, pp. 610–629, 2008.
- [116] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, “Fast data collection in tree-based wireless sensor networks,” *IEEE Transactions on Mobile computing*, vol. 11, no. 1, pp. 86–99, 2012.
- [117] R. Soua, P. Minet, and E. Livolant, “MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks,” in *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)*, 2012, pp. 91–100.
- [118] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, “Throughput and fairness guarantees through maximal scheduling in wireless networks,” *IEEE Transactions on Information Theory*, vol. 54, no. 2, pp. 572–594, 2008.
- [119] X. Wu, R. Srikant, and J. R. Perkins, “Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 595–605, 2007.
- [120] A. Makhorin, “GNU Linear Programming Kit (GLPK),” 2012. [Online]. Available: <https://www.gnu.org/software/glpk/>
- [121] Gurobi Optimization, LLC, “Gurobi optimizer,” 2024. [Online]. Available: <http://www.gurobi.com>
- [122] T. Zhang, S. Chen, Z. Chen, Z. Tian, Y. Jia, M. Wang, and D. O. Wu, “AoI and PAoI in the IoT-based multisource status update system: Violation probabilities and optimal arrival rate allocation,” *IEEE Internet of Things Journal*, vol. 10, no. 23, pp. 20 617–20 632, 2023.
- [123] T. Zhang, Z. Chen, Z. Tian, L. Zhen, Y. Jia, M. Wang, D. O. Wu, and T. Q. S. Quek, “On the timeliness of the stalest stream among multiple status updating streams,” *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 13 622–13 635, 2024.
- [124] T. Zhang, Z. Chen, Z. Tian, Y. Jia, M. Wang, and T. Q. S. Quek, “Real-time monitoring Ornstein–Uhlenbeck process: Improving fidelity under freshness guarantees,” *IEEE Communications Letters*, vol. 27, no. 6, pp. 1510–1514, 2023.

- [125] H. W. Lenstra Jr, “Integer programming with a fixed number of variables,” *Mathematics of operations research*, vol. 8, no. 4, pp. 538–548, 1983.
- [126] C. Kam, S. Kompella, and A. Ephremides, “Effect of message transmission diversity on status age,” in *International Symposium on Information Theory*. IEEE, 2014, pp. 2411–2415.
- [127] M. Costa, M. Codreanu, and A. Ephremides, “Age of information with packet management,” in *IEEE International Symposium on Information Theory*, 2014, pp. 1583–1587.

List of Publications

Publications Related to This Thesis

Refereed Journals

- J1. Nibin Raj, Vineeth B. S., and Pratik Sharma, “Periodic Deterministic Sampling and Scheduling Algorithms for Minimum Age of Information in Time-Slotted Multihop Networks,” *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2024.3523024.
- J2. Nibin Raj and Vineeth B. S., “Design of Centralized Scheduling Policies for Minimizing Average Age of Incorrect Information in Multi-hop Wireless Networks with Interference,” *IEEE Access*, doi: 10.1109/ACCESS.2026.3674431.
- J3. Nibin Raj and Vineeth B. S., “Priority-index based policies for minimizing average AoI in multi-hop network convergecast with complete interference and link losses”, in *IEEE Wireless Communications Letters*, vol. 15, pp. 4215-4219, 2026, doi: 10.1109/LWC.2026.3715677.

Refereed Conferences

- C1. Nibin Raj and Vineeth B. S., “On Average Age of Incorrect Information in Multihop Wireless Networks with Interference for Periodic Sampling and Scheduling Policies,” in *Proceedings of National Conference on Communications (NCC)*, 2026.

Other Publications Related to Network Design

Refereed Conferences

- C2. Nibin Raj and Vineeth B. S., “IndoorRSSINet - Deep Learning Based 2D RSSI Map Prediction for Indoor Environments with Application to Wireless Localization”,

- IEEE International Conference on COMmunication Systems & NETworkS (COMSNETS), Bangalore, India, 2023, pp. 609-616,
doi: 10.1109/COMSNETS56262.2023.10041394.
- C3. Nibin Raj, “Indoor RSSI Prediction Using Machine Learning for Wireless Networks”, IEEE International Conference on COMmunication Systems & NETworkS (COMSNETS), Bangalore, India, 2021, pp. 372-374,
doi:10.1109/COMSNETS51098.2021.9352852.
- C4. Nibin Raj, D.V.V. Sai Teja, and Vineeth B.S., “Pano2RSSI: Generation of RSSI Maps for a Room Environment from a Single Panoramic Image”, IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), New Delhi, India, 2020, pp. 1-6, doi: 10.1109/ANTS50601.2020.9342750.