AGE OF INFORMATION ANALYSIS AND OPTIMIZATION IN WIRELESS NETWORKS

by

Alper Köse

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2015

M.S., Electrical and Electronic Engineering, École Polytechnique Fédérale de Lausanne, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2024

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisors, Prof. Emin Anarım and Prof. Mutlu Koca, for their invaluable guidance, patience, and support throughout this research journey. Their expertise, encouragement, and constructive feedback have been instrumental in shaping this thesis.

I extend my heartfelt thanks to my family, Prof. Halil Köse and Prof. Süheyla Köse, for their unwavering love, encouragement, and understanding during this challenging phase of my academic pursuit. Their constant support has been my source of strength and motivation.

Special gratitude is extended to my wife, İrem Köse, whose enduring affection, patience, and empathy have provided a steadfast foundation of support. Her selflessness and words of encouragement have been essential throughout this academic endeavor.

I am also grateful to my colleagues and friends İbrahim Kahraman, Oğuzhan Sayın and Arda Atalık for their collaboration, camaraderie, insightful discussions and moral support.

Lastly, I would like to acknowledge the funding support provided by TUBITAK (The Scientific and Technological Research Council of Türkiye) under projects numbered as 119N154 and 123E225, and by Boğaziçi University Research Projects Fund under project number 19842 which made this research possible.

ABSTRACT

AGE OF INFORMATION ANALYSIS AND OPTIMIZATION IN WIRELESS NETWORKS

The age of information (AoI) has emerged as a crucial metric in wireless communication systems, representing the freshness of information from its generation to its reception. This thesis focuses on optimizing AoI within wireless networks to improve the timeliness and reliability of information delivery and begins with a comprehensive review of existing literature on AoI, examining its significance, metrics, and challenges in modern wireless networks. One key area of investigation is the optimization of dynamic user pairing in downlink non-orthogonal multiple access (NOMA) systems. By dynamically pairing users based on AoI metrics, these algorithms aim to minimize information staleness while maximizing network efficiency. Furthermore, the thesis presents a novel scheduling strategy tailored for priority-aware Internet of things (IoT) networks to optimize AoI. This strategy prioritizes data packets based on their urgency, ensuring timely delivery of critical information. The impact of block coding on AoI in centralized IoT systems is also explored, leveraging insights from binary erasure channel (BEC) and binary symmetric channel (BSC). The effectiveness of block coding techniques in reducing AoI is evaluated through theoretical analysis and numerical experiments. Moreover, the role of network coding in improving information freshness and reliability across diverse Internet of things (IoT) networks is investigated. This includes examining its impact on AoI in multi-source multi-hop IoT networks, providing insights into optimizing AoI in complex network scenarios. In conclusion, this thesis presents a comprehensive exploration of AoI optimization in wireless networks, spanning algorithm design, network architecture, coding strategies and contributes to the evolution of next-generation wireless communication systems towards achieving timely and reliable information delivery.

ÖZET

KABLOSUZ AĞLARDA BİLGİ YAŞI ANALİZİ VE OPTİMİZASYONU

Bilgi yaşı (AoI), kablosuz iletişim sistemlerinde kritik bir ölçüt olarak ortaya çıkmıştır ve bilginin tazeliğini temsil etmektedir. Bu tez, kablosuz ağlarda AoI'yi optimize etmeye odaklanarak bilgi iletiminin zamanında ve güvenilirliğini artırmayı amaçlamakta ve modern kablosuz ağlardaki AoI'nin önemini, ölçümlerini ve zorluklarını inceleyerek mevcut literatürün kapsamlı bir değerlendirmesiyle başlamaktadır. Araştırmanın anahtar alanlarından biri, aşağıya doğru yarı-ortogonal çoklu erişim sistemlerinde dinamik kullanıcı eşleme optimizasyonudur. Kullanıcıları AoI ölçümlerine dayalı olarak dinamik olarak eşleyerek, bu algoritmalar bilgi tazeliğini en aza indirgerken ağ verimliliğini en üst düzeye çıkarmayı amaçlamaktadır. Ayrıca, tez, öncelikli farkındalıklı nesnelerin interneti (IoT) ağları için özelleştirilmiş bir zamanlama stratejisi sunmaktadır. Bu strateji, kritik bilgilerin zamanında iletilmesini sağlayarak veri paketlerini aciliyetlerine göre önceliklendirmektedir. Merkezi IoT sistemlerinde blok kodlamanın AoI üzerindeki etkisi de ikili silme kanalı ve ikili simetrik kanal araçlarından faydalanarak incelenmektedir. Blok kodlama tekniklerinin AoI'yi azaltmadaki etkinliği teorik analiz ve simulasyon çalışmalarıyla değerlendirilmektedir. Dahası, ağ kodlamanın çeşitli IoT ağlarında bilgi tazeliğini ve güvenilirliğini artırmadaki rolü incelenmektedir. Bu, çoklu kaynaklı çoklu atlama IoT ağlarında AoI üzerindeki etkisini inceleyerek karmaşık ağ senaryolarında AoI'nin optimize edilmesine yönelik iç görüler sunmaktadır. Sonuç olarak, bu tez, kablosuz ağlarda AoI eniyilemesinin kapsamlı bir çalışmasını sunmakta, algoritma tasarımı, ağ mimarisi, kodlama stratejileri gibi alanları kapsamakta ve zamanında ve güvenilir bilgi iletimini sağlamak için gelecek nesil kablosuz iletişim sistemlerinin evrimine katkıda bulunmaktadır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES x
LIST OF TABLES xiii
LIST OF SYMBOLS xiv
LIST OF ACRONYMS/ABBREVIATIONS xvi
1. INTRODUCTION 1
1.1. Thesis Organization 8
2. LITERATURE REVIEW 9
2.1. Internet of Things 11
2.2. Age of Information 16
2.3. AoI in IoT 18
2.3.1. Performance Analysis for AoI in IoT Networks 18
2.3.2. Design Issues Under AoI Constraints 20
2.4. AoI Optimization in IoT 23
2.4.1. Joint Optimization of AoI and Energy Efficiency 24
2.4.2. AoI Optimization for Multiple Sources 26
2.4.3. AoI Optimization for Correlated Data 27
2.4.4. AoI Optimization in Multi-Hop Networks 28
2.4.5. AoI Optimization in Relay Networks 30
2.4.6. Effect of Preprocessing on AoI Optimization 31
2.5. Learning Based Methods for AoI Optimization 34
2.5.1. AoI Optimization with Reinforcement Learning 34
2.5.2. AoI Based Trajectory Optimization for Data Collection Model . 37
2.6. Scheduling and Sampling Issues for AoI in IoT 40
2.6.1. Scheduling Optimization 40
2.6.2. Impact of Varying Packet Size on Scheduling 42

2.6.3. Sampling Optimization 43
2.7. Experimental and Application Based Research 45
2.8. AoX in IoT 47
2.9. Conclusion and Discussion 49
3. AGE OF INFORMATION MINIMIZING DYNAMIC USER PAIRING IN DOWNLINK NOMA SYSTEMS 51
3.1. Introduction 51
3.2. User Pairing Benchmarks in NOMA 52
3.3. System Model 53
3.3.1. AoI Model 53
3.3.2. NOMA System Model and the Outage Probabilities 54
3.4. Problem Formulation and Graph-Theoretical Approach to AoI 57
3.4.1. Construction of the Conflict Graph 58
3.4.2. The MWIS Solution 61
3.4.2.1. The Computational Complexity 62
3.5. Simulation Results 62
3.6. Conclusion 64
4. A NOVEL SCHEDULING STRATEGY IN PRIORITY-AWARE IOT NETWORKS FOR AGE OF INFORMATION OPTIMIZATION 65
4.1. Introduction 65
4.2. System Model and Problem Formulation 66
4.3. Knapsack Strategy 69
4.4. Benchmarks 72
4.4.1. Epsilon-Greedy Algorithm 72
4.4.2. Whittle's Index Algorithm 73
4.5. Simulation Results 73
4.6. Conclusion 77
5. IMPACT OF BLOCK CODING ON AGE OF INFORMATION 78
5.1. Introduction 78
5.2. System Model 79
5.3. Block-Error Probability Derivations 80

5.3.1. Block-Error Probability for Block Codes 80
5.3.2. Block-Error Probability for Polar Coding 82
5.4. AoI Analysis 84
5.5. Simulation Results 86
5.6. Conclusion 89
6. IMPACT OF NETWORK CODING ON AGE OF INFORMATION 90
6.1. Introduction 90
6.2. Preliminaries on Network Coding 93
6.2.1. Encoding 93
6.2.2. Decoding 94
6.2.3. Packet Combining 95
6.3. System Model and Problem Formulation 95
6.4. Two-User Multicast Scheme 99
6.4.1. Peak and Average Expected AoI without Coding 100
6.4.2. Peak and Average Expected AoI with Network Coding 102
6.4.3. Comparison of Peak and Average Expected AoIs with or without Coding 103
6.5. General Multicast Scheme 104
6.5.1. Peak and Average Expected AoI without Coding 104
6.5.2. Peak and Average Expected AoI with Coding 106
6.5.3. Comparison of Peak and Average Expected AoIs with or without Coding 111
6.6. Behavioral Analyses 112
6.7. Simulation Results 114
6.7.1. UAV IoT 115
6.7.2. Military IoT 116
6.7.3. Household IoT 117
6.7.4. A General Perspective 119
6.8. Conclusion 120
7. IMPACT OF NETWORK CODING ON AGE OF INFORMATION IN MULTI-SOURCE MULTI-HOP IOT NETWORKS 123

7.1. Introduction . . . 123
7.2. System Model . . . 124
7.3. AoI Analyses . . . 127
7.3.1. Transmission From Sources to Servers . . . 128
7.3.1.1. $N_t = N_s = 1$ . . . 128
7.3.1.2. $N_t = 1$, $N_s > 1$ . . . 129
7.3.1.3. $N_t, N_s > 1$ . . . 130
7.3.2. Uncoded Transmission From Servers to Monitors . . . 131
7.3.2.1. $N_s = 1, N_r > 1$ . . . 131
7.3.2.2. $N_s, N_r > 1$ . . . 132
7.3.3. Network Coded Transmission From Servers to Monitors . . . 133
7.4. Numerical Results . . . 134
7.5. Conclusion . . . 137
8. IMPACT OF NETWORK CODING ON AGE OF INFORMATION IN ENERGY HARVESTING ENVIRONMENT . . . 138
8.1. Introduction . . . 138
8.2. Energy Harvesting Model . . . 139
8.3. AoI Analysis . . . 140
8.4. Simulation Results . . . 144
8.4.1. A Brief Note on the Relation of ω and h . . . 147
8.5. Conclusion . . . 148
9. CONCLUSION . . . 149
REFERENCES . . . 150
APPENDIX A: COPYRIGHT FOR REUSE OF THE FIGURES . . . 179

LIST OF FIGURES

Figure 2.1. Application domains of IoT. 10

Figure 2.2. Types of IoT systems. 12

Figure 2.3. Examples of use cases enabled by time-sensitive IoT (adopted from [69]). 13

Figure 2.4. An example graph for the age of information at a receiver. 16

Figure 2.5. An example of system architecture for IIoT [104], © 2020 IEEE. . 22

Figure 2.6. An example of multi-hop network [42], © 2017 IEEE. 30

Figure 2.7. An example of UAV data collection model. 37

Figure 2.8. An example of AUV-assisted data collection model [137], © 2021 IEEE. 40

Figure 2.9. An IoT system with three IoT devices and two serverless functions [164]. 48

Figure 2.10. An example of status sampling and processing procedure [167], © 2021 IEEE. 49

Figure 3.1. A claw. 59

Figure 3.2. User pairing conflict graph for a 4-user NOMA scheme. 60

Figure 3.3. Average AoI comparison of pairing strategies ($N = 10$). 62

Figure 3.4. Average AoI comparison of pairing strategies ($\gamma = 3.6$). 63

Figure 4.1. Average number of sensors exceeding their PAoI limits. 74

Figure 4.2. Average sum of AOIs exceeding PAoI limits. 75

Figure 4.3. Average AOIs. 75

Figure 5.1. Upper bounds for normalized average AoIs. 87

Figure 6.1. Multicast network. 96

Figure 6.2. Transmissions without (left) and with (right) network coding. . . . 100

Figure 6.3. Probability of perfect recovery with respect to q and n. 108

Figure 6.4. AoI measurements for a UAV IoT network ($n = 30$). 115

Figure 6.5. AoI measurements for a military IoT network ($n = 30$). 117

Figure 6.6. AoI measurements for a household IoT network ($n = 30$, $\frac{n}{k} \geq 1$). . 118

Figure 6.7. AoI measurements for a household IoT network ($n = 30$, $\frac{n}{k} \leq 1$). . 119

Figure 6.8. AoI measurements for all scenarios ($n = 5$) where subfigure 6.8(a) shows average AoI values and subfigure 6.8(b) shows peak AoI values. 121

Figure 6.9. AoI measurements for all scenarios ($k = 5$) where subfigure 6.9(a) shows average AoI values and subfigure 6.9(b) shows peak AoI values. 122

Figure 7.1. Multi-hop, multi-source network model. 125

Figure 7.2. Number of servers vs average AoI. 135

Figure 7.3. Number of sources vs $\frac{\Delta_{NC}}{\Delta_{UC}}$. 136

Figure 7.4. Number of servers vs $\frac{\Delta_{NC}}{\Delta_{UC}}$. 136

Figure 8.1. Illustration of energy walk. 141

Figure 8.2. σ_f values with respect to harvested amount h and harvesting efficiency ω. 142

Figure 8.3. AoI measurements for a UAV IoT network ($n = 30$, $k = 5$). 144

Figure 8.4. AoI measurements for a military IoT network ($n = 30$, $k = 5$). . . 145

Figure 8.5. AoI measurements for a household IoT network ($n = 30$, $k = 5$). . 146

Figure 8.6. AoI measurements ($\omega \times h = 1$, $n = 30$, $k = 5$). 147

LIST OF TABLES

Table 2.1. Comparison of learning-based AoI optimization works. 33

Table 2.2. Methods to optimize UAVs trajectory. 39

Table 6.1. Simulation parameters of IoT networks. 113

LIST OF SYMBOLS

a_k	Power allocation coefficient for user k
C_{comp}	Computational power of sensor nodes
C_X	Channel capacity of X
d_k	Distance of user k to base station
E	Encoded message
f_i	Encoding coefficient
$\mathbb{E}[.]$	Expectation operator
$f(.)$	Probability density function
$F(.)$	Cumulative distribution function
h_k	Channel coefficient of user k
k	Number of receivers
l	Number of bits for the packets at each information source
M	Information message
n	Number of packets a status update is divided to
N	Number of users or sensors
N_b	Block-length of a code
N_r	Number of monitors
N_s	Number of servers
N_t	Number of sources
p_{ij}	Transmission success probability from the i-th source to the j-th server
q_{jk}	Transmission success probability from the j-th server to the k-th monitor
P_e	Block-error probability
R	Transmission rate
s_j	Number of packets at the j-th server
t_τ	Transmission times for uncoded strategy
t_τ^{NC}	Transmission times for network coded strategy

T	Number of iterations in Monte Carlo simulations
W	Number of frequency bands
$W(h)$	Whittle's Index in state h
Δ	Age of information
γ	Path-loss exponent
λ	Inverse of the expected waiting time at the buffer of the receiver nodes
σ_f	Expected number of energy failures per transmission trial
ψ_B	Energy level of B's battery

LIST OF ACRONYMS/ABBREVIATIONS

ACK	Acknowledgement
ADRA	Age-dependent Random Access
AF	Amplify and Forward
AmI	Ambient Intelligence
AoCI	Age of Changed Information
AoD	Age of Data
AoI	Age of Information
AoP	Age of Processing
AP	Access Point
ARQ	Automatic Repeat Request
AuD	Age upon Decisions
AUV	Autonomous Underwater Vehicle
AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BLE	Bluetooth Low Energy
BS	Base Station
BSC	Binary Symmetric Channel
CC	Chase Combining
CDF	Cumulative Distribution Function
CoI	Coage of Information
CoI	Information Cofreshness
CR-IoT	Cognitive Radio-based IoT
CSMA	Carrier Sense Multiple Access
D2D	Device-to-Device
DCAAs	Dynamic Channel Access Attacks
DF	Decode and Forward
DRDPG	Deep Recurrent Deterministic Policy Gradient
DRL	Deep Reinforcement Lerning

EH	Energy Harvesting
FCFS	First-Come-First-Served
FD	Full-Duplex
FIFO	First-In-First-Out
FPTAS	Fully Polynomial Time Approximation Scheme
GATS	Grant Assignment and Transmission Scheduling
GIoT	Green IoT
GPD	Generalized Pareto Distribution
HAP	Hybrid Access Point
HD	Half-Duplex
IFE-DGCS	Information Freshness-Guaranteed and Energy Efficient Data Generation Control System
IIoT	Industrial IoT
IoT	Internet of Things
IUDA	Information Update Delivery and Acquisition
JSCC	Joint Source Channel Coding
KKT	Karush-Kuhn-Tucker
LDPC	Low-Density Parity-Check Code
LGFS	Last-Generated First-Served
LOS	Line-Of-Sight
LP	Linear Programming
LTE	Long Term Evolution
LZW	Lempel-Ziv-Welch
M2M	Machine-to-Machine
MC	Markov Chain
MDP	Markov Decision Process
MEC	Mobile-Edge Computing
MIMO	Multiple-Input Multiple-Output
MLD	Maximum-Likelihood Decoding
MTDs	Machine-Type Devices
MWIS	Maximum Weighted Independent Set

NACK	Negative Acknowledgement
NBU	New-Better-Than-Used
NC	Network Coding
NE	Nash Equilibrium
NOMA	Non-Orthogonal Multiple Access
OLTD	Once-Lost-Then-Drop
OMA	Orthogonal Multiple Access
OPG	Ordinary Potential Game
PAoI	Peak Age of Information
PDF	Probability Density Function
PPP	Poisson Point Process
PSO	Particle Swarm Optimization
PtS	Preprocess-Then-Sense
PwS	Preprocess-While-Sensing
QoS	Quality of Service
RMAB	Restless Multi-Armed Bandits
RPT	Reliable Packet Transmission
S-IoT	Satellite-Based IoT
SC	Successive Cancellation
SIC	Successive Interference Cancellation
STI	Spatially Temporally Correlated Mutual Information
TARQ	Truncated Automatic Repeat Request
TD3	Twin-Delayed Deep Deterministic
UAVs	Unmanned Aerial Vehicles
UE	User Equipment
ULTD	Uplink-Lost-Then-Drop
VNF	Virtual Network Function
WET	Wireless Energy Transfer
WN	Wireless Node
WNCS	Wireless Networked Control System

1. INTRODUCTION

In the information age, 5G and beyond communications are anticipated to support billions of data-generating devices such as mobile phones, cars, drones and industrial robots. According to the Ericsson Mobility Report published in November 2022 [1], the number of 5G subscriptions is expected to reach 5 billion by the end of 2028, accounting for 55 percent of all mobile subscriptions. With this unprecedented growth, time-sensitive applications such as real-time surveillance, monitoring machine conditions for predictive maintenance and autonomous driving as well as IoT-enabled devices such as industrial robots, fitness trackers, and household appliances will increase in importance and continuously gather new data. Thus, they rely greatly on the timely supply of information, i.e. the data freshness. A central unit in a factory, for instance, may use artificial intelligence to analyze the collected data from the scattered sensors and take immediate action to optimize the production line. Such time-sensitive optimization issues necessitate seamless device-to-device connections, i.e., accurate and up-to-date data. Due to the rise of new high-rate, resource-intensive applications such as virtual reality, augmented reality, and gaming, data freshness is necessary not just for IoT-enabled devices but also for mobile users.

AoI, a novel performance measure proposed in [2] to measure the timeliness and freshness of data packets in a network, has emerged as an important metric and attracted significant research interest. AoI at a receiver is defined as the elapsed time since the source created the last received information update packet [2]. AoI can be quantified as the elapsed time between the current timestamp and the creation time of the most recent received status update packet at the destination.

In wireless networks, non-orthogonal multiple access (NOMA) is a promising technology to realize the future radio access requirements envisioned for the next generation wireless networks [3]. Since more than one user can be supported in a single resource block in NOMA, it is possible to realize the potential leading to increased connectiv-

ity [4] and data rates [5] compared to orthogonal multiple access (OMA) technologies. Contemporary NOMA techniques rely on differentiating the signals in code domain, waveform domain or power domain, and detecting the superimposed user signals on each resource block by successive interference cancellation (SIC) [3].

The performance of wireless networks employing NOMA may be evaluated and optimized in terms of a number of metrics such as the sum-rate as done in [6–8], energy efficiency as in [9–11] and fairness among users as in [12,13]. A new performance metric named AoI, which was proposed in [2], has emerged as a key metric to measure the freshness of information in a network and started to gather significant research interest due to the need for timely arrival of status updates. AoI, at a receiver, is defined as the elapsed time since the last received information update packet was generated at the source [2]. Prior works such as [14] show that the optimization of the sustainability of timely updates is not equivalent to the maximization of the sum-rate or the minimization of the delay. In addition, note that making informed decisions, using precise location-based services, getting personalized recommendations, facilitating successful collaboration and guaranteeing mobile security and privacy all depend on having access to real-time and up-to-date information [15]. Mobile users may encounter inaccuracies, and compromised experiences in the absence of fresh data, underlining the critical role that timely and appropriate data plays in improving the mobile user experience. Therefore, AoI optimization/minimization in wireless networks has become a significant problem and has been addressed as in [16, 17] in different contexts with various constraints and in NOMA systems as seen in [18–20].

The first optimization problem is on the pairing of users in a NOMA network with an aim of minimizing the average AoI.

Moreover, customizing sensor priorities to meet specific demands can be critical in a large number of applications. For instance, in healthcare systems, vital sign sensors hold greater significance compared to environmental sensors [21], while in transportation, real-time tracking sensors are prioritized [22]. In unmanned aerial vehicle

(UAV) assisted wireless sensor networks, a priority-based data gathering framework is introduced in [23]. Furthermore, a priority-aware protocol is proposed in [24] for mixed-criticality industrial wireless sensor networks. As emphasized in [22], AoI (or PAoI) plays a significant role to capture the sensor prioritization. For this reason, sensor-based PAoI limits are generally employed to prioritize more vital sensors as also done in this work.

The success of AoI optimization is primarily dependent on the sampling and scheduling strategies. In this thesis, we focus on obtaining a feasible scheduling method in a resource-constrained environment in order to minimize the number of sensors violating their PAoI limits and the sum of the violation amount within the network. [25] and [26] are the examples of works which focus on minimization of AoI violation probabilities.

The second optimization problem is on the scheduling of sensors in an IoT network with an aim of properly addressing the sensor priorities.

In the realm of channel coding, the reliable communication in diverse environments is a crucial factor for the efficient operation of the wireless communication systems. Similar to other networks, in IoT systems reliable and efficient communication is accomplished by the use of channel coding techniques as exemplified in [27–30]. For instance, [27] employs a joint source channel coding (JSCC) scheme using DP-LDPC codes for wireless image communication in IoT, optimizing finite-length DP-LDPC codes and rate allocation for improved reliability and efficiency. Similarly, [28] proposes a joint learning and channel coding scheme with LDPC, reducing power dissipation in ML-based IoT systems through remote model retraining. In [29], an information-centric approach based on JSCC via type-based multiple access is explored, highlighting the advantages of decentralized detection in the presence of stringent fronthaul constraints. Additionally, [30] investigates channel coding at low capacity, examining fundamental limits and code constructions tailored for IoT networks.

Data freshness and reliability present a fundamental trade-off, wherein the choice of channel coding impacts system latency. More complex coding schemes may introduce processing delays, leading to higher latency and increased AoI. Hence, analyzing AoI within the context of channel coding is crucial to strike a balance between data freshness and reliability. Prior research, such as [31] and [32], has investigated the relationship between channel coding and AoI. For instance, [31] proposes a two-layer coding strategy and formulates an AoI-optimal redundancy-allocation problem, while [32] explores average AoI performance for polar coded status updates. However, a comprehensive comparison between theoretical AoI limits achievable through block coding and practical realizations remains lacking in the literature.

The third problem is on analyzing the effect of block coding, polar coding as a special case, on AoI in an IoT network.

Furthermore, network coding was introduced by Ahlswede *et al.* in [33] as one of the potential solutions to meet growing network demands. In networks which do not employ network coding, network resources such as intermediate nodes and routers can only independently process and forward data packets. With the advent of network coding, intermediate nodes in a network are allowed to combine multiple packets to create and forward new ones. The benefits are increased throughput and network robustness, as demonstrated in a number of works, including [33–37]. Such advantages incur an increase in the computational complexity of network nodes. At the heart of this chapter's discussion is the trade-off between such a computational burden and the AoI gain resulting from the need for fewer transmissions over unreliable channels.

The AoI performance of network-coded flows is investigated in [38–41]. In [38], AoI for two-user broadcast networks with feedback deployed is considered and lower and upper bounds on the expected weighted aggregate AoI of the users are determined. Explicit conditions are found under which coded randomized and deterministic policies outperform their uncoded counterparts for symmetric dependent channels. In [39], the timeliness of delivering a file with a specified number of packets over a multicast

wireless network employing network coding is evaluated, and the timeliness is quantified by AoI. On average and peak AoI, the effects of network characteristics such as the number of packets in a coding block, the number of destinations, and the magnitude of Galois field used for packet encoding are explored. It is also indicated that optimizing AoI is not equivalent to minimizing delay under the assumption that coded packets are obtained at random. In a separate work [40], the authors consider a wireless communication network with an adaptive system for determining the number of packets to be encoded for each transmission, and they characterize the information timeliness by providing closed form expressions for the average and peak AoI as functions of parameters of interest. In the same study, it is demonstrated that significant AoI performance gains can be attained over uncoded transmissions, and that these gains are robust to changes in network parameter values. In [41], the difference between once-lost-then-drop (OLTD), reliable packet transmission (RPT) and uplink-lost-then-drop (ULTD) protocol is examined in a two-way relay network, where computational complexity of network coding is disregarded, and ULTD is reported to outperform others significantly on software-defined radio.

The fourth problem emphasizes the benefits of network coding on the AoI of various IoT networks despite the computational burden that comes along with network coding.

In various IoT models such as cellular IoT, UAV-assisted IoT, and dense IoT, intermediate nodes are required for information to reach from the source to the destination. In these so-called multi-hop IoT networks, data does not travel directly between the source and destination, which, in turn, makes it necessary to optimize the AoI within a multi-hop deployment scenario. For this reason, this problem, namely AoI optimization for multi-hop IoT systems, has been considered in a number of previous works such as [42–46]. In [42], AoI minimization in multi-hop IoT networks is considered with interference constraints for multiple source-destination pairs and stationary policies based on link activation probabilities are proposed. Interference-free multi-hop networks are explored in [43], analyzing the impact of preemptive last-generated

first-served (LGFS) policy on the AoI and its optimality. In the context of satellite-based networks, the age-optimal redundancy allocation problem is considered in [44] using hybrid automatic repeat request (HARQ) and incremental redundancy HARQ (IR-HARQ) transmission schemes. For single-hop IoT networks, [45] presents a network code HARQ (NC-HARQ) protocol that outperforms the existing HARQ schemes in terms of average AoI. Additionally, different ARQ schemes are investigated in [46], including truncated hybrid ARQ with chase combining (HARQ-CC), revealing insights into their average AoI and energy efficiency performance under varying network conditions.

Contemporary IoT systems commonly generate singular status updates based on the data from a single sensor. However, practical IoT networks frequently necessitate multiple data inputs from various sensors, leading to the existence of correlated devices or diverse information sources within the system [47]. In scenarios where a status update is contingent upon the successful reception of data from all sources, it becomes imperative to assess the AoI independently for each individual physical observation. In [48], a discrete-time queueing model is considered in multi-source IoT systems. In [49], both AoI-aware and AoI-reduction-aware multi-source information update problems are investigated.

The fifth problem emphasizes the benefits of network coding on AoI in multi-source and multi-hop network scenario.

Moreover, the energy requirements of sensors used in IoT networks are vital in terms of sustainability and efficiency. In this context, the concept of energy harvesting has emerged as a promising solution for powering IoT devices, offering the potential to alleviate dependence on traditional power sources and enable sustainable operation in diverse environments. This paradigm shift towards energy-autonomous IoT systems has spurred significant research efforts aimed at optimizing energy utilization while ensuring timely and reliable data transmission.

In recent years, the integration of energy harvesting capabilities with IoT networks has given rise to novel challenges and opportunities, particularly concerning the management of information freshness and energy efficiency. For instance, an information freshness-guaranteed and energy-efficient data generation control system tailored for energy harvesting IoT deployments is proposed in [50]. A dynamic age minimization approach, introduced in [51], coupled with real-time information preprocessing to enhance the performance of edge-assisted IoT devices equipped with energy harvesting capabilities. The tradeoff between timeliness and distortion in energy harvesting-powered sensing systems within IoT contexts is analysed in [52]. Additionally, age-aware status update control strategies leveraging reinforcement learning techniques to optimize energy harvesting IoT sensor operations are explored in [53]. In [54], the average age of information of cached status updates for processes monitored by energy harvesting sensors is investigated. Moreover, the main point is minimizing the average on-demand age of information in IoT networks featuring energy harvesting sensors in [55]. These works collectively underscore the significance of energy harvesting in shaping the evolution of IoT networks while highlighting the intricate interplay between energy efficiency and data freshness.

The sixth and the final problem analyzes the impact of network coding on AoI in an energy harvesting environment.

In short, this thesis focuses on optimizing AoI within wireless networks to improve the timeliness and reliability of information delivery and begins with a comprehensive review of existing literature on AoI. The main contributions can be listed as below:

- A comprehensive survey of contemporary works in the field of AoI in IoT networks covering the factors influencing network performance, including energy constraints, network architectures, packet formatting, pre-processing and scheduling strategies.
- Optimization of dynamic user pairing in downlink NOMA systems, focusing on minimizing AoI while maximizing network efficiency.

- Introduction of a novel scheduling strategy tailored for priority-aware IoT networks, aimed at optimizing AoI by prioritizing data packets based on their urgency.
- Exploration of the impact of block coding on AoI in centralized IoT systems, utilizing insights from BEC and BSC models to evaluate the effectiveness of block coding techniques.
- Investigation into the role of network coding in improving information freshness and reliability across diverse IoT networks, including its impact on AoI in multi-source multi-hop IoT networks, providing insights into optimizing AoI in complex network scenarios and energy harvesting environments.

1.1. Thesis Organization

The rest of this thesis is organized as follows. A detailed literature review on the state-of-the-art research on related subjects is provided in Chapter 2. The AoI minimizing dynamic user pairing optimization problem is discussed and solved in Chapter 3. A novel scheduling strategy is proposed in priority-aware IoT networks for AoI optimization in Chapter 4. The effect of block coding on AoI is investigated in a centralized IoT network in Chapter 5. The impact of network coding on AoI for diverse IoT networks is investigated in detail in Chapter 6. Chapter 7 provides an extended work for the impact of network coding on AoI by considering multi-source multi-hop IoT networks. Chapter 8 provides additional analysis of network coding on AoI for the case of energy harvesting environment. Finally, we conclude the thesis in Chapter 9.

2. LITERATURE REVIEW

The Internet of things (IoT), which is the connection of physical items to the internet via sensors and wireless links, delivers life-improving solutions and improves our quality of life. It is estimated that by 2024 there will be 83 billion internet-connected IoT devices, up from 35 billion in 2020 [56]. As shown in Fig. 2.1, currently the primary application areas of IoT can be identified as agriculture, health, transportation, industry, energy, smart home, smart city, and wearable objects but new application domains are also expected to emerge in the near future. Improvements in the sensor technology, energy harvesting device efficiencies, advancements in wireless energy transfer (WET), and research on low-power wireless communication all have a significant impact on this anticipated widespread adoption. In addition, monitoring the status of items, analyzing the data obtained from sensors, and developing optimal systems enhance the evergrowing demand for IoT applications.

In many wireless communication standards such as 4G and 5G, throughput, outage probability, bit error rate, packet error rate and spectral efficiency are the key performance indicators. In contrast, freshness of information is a more important performance metric the context of IoT applications due to the presence of multitude of devices most of which are required to transmit at a fixed rate or at variable but relatively lower rates than most conventional communication applications. In this regard, age of information (AoI) has been proposed in [2] as a new performance measure commonly used in the literature to assess the freshness of information. AoI is defined as the elapsed time since the source node generated the most recent received status update and is quantified as the time difference between the current timestamp and the generated timestamp of the most recent received status update.

In time-sensitive applications such as remote surgery, for instance, the information displayed on the monitor from sensors such as temperature, blood pressure and heart rhythm must be extremely fresh and accurate. In order to avoid any risk on the

human life, medical-IoT applications should be guaranteed to have real-time haptic feedback [57]. Furthermore, in time-critical IoT networks, AoI may be constrained to ensure quality of service (QoS). For instance, in order to detect the forest fires before they spread, it may be essential to construct an IoT network with temperature-measuring devices which are randomly dispersed around the landscape [58]. Unmanned aerial vehicles (UAVs) can be used to collect and transmit data produced by these sensors. For a destination node, to identify a fire before it grows out of control, the temperature data supplied from sensors to the destination must be fresh. Therefore, the network should be constructed to maintain the AoI below a certain level, taking into account elements such as the number of sensors, number of UAVs, UAV flight path, etc. In numerous IoT applications, the sensors are battery-free since they harvest the energy required to operate [59]. Energy constraint makes it harder for sensors to frequently collect data and therefore making information freshness essential once again. In addition to the energy constraints other parameters such as the scheduling strategy, sampling frequency of the sensor, periodic and non-periodic packet sizes, pre-processing effects and the number of network hops also affect the AoI performance.

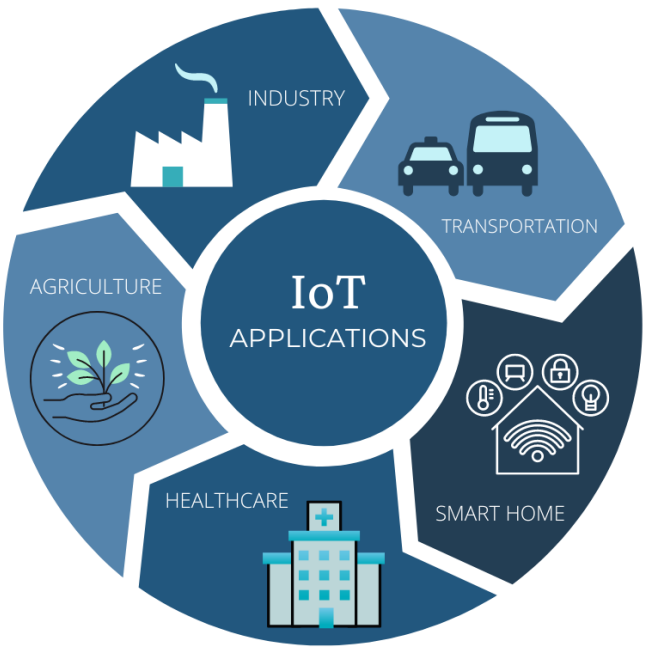


Figure 2.1. Application domains of IoT.

Observing the growing trend in the both the number and diversity of IoT applications in next generation wireless world and the potential role of AoI as a relevant design and optimization metric in them, this thesis includes a comprehensive overview of contemporary research works on AoI in IoT networks.There have been a number of surveys on the application of AoI such as those given in [15] [60], [61]. [15], being the first ever survey on AoI presents an overview for AoI in low-latency cyber-physical systems where the emphasis is more on the use of different AoI metrics and less on specific applications such as IoT. Moreover, only works published up to and including 2020 have been considered whereas most of the advances especially in the IoT domain has taken place since then. AoI in cellular IoT has been reviewed in [60] where the challenges of achieving real-time updates over wireless fading channels are explored and a survey of optimization methods is presented with a novel prediction-based scheme. A recent survey of AoI in IoT networks is given in [61] where the main focus has been the large-scale networks specifically in the context of ambient intelligence (AmI) and IoT systems. Among many factors which have an impact of AoI, the emphasis has mostly been on the queuing policies, scheduling and multiple access strategies. Within this background, this thesis aims to provide an overview of contemporary works in this important and growing field within a broader view on not only the factors affecting the network performance such as the energy constraints, network architectures (single vs. multiple source, single vs. multiple hops, etc.), packet formatting, pre-processing, scheduling strategies, but also the range of IoT applications where AoI may be applicable. We also present optimization methods and design strategies for various IoT network structures which include both the traditional optimization formulations and solutions and also the design approaches based on machine learning techniques.

2.1. Internet of Things

IoT is a system in which physical objects can collect data using smart sensors and transmit the gathered information to another device or the cloud via a wireless network. Identification, sensing, communication, computation, services and semantics are the six fundamental elements for IoT functionality as described in [62]. Even though

there are numerous IoT applications, here we focus on the most commonly used ones in the literature: satellite-based IoT (S-IoT), industrial IoT (IIoT), cellular IoT, UAV-assisted IoT, green IoT (GIoT), ultra-dense IoT, massive IoT, cognitive radio-based IoT (CR-IoT) which are illustrated in Fig. 2.2.



Figure 2.2. Types of IoT systems.

In addition to the previously mentioned IoT network types, the importance of the time-sensitive IoT networks, which caters to the data delivery needs of individuals, businesses and public institutions within a specified time duration with the required reliability, should be emphasized. Majority of the use cases, which include but not limited to industrial control for manufacturing jet engines, remote control in mining, remote bus driving, augmented reality, and smart harbor, etc., can be classified as shown in Fig. 2.3. In addition, numerous real-world examples may well be provided such as i) multimedia IoT networks collecting data from sensors located at traffic lights, vehicles, and traffic cameras in order to perform vehicle accident prediction in near-real time (e.g., within 30 ms) [63, 64], ii) distributed measurement systems for time, mission, or safety-critical data traffic for Industry 4.0 [24, 65], iii) web front-ends of

distributed systems which have quite common temporal requirements across different domains [66], and iv) healthcare event predictions and warnings for timely intervention [21,67]. According to [68], the current Wi-Fi equipment cannot meet the stringent latency and reliability requirements of time-sensitive Industry 4.0 applications. Consequently, next-generation Wi-Fi technologies and how they can be leveraged to enable various time-sensitive Industry 4.0 use cases, including wireless industrial automation control, remote rendering in extended reality applications, and cooperative simultaneous localization and mapping using autonomous mobile robots in a manufacturing plant, have been presented in the same work. The significance of time-sensitive IoT applications cannot be overstated.

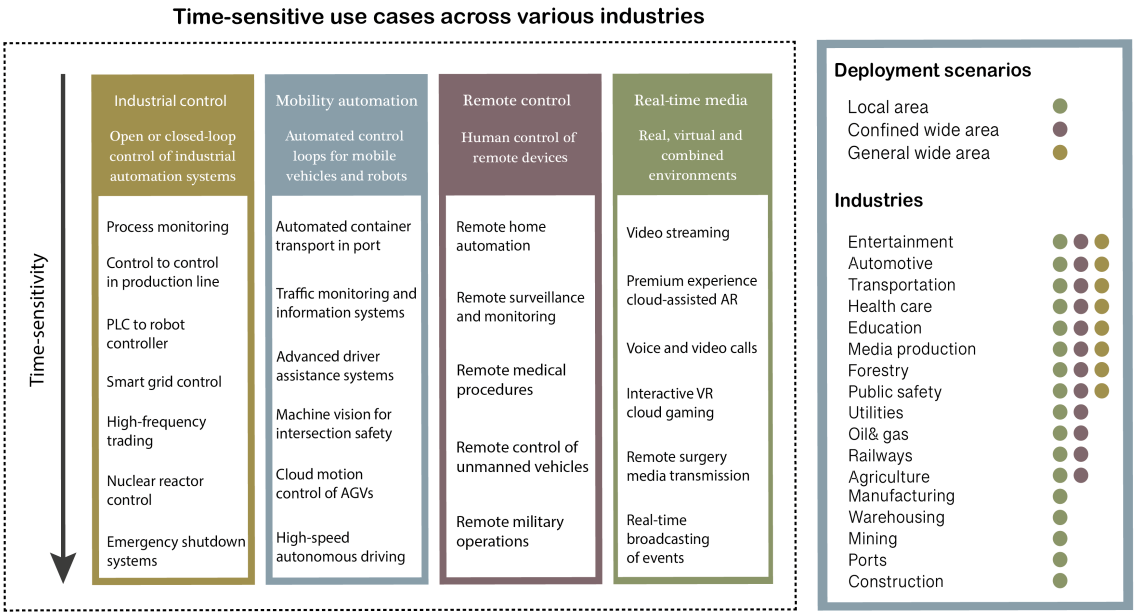


Figure 2.3. Examples of use cases enabled by time-sensitive IoT (adopted from [69]).

In some IoT applications, the sensors are dispersed throughout a vast geographical area that can encompass forests, oceans and even aircrafts. In such systems, a network with a broader coverage area is required as opposed to local networks. Owing to their pervasive coverage, satellites could be a viable option from this perspective. In order to connect the sensors of a flying item to the Internet, for example, we must construct a S-IoT network, which is commonly referenced in literature such as [45]. On the

other hand, local IoT applications can be integrated into S-IoT networks. The S-IoT network is comprised of machine-type devices (MTDs), satellites, an IoT gateway, and a control panel according to [56]. MTDs transmit sensor data to a nearby IoT satellite. In some cases, a group of MTDs send their data to a local IoT gateway that is in communication with the satellite. After satellites transmit data gathered from MTDs to a central satellite, the central one collects and transmits data to the control panel. Moreover, when information is to be transferred from the control panel to MTDs, data is sent from the control panel to the central satellite, from the central one to others, and from satellites to MTDs directly via the IoT gateway.

The evolution of the industry has necessitated monitoring the condition of machines and their components and optimizing the production line in order to maximize productivity. Changes in temperature, humidity, pressure, location, or vibration, for instance, may have a direct impact on the production line, making sensor-based machine management essential. In this sense, IIoT is frequently used to differentiate such applications from general IoT and this concept is considered in a large number of IoT applications such as the one given in [70]. In [71], a three-tier IIoT system architecture is shown, consisting of an edge, platform, and enterprise tier. The edge tier collects data from edge nodes and transfers it to the edge gateway. Edge gateways transmit data to the service platform, which manages data analysis and operations. The third step, the enterprise layer, provides end-user interfaces.

The fact that end devices are energy-constrained and there may be no direct LOS between the IoT device and the IoT gateway is one of the greatest obstacles in IoT networks. In many IoT applications, sensors are battery-free and feature energy-harvesting tags. In such situations, one or more relays are utilized to improve communication between the end device and the IoT gateway. Relays are advantageous not just due to the energy constraints of the end devices, but also to minimize BS overload. In a cellular IoT network, for example, clustered IoT end devices transfer data to a relay, which forwards it to the IoT gateway. The authors of [72] categorized relays according to their functionalities: amplify and forward, decode and forward, and compress and forward.

The IoT device's signal is amplified and sent through an AF-type relay. In a DF-type relay, the received signal is decoded before the message is re-encoded and transmitted. A CF-type relay compresses decoded messages, then re-encodes and sends the compressed messages. It should be noted that CF-type relays require a longer processing time than DF-type relays, which has a direct impact on the freshness of information provided by IoT devices. In [73] and [74], the freshness of information in relay-based IoT applications is evaluated. In addition, as the number of relays in a multi-hop IoT network increases, the freshness of the information is diminished as explained in [43]. In [42], the freshness of information in multi-hop wireless networks is optimized.

Relays can be fixed or mobile. For instance, gateways can be used as fixed relay in a cellular IoT network, whereas drones can be used as a mobile relay. In UAV-aided IoT networks, UAVs are usually employed as relays thanks to their high mobility. UAVs follow a path and collect data from sensors and transfer the collected data to a central monitor. From this perspective, data collection and trajectory optimization for UAVs are investigated to increase information freshness in some works such as [75].

GIoT, referring to the reduction of the greenhouse effect by decreasing energy consumption and boosting the efficiency of IoT applications, is investigated in [76]. Ultra-dense type IoT networks, which are considered in [77], are defined in [78] as the number of cells exceeding the number of active users and denseness is measured as more than 10^3 cells/km^2. Massive IoT, defined in [79] as a large number of interconnected items that produce machine-to-machine (M2M) communication, is investigated in works such as [80].

Cognitive radio technology enables devices to dynamically detect available radio frequencies in the surrounding environment and use them for communication. As the number of IoT devices continues to increase, it is becoming difficult and expensive to assign a spectrum band to each device. The authors of [81] state that spectrum sensing and spectrum sharing may be integrated into IoT networks using the CR-IoT system discussed in many studies such as [82].

2.2. Age of Information

AoI can be quantified as the elapsed time between the current timestamp and the creation time of the most recent received status update packet at the destination such that it may mathematically be stated as

$$\Delta(t) = t - u(t). \tag{2.1}$$

AoI model can be considered as shown in Fig. 2.4 where y-axis denotes the age of information and x-axis represents the time. When the generation of update packets is complete, the timestamp $u(t)$ is saved, where this first step corresponds to points A or C in Fig. 2.4. After generation, the source transmits the data through the channel. The destination node receives the transmitted signal. When the incoming packet is successfully received, the AoI decreases, which corresponds to points B or D. If the sent packet is lost or incorrectly decoded, then the AoI will continue to increase. Therefore, receipt itself does not affect AoI measurements.

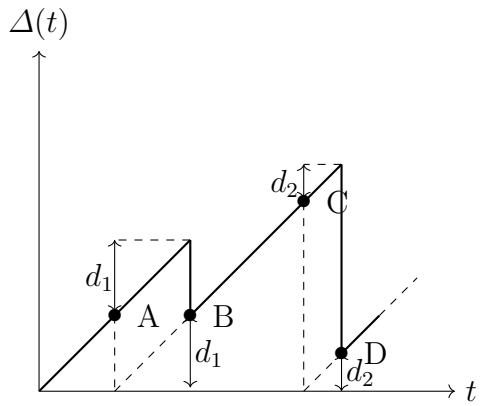


Figure 2.4. An example graph for the age of information at a receiver.

In the AoI literature, average and peak AoI are frequently used metrics. The average AoI measures the mean age of the entire system in a given time range, hence it accounts for the general freshness of packets circulating in the network. In works such as [83, 84], the time average of AoI are found. On the other hand, peak AoI stands for the maximum AoI in the system in a given time range, therefore it highlights the most obsolete data flow. Peak AoI is especially important for applications with strict data

freshness requirements such as real-time surveillance, autonomous driving and remote surgery. In [85–87], peak AoI related optimization problems are solved for different scenarios.

It has been shown in [14] that updating a destination about a remote system in a timely manner is not the same as maximizing the utilization of the communication system or ensuring that generated status updates are received with the least amount of delay. This is due to the fact that utilization can be maximized by having the source send updates as quickly as possible, which would result in the destination receiving delayed statuses due to the backlog of messages in the studied communication system. In this case, the stream of status updates can experience less delay if the rate of updates is decreased. Alternately, decreasing the update rate may result in the destination having status information that is unnecessarily out of date due to a lack of updates as mentioned in [88]. Consequently, the optimization/minimization of AoI in wireless networks has become an important issue.

AoI demonstrates notable proficiency to capture the data freshness in the realm of the real-time or time-sensitive communication systems, encompassing all of the potential use cases shown in Fig. 2.3. In these domains, the significance of promptly accessible information cannot be overstated, as it underpins the informed decision-making, the system performance optimization, and the provision of dependable services. Quantifying and refining the AoI enables these systems to achieve increased efficiency, reduced latency and enhanced overall performance.

Following its inception in 2011 [2], the AoI has garnered increased recognition after 2015 for a variety of reasons. Initially, the widespread deployment of IoT devices and the exponential growth of data generation across diverse domains engendered an escalated demand for real-time information. Industries encompassing transportation, healthcare, and smart grids necessitated the low-latency data in order to streamline operations and facilitate informed decision-making processes. Furthermore, the advent of 5G networking technologies [89] expedited and fortified the data transmission,

thereby enabling the rapid proliferation of various IoT systems in everyday life. Consequently, the real-time measurement and optimization of the AoI have gained even more prominence, particularly in the time-sensitive IoT applications.

The accuracy of output decisions, which determines the performance of IoT-enabled applications, is directly proportional to the freshness of the aggregated data measurements of the IoT devices at the destination nodes [90]. Different from [15] which considers recent works on AoI, IoT is the main setup that the idea of AoI came out. Therefore, we consider AoI in the context of not only for cellular IoT as in [60] but also for general IoT.

2.3. AoI in IoT

As discussed earlier, AoI is a commonly encountered and widely used performance metric in IoT networks, particularly in the time-sensitive applications. In this regard, while an overwhelming majority of the literature is dedicated to works involving AoI optimization, there also works dealing with AoI differently. For instance, some research works focus on the impact of system parameters on AoI which requires the theoretical analysis and derivation of closed-form expressions for it, while some others consider AoI as an additional constraint (instead of having as an objective function) within the system design. In this section we present an overview of works which fall into these two categories before presenting works dealing with AoI optimization issues in the next section.

2.3.1. Performance Analysis for AoI in IoT Networks

In the literature, several research works have focused on conducting theoretical analyses and deriving closed-form expressions for AoI. These investigations aim to showcase and quantify the influence of various system parameters on the freshness of data within IoT networks. In [91], the interaction between device-to-device (D2D) communications and real-time monitoring systems in a cellular IoT network is analyzed,

with the locations of IoT devices treated as a bipolar Poisson point process (PPP). The authors characterize the performance of D2D communications based on the average network throughput, whereas the performance of real-time applications is measured by the AoI, given that IoT devices employ a distance-proportional fractional power control scheme when sending status updates to their serving BSs. In this approach, IoT devices located beyond a specified distance from BSs cannot provide status updates to the serving BS. The work characterizes the spatial moments of the temporal mean AoI of a given status update link, as its transmission success rate is a function of the interference field observed by its receiver. Thus, [91] provides effective design principles for future huge IoT networks that jointly support D2D and cellular real-time applications.

In [92], a random access-based IoT network with periodic status updates is considered, in which IoT devices broadcast their updates to an access point utilizing state-dependent slotted Aloha access. An analytical approach is proposed for deriving the average and peak AoIs and system throughput and the ideal update interval for IoT devices are examined based on the minimization of the average AoI.

The uplink of an IoT network in which a number of sensors broadcast update packets to a BS has been examined in [93]. By establishing a discrete time Markov Chain (MC), the authors examine the effects of the packet arrival rate, number of sensors in the network, and queue length of each node in order to forecast the AoI levels and packet loss probabilities in the network. According to reported results, the AoI is proportional to the queue length, the number of nodes, and the system's arrival rate.

In [94], a dense IoT monitoring network is presented in which IoT devices broadcast updates to their respective receivers over noisy channels employing a carrier sense multiple access (CSMA) scheme. According to various feedback instances, such as perfect feedback, no feedback and different policies, such as entering an inactive state or continuing to transmit, closed-form expressions of the average AoI of devices are

provided. The authors of this paper demonstrate that preemption in service always reduces the average AoI of a device relative to when preemption is not present. To evaluate the asymptotic performance of a system with an infinite number of devices, a mean-field approximation approach is utilized, and the impacts of system characteristics on the average AoI are characterized.

The authors in [95] propose a mobile-edge computing (MEC)-assisted IoT system in which UAVs serve as MEC servers. Using a MC, performance of data collection in terms of packet loss rate and data amount is evaluated. The computation frequency of a UAV is designed to account for energy and AoI constraints where IoT devices adopt first-come–first-served (FCFS) principle and M/M/1 queuing.

2.3.2. Design Issues Under AoI Constraints

In time-sensitive IoT applications, AoI may not always be considered as the primary objective function of a minimization problem but instead used as an upper-bound constraint to ensure data freshness. This subsection is devoted to the review of works which falls under this category. To start with, for example, the optimal arrival rate is analyzed under average and peak AoI constraints in [96] where the optimal arrival rate is demonstrated to converge to the half of the service rate.

To prevent congestion and delay in IoT networks, edge caching is an effective way to store data close to nodes in case of a requirement. However, the cached data must be kept fresh so that outdated data is not transmitted to network devices. A model-driven cache management system that is implemented by a cache-enabled edge proxy to reduce the amount of exchanged messages and meet AoI criteria is proposed in [97] to retain the freshness of cached data without causing congestion due to updates.

The objective of [98] is to minimize the time average cost of sampling and transmitting status updates over wireless channels with average AoI constraints. Solving a Lyapunov optimization problem, the authors provide an optimal method in terms of

time average cost, which is proven by simulations, for deciding whether to sample a new packet, transmit the old packet, or keep silent.

An information freshness-guaranteed and energy-efficient data generation control system (IFE-DGCS) is designed in [50]. During the design phase, the probability of energy outages is minimized while the AoI is maintained below a specified threshold. The optimization problem is formulated by a constrained Markov decision process (MDP) and the optimal stochastic policy is derived by linear programming (LP). The comparison to typical energy-oriented systems reveals that a comparable energy outage probability behavior is achieved while ensuring a significantly reduced AoI in the network. Similar to [50], the authors of [99] investigate an IoT network aimed for joint optimization of AoI and energy efficiency, in which sensors transmit information updates to a central node situated on a mobile platform in an agricultural field. Using a modified nearest neighbor method and a data clustering strategy, an ideal trajectory for the mobile platform is proposed in order to improve the overall energy efficiency while maintaining the target AoI level and outage probability.

In [100] a scenario in which wireless sensors continuously collect and send environmental data to a centralized controller node is examined. As a goal, the mean, variance, and higher-order statistics of the network's energy consumption are minimized, subject to constraints on the probability of AoI threshold violations. Using results from the extreme value theory, the extreme AoI staleness is characterized and a distributed power allocation strategy is proposed.

Similar to [96], in [101], the violation likelihoods of AoI and peak AoI in IoT systems are described when a sensor transmits updates to a monitor assuming M/M/1 and M/D/1 queues with a first-come–first-served policy. In the same work, closed-form expressions are provided for peak AoI distribution in the network and violation probability for any AoI constraint. In addition, the optimal generation rate of the status update that leads to the lowest probability of violation is given.

The authors in [102] design and analyze an age-oriented random access for massive IoT networks by developing a new age-dependent random access (ADRA) protocol with a stationary threshold. In this protocol, each IoT device can only access the channel with a specific probability if its instantaneous AoI exceeds a pre-defined threshold. The authors then examine the proposed protocol's average AoI. Another random access scheme in which AoI has been considered can be found in [103]. The authors evaluate the average AoI of an uplink IoT monitoring network by considering a SIC-aided age-independent random access (AIRA-SIC) scheme, where the access point performs SIC with the intention of recovering collisions of multiple packets that are simultaneously transmitted by different devices in a slotted ALOHA manner. Based on the AoI levels of the devices, the authors demonstrate through simulations that the proposed AIRA-SIC outperforms the ADRA presented in [102].

In [104], AoI is considered in industrial IoT systems owing to its utility in the control and communication co-design to improve control performance in the event of a communication interruption which is illustrated in Fig. 2.5. The authors create a control cost metric and consider average AoI under the assumption that packet loss occurs in a network with finite retransmission durations. The authors also establish that the optimal joint horizon cost is bounded by a linear function of AoI by merging energy consumption and control cost into a joint cost.

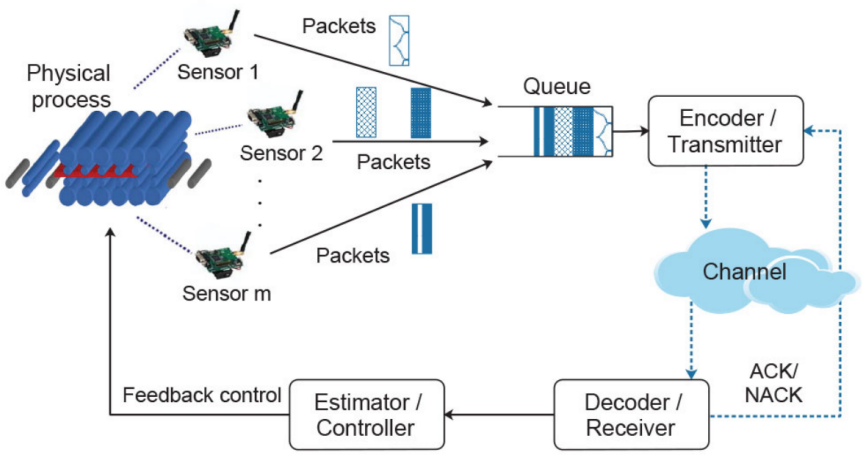


Figure 2.5. An example of system architecture for IIoT [104], © 2020 IEEE.

Joint optimization of the average cost of sampling and status update transmission subject to average AoI is considered in [105], where transmission errors may occur and a scheduling policy must determine whether a user must sample a new packet or resample in order to repeat a failed transmission. One of the analyzed scheduling strategies is dynamic, while the other two are stationary and random. Simulations demonstrate the significance of broadcasting an older packet in order to reduce the total average cost.

Unmanned surface vehicles are considered in [106] in the content of coordinated monitoring of the marine environment. AoI is used to characterize the effects of packet loss and transmission delay on the state estimate error of the maritime IoT system, as transmission performance such as packet loss and sensor data freshness play an essential role in the estimation of the system. To alleviate the impact of limited resources and path loss in the examined case, a mother ship-assisted cooperative transmission system is utilized. Then the position of the mothership, the channel allocation, and the transmit power are jointly optimized.

2.4. AoI Optimization in IoT

Optimizing AoI in IoT networks is critical because it affects the freshness and reliability of data. Minimizing AoI ensures up-to-date information for real-time applications, making IoT networks more responsive and effective. Therefore, the majority of the works in AoI literature is dedicated to the AoI optimization problems. In this section, we try to review works on AoI optimization where the different use case scenarios are formulated within a constrained optimization framework. In this regard, the initial focus is given to the AoI-energy tradeoff, followed by an exploration of the complexities involved when multiple information sources contribute to a single status update. Optimization in the context of correlated data is then discussed, along with considerations for multi-hop and relaying networks. Lastly, the impact of pre-processing on AoI is evaluated. Notice that AoI optimization is a wide area of research spanning a large body of applications and involving many different metrics and solution methodologies.

An exciting and growing line of work is the solution of large scale AoI optimization problems using machine learning/deep learning methods. It is also worth mentioning that scheduling and sampling strategies used in the IoT networks have a huge impact on any AoI optimization routine. For clarity of presentation we omit these two important areas in this section and refer the reader to the two subsequent sections, namely sections 2.5 and 2.6, for the reviews of learning-based methods and scheduling/sampling issues for AoI optimization, respectively.

2.4.1. Joint Optimization of AoI and Energy Efficiency

AoI-energy tradeoff constitutes a key challenge in contemporary IoT networks, which represents a balancing act of maintaining minimal AoI while simultaneously regulating the energy consumption for sustainable operation. Especially in terrestrial IoT and S-IoT networks, incentive schemes and retransmission mechanisms are investigated with various optimization approaches such as particle swarm optimization and dynamic power control strategy.

In [107], a multi-node wireless powered communication network is considered in which IoT devices harvest energy from various selfish wireless nodes (WNs) and then use this energy to transmit real-time status updates to a hybrid access point (HAP). To combat the WNs' selfishness, energy-incentive scheme and price-incentive plan are proposed. AoI-energy utility-based optimization problem is established and closed-form expressions are derived to find the optimal solution for the first problem. Then, a Stackelberg game is built to maximize the AoI-price utility function and semi-closed-form expressions are derived to determine the optimal solution for the latter task. Superiority of the proposed systems are demonstrated numerically comparing to the benchmark scheme in terms of network utility. Furthermore, the effect of the distance between WNs and IoT devices, as well as the distance between HAP and IoT devices are also investigated. In [108], an IoT monitoring system is investigated, in which an IoT device observes a physical process and delivers arbitrarily generated status updates to its associated access point (AP) via a wireless error-prone channel, which results in

packet loss for status updates. The authors employ the truncated automatic repeat request (TARQ) mechanism because it is presumed that there is no feedback from the AP to the IoT device. The TARQ system requires IoT devices to repeatedly transmit status updates until a new physical process is detected or the maximum transmission time limit is achieved. After deriving closed-form expressions of the average AoI, the average peak AoI, and the average energy consumption, the average AoI is minimized by jointly optimizing the transmit power of the IoT device and maximum allowable transmission times under a constraint on the average transmit power. Simulations reveal that the employed TARQ system outperforms the conventional ARQ scheme.

A NOMA-based S-IoT network is considered in [109] in which satellites send status updates to numerous user equipments (UEs). Minimization of AoI under the long-term and short-term constraints of power and throughput is formulated. To solve the optimization problem, the authors turn it into a series of online power allocation problems using Lyapunov optimization methods. The particle swarm optimization (PSO) algorithm is proposed to find the optimal solution with linear computational complexity. The proposed algorithm outperforms the benchmark schemes in terms of AoI performance.

In [110], the dynamic power control policy for IoT networks is investigated in order to minimize the average AoI in the presence of stochastic status update arrivals. The authors assume that BS knows the age of information of the status update only when source nodes successfully transmit the packet and that BS is unaware of instantaneous channel power gains. The AoI optimal dynamic power control problem is formulated as a partially observable Markov decision process (POMDP) with continuous action space. To reduce the weighted average AoI at the BS, the authors propose a deep recurrent deterministic policy gradient (DRDPG) based power control (DRPC) method, which outperforms the baseline approaches.

2.4.2. AoI Optimization for Multiple Sources

In order to generate a single status update, the majority of contemporary IoT systems consider the status data from a single sensor. Nonetheless, practical IoT networks may require several physical observations from many IoT sensors. This description refers to an IoT system that contains correlated devices or various information sources. Due to the fact that a status update is not generated until all multiple information has been successfully received, AoI should be evaluated for each physical observation independently in such scenarios. Moreover, multi-hop and multi-source network model is a model that defines data transmission in networks involving multiple sources and multiple hops. In this model, data packets are delivered to the destination by being carried by multiple intermediate nodes or routers.

The authors in [48] investigates discrete-time queueing model in multisource IoT systems with Bernoulli information packet arrivals and discrete phase-type service time distribution for all sources. For three distinct queueing disciplines, namely nonpreemptive bufferless, preemptive bufferless and nonpreemptive single buffer with replacement, the authors determine numerically the exact per-source distributions of AoI and peak AoI. The correctness of the proposed model is tested and the optimal selection of Bernoulli parameters for two IoT sources is analyzed using a numerical example.

The scenario in which multi-packet must be received successfully in order to generate a status update is considered in [111]. Learning-based strategies are examine in restless multi-armed bandits (RMAB) and a Whittle index-based scheduling policy is also presented in order to build an effective scheduling method. A new transmission policy based on erasure codes for lossy networks is proposed in order to enhance the performance of scheduling policies. In terms of AoI performance, the proposed method is found to be superior to existing baseline policies such as the naive Whittle index policy and the greedy policy regarding both lossless and lossy networks. A similar network architecture is also studied in [112], and the authors design two optimization problems: AoI-aware multi-source information update issue and AoIR-aware multi-

source information update problem. After establishing that the optimization problem is NP-hard, an efficient greedy algorithm with a guaranteed approximation ratio is proposed for the first case. For the latter case, a polynomial-time optimal maximum weight bipartite matching solution on an auxiliary graph is proposed. Simulation results of the superior performances of the proposed methods are presented.

Similarly, scheduling of IoT devices is investigated in [113] to jointly decrease the average AoI and total energy consumption in a scenario where multiple information packets are necessary for a single status update. The authors analyze two types of IoT devices: status updates of Type-I devices arrive randomly, while status updates of Type-II devices can be generated with a sampling cost upon BS's request. This problem is formulated as an infinite-horizon, average-cost MDP and show that, even if all devices are Type-II, the optimal scheduling policy is threshold-based with regard to the AoI. Simulations reveal that the proposed policy is more accurate than the myopic baseline policy.

2.4.3. AoI Optimization for Correlated Data

On the other hand, By extracting the correlated information from data, the energy consumption of IoT end devices can be decreased given certain AoI constraints, taking into account the massive data generated from thousands of IoT sensors. In [114], the authors propose a status updating policy that optimizes the sampling frequency of IoT devices and evaluate the effectiveness of the proposed policy using real data. The objective in [115] is to reduce the transmit power of sensors according to the peak AoI constraint and probabilistic constraint on queuing latency in industrial IoT networks. The authors formulate the tail behavior of the latency as a generalized Pareto distribution (GPD) in order to address the power allocation problem via Lyapunov optimization. A combination of local model selection and federated learning is proposed, in which each sensor utilizes its own data to train the GDP model. It is concluded that the correlation-aware method presented for selecting local models in federated learning outperforms the standard methods. In addition, the authors in [116] examine the

influence of the heterogeneous traffic pattern of IoT devices on AoI performance and energy efficiency in heterogeneous CR-IoT networks. After deriving the closed-form expressions for energy and AoI, an optimal transmit power optimization algorithm is proposed to maximize energy efficiency by utilizing diverse traffic patterns of IoT devices under a predefined AoI threshold constraint. Finally, benefits of the suggested approach are demonstrated numerically.

2.4.4. AoI Optimization in Multi-Hop Networks

There may be no direct link between the IoT end device and the destination node; therefore, some intermediate nodes may be utilized to route packets in multi-hop IoT networks. For this reason, AoI optimization under various constraints and rules is attracting attention in multi-hop IoT networks. In the literature, AoI optimization is considered under interference-free and general interference constraints. On the other hand, S-IoT has been particularly explored due to the significant propagation delays in multi-hop transmission. Moreover, various ARQ schemes are proposed to compare their efficiency for AoI performance.

Talak *et al.* investigate AoI minimization under general interference constraints with R source-destination pairs where there is a graph between the R source nodes and the R destination nodes [42]. Fig. 2.6 illustrates the corresponding age propagation over nodes with $R = \{r_1, r_2\}$ as a network snapshot. Simple stationary policies based on a stationary probability distribution for link activation are derived. AoI, which is a function of link activation rates, is demonstrated as a convex function for a single source-destination pair. Then the optimal policy is obtained. For an R source-destination pair, the optimal policy is obtained by considering all source-destination pairs as independent single source-destination pairs. Unlike [42], in [43] interference-free multi-hop IoT networks are considered for a single source-destination pair. When packet transmission times are exponentially distributed and preemption is allowed, the authors prove that a preemptive last-generated first-served (LGFS) policy improves AoI performance more than any other causal approach. However, the authors indi-

cate that preemptive LGFS policy may be far from optimal when modeling one of the new-better-than-used (NBU) distributions. In addition, the scenario that preemption is prohibited and packet transmission times are arbitrarily distributed is analyzed.

Satellite-based multi-hop IoT networks are constrained by nontrivial propagation delay due to the long communication distances. Therefore, traditional automatic repeat request (ARQ) strategies have low efficiency and it is important to decide whether retransmission is beneficial for AoI or not.

In [44], both simple hybrid-ARQ (HARQ) and incremental redundancy HARQ (IR-HARQ) transmission schemes are adopted and the age-optimal redundancy allocation problems under the reliability constraint are investigated. The authors first derive explicit expressions of the optimal codeword, then find the threshold of the propagation delay for beneficial retransmissions.

On the other hand, a network code HARQ (NC-HARQ) transmission protocol is proposed for S-IoT in [45]. The authors derive the closed form expressions for average AoI of the proposed NC-HARQ transmission protocol by establishing a four states MC. Finally, it is shown by simulation that the NC-HARQ protocol outperforms several state-of-the-art HARQ schemes in terms of average AoI.

Moreover, non-ARQ, classical ARQ, truncated ARQ, and truncated hybrid ARQ with chase combining (HARQ-CC) schemes are investigated in [46] regarding the average AoI and energy consumption efficiency under fixed redundancy constraint. The authors derive the closed form expression for the average AoI and the average energy cost of each considered scheme by using the renewal processes theory. ARQ schemes are compared in terms of average AoI and average energy cost under different network parameters such as code length and SNR. The truncated HARQ-CC is shown to achieve the best average AoI and the moderate average energy cost compared to other schemes.

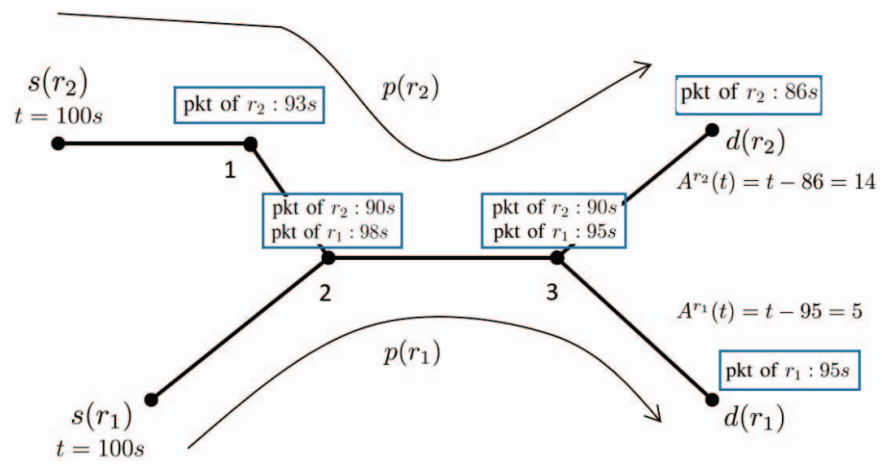


Figure 2.6. An example of multi-hop network [42], © 2017 IEEE.

2.4.5. AoI Optimization in Relay Networks

Due to the energy constraints on IoT end devices, relay is a widely used element in IoT networks. However, this affects the freshness of information depending the operation performed. In [74], AoI is considered as a QoS metric in relay-based IoT networks where IoT gadgets provide a variety of services, including CCTV monitoring and video applications. Utilizing AoI, object location and service relevance measures, the authors cluster IoT objects and propose an algorithm to improve network mobility performance.

Decode-and-forward (DF) type relaying in an intelligent IoT monitoring system in which relay modes can be either half-duplex (HD) or full-duplex (FD) is considered in [73]. Minimization of AoI for two distinct relaying modes in the context of finite block length is examined. To minimize average AoI, the average AoI with optimal block lengths in two phases is approximated for the HD-DF type relaying, whereas for FD-DF type relaying, an iterative algorithm is proposed. Finally, the efficacy of the proposed methods is validated numerically.

2.4.6. Effect of Preprocessing on AoI Optimization

In context-aware real-time IoT applications, the raw data should be preprocessed to optimize the use of wireless spectrum and to provide end users with context-aware services. Computing-enabled IoT networks represent these applications in the literature. In this context, researchers have investigated packet combination, packet length optimization, and compression techniques to enhance AoI performance. Additionally, there is a growing focus on jointly optimizing preprocessing and re-transmission policies, particularly in dense IoT monitoring scenarios.

Peak AoI is considered in [117] as a freshness metric in computing-enabled IoT networks with numerous sensors. Closed-form expressions for the average PAoI are obtained, then computation and transmission are characterized as tandem queues. To minimize the average PAoI, a derivative-free approach that determines the optimal frequency of status updates is employed.

Instead of broadcasting separately, the sink node can recreate a single transmission packet by merging packets from many IoT sensors. This allows for the elimination of unnecessary information, such as headers, and the improvement of transmission efficiency. This system is considered in [118] for two distinct data gathering modes, namely periodic request and proactive request. After deriving the AoI, the authors propose a low-complexity scheduling algorithm to minimize instant AoI and numerically demonstrate the efficiency of the packet combination for minimizing AoI.

In [70], a wireless networked control system (WNCS) is considered in I-IoT networks in which the transmission packet length is adaptively adjusted to optimize reliability and AoI performance. A novel WNCS is proposed and a decision problem is formulated. The optimal variable-length packet-transmission policy for cost minimization is then presented and a sufficient and necessary condition for the existence of the optimal policy is derived.

In [77], a computing-enabled dense IoT monitoring system with CSMA scheme is considered, in which a large number of IoT devices may preprocess the raw sensor data. Two policies are considered: the preprocess-then-sense (PtS) policy and the preprocess-while-sensing (PwS) policy. For the former approach, each IoT device begins sensing the channel when preprocessing is complete, whereas for the later policy, preprocessing and sensing are performed simultaneously. The authors derive closed-form expressions of the average AoI for both policies and propose a mean-field approximation framework for the large population regime. Employing extensive simulations, the accuracy of the proposed approximation is demonstrated under policy PtS.

In [119], the joint design of the preprocessing and transmission is considered to minimize the weighted total of the average AoI and energy consumption of the IoT device. Due to the nonuniform durations of transmission and computation, the problem is formulated as an infinite-horizon, average-cost, semi-Markov decision process (SMDP). The authors then transform the problem into a discrete-time MDP and show that a policy based on thresholds is the optimal solution. In conclusion, simulations reveal that preprocessing is successful for high AoI regimes and that the proposed approach outperforms two baseline policies.

In [51], time-consuming and computationally intensive preprocessing effect is considered in edge-assisted IoT devices with energy harvesting tags in order to minimize the average AoI within an energy cost budget. Due to the stochastic nature of the energy harvesting process and the energy budget constraint, the authors propose a Lyapunov-based average AoI minimization algorithm to approximate an optimal solution and also deduce how distant the approximate solution is from the optimal one. Extensive simulations show that the proposed algorithm outperforms existing schemes. The authors in [120] assume that there is no energy budget constraint and aim to minimize average AoI and total energy consumption by proposing a joint preprocessing and transmission policy. The problem is formulated as an average cost SMDP before transforming it into a discrete-time MDP. The optimal solution is proven to be threshold-based policy. A low-complexity relative policy iteration algorithm is proposed for determining the

optimal policy of the SMDP. The authors conclude by simulations that the proposed policy outperforms two baseline policies. The authors in [121] investigate a novel online compression technique that eliminates the tradeoff between AoI and compression ratio by considering an instantly decodable variant. Using both predefined and dynamically generated dictionaries at the source nodes, techniques for enhancing AoI and compression performance are proposed. Using real-world datasets, it is demonstrated that the proposed method improves AoI (by up to a factor of 2.3) and compression ratio (by up to an order of magnitude) in comparison to DEFLATE and Lempel–Ziv–Welch (LZW).

Table 2.1. Comparison of learning-based AoI optimization works.

Paper	**Proposed Method**	**Algorithm**	**Objective**
[122]	Model free RL	Zero-wait baseline policy	AoI-energy consumption in caching-enabled IoT
[123]	DRL	MDP policy	Offloading in multi-device computing-enabled IoT
[124]	RL	Greedy policy	VNF scheduling
[125]	Distributed RL	Deep Q-network, Uniform sampling policy	AoI and sampling policy
[126]	DRL	5 baseline policies	AoI and energy consumption
[53]	Q-learning	Several baseline policies	AoI and energy consumption
[127]	Deep Q-network with Tabular R-learning	5 fundamental DRL algorithms and policies	AoI and energy consumption
[128]	Low-complexity online algorithm	Greedy policy, Max-Ratio policy	Under noisy channel
[129]	DRL	-	Autonomous maintenance
[130]	RL	-	Under dynamic channel access attacks

2.5. Learning Based Methods for AoI Optimization

In dense IoT networks, the number of IoT end devices is high. Therefore, most of the problems such as scheduling and path optimization are NP-hard and non-convex. Finding globally optimal solutions is difficult. In recent years, machine learning algorithms have been widely used to optimize such problems such as deep Q-network and DRL. In this section, we will present works that use machine learning algorithms to optimize AoI in IoT networks. In the first subsection, works considering reinforcement learning for AoI optimization are examined, while in the second subsection, works considering AoI as a cost metric and perform trajectory optimization with learning-based methods are examined for the data collection models.

2.5.1. AoI Optimization with Reinforcement Learning

The authors in [122] address dynamic status update problem to balance the information freshness experienced by users and energy consumed by the sensor in caching-enabled real-time IoT networks. In order to solve this problem, a MDP is formulated and a model-free reinforcement learning algorithm is proposed. Finally, the convergence of the proposed algorithm and its efficacy are demonstrated by simulations comparing to the zero-wait baseline policy.

In [123], a multi-device computing-enabled IoT system with a common destination is considered. In this system, an IoT end device can either offload the computation of the status update to the destination node or send the computed status update to the destination. To minimize the average weighted sum of AoI and energy consumption, the design of offloading and scheduling policies is considered jointly. The optimization problem as a bilevel discrete-time MDP is formulated and the optimal solution is approximated by relative value iteration. The authors develop a DRL policy to reduce the dimensionality caused by the expansion of the system's scale; simulation results show that the proposed DRL policy is close to optimal.

Virtual network functions (VNFs) provide service providers with the flexibility to deliver a particular network service. However, VNF scheduling is an NP-hard problem, and traditional methods for finding an optimal solution are too complex. Recently, DRL has emerged as a viable solution for such problems. In [124], both single agent RL and multi-agent DRL are utilized to optimize VNF cost and AoI in terms of network resources under QoS constraints in delay-sensitive industrial IoT. Simulation results indicate that a single-agent solution is superior to a greedy algorithm in terms of AoI and network cost. In addition, the authors state that a multi-agent DRL design in which agents cooperate significantly reduces the average cost despite requiring more iterations.

In [125], minimizing the weighted sum of AoI and total energy consumption of IoT devices is aimed in real-time IoT systems. In the considered model, each IoT device samples the physical observation at a cost of some energy under nonlinear real-time dynamics and transmits data to a BS. However, wireless resource limitations compel BS to select a subset of IoT devices for transmission, resulting in an increase in the monitoring time for physical observations. From this perspective, it is essential to jointly optimize the device selection procedure at the BS and sampling policy of each IoT device to reduce total energy consumption and AoI. The authors propose distributed reinforcement learning for the AoI and sampling policy optimization, in which end devices discover optimal sampling frequency in collaboration. Using simulations on real pollution data, it is demonstrated that the proposed strategy outperforms a conventional deep Q network method and a uniform sampling policy in terms of the weighted sum of AoI and energy consumption.

The primary objective of [126] is to identify the best status update control mechanism on the edge node in order to reduce the average long-term cost of AoI and energy consumption. In this model, energy harvesting sensors measure a physical observation and transmit the data to a wireless edge node that serves as a gateway between sensors and users. Due to storage limitations, the edge node only stores the majority of received data. The edge node either sends a request to sensors to update its status or transmits

the data already stored in the storage. Therefore, the edge node control mechanism faces a trade-off between energy consumption and AoI. The authors propose a DRL algorithm to find the optimal control policy without knowledge of the sensor's energy levels. The authors in [53] also consider that when the status is updated, edge nodes are aware of the battery levels of the sensors. The problem is modeled as a MDP, and a Q-learning algorithm to minimize the average AoI is proposed. Simulations show that a threshold-based policy is optimal and the proposed reinforcement learning algorithm significantly reduces the average cost compared to several baseline methods.

A similar network model is considered in [127]. where a MDP is formulated and a dueling deep R-network based dynamic status update algorithm is designed by combining a deep Q-network with tabular R-learning. The effectiveness of the proposed algorithm is validated by comparing it to five fundamental DRL algorithms and policies.

The authors of [128] examine a scenario in which edge nodes sample the status using multiple packets and transmit it to the destination node over a noisy channel. Low-complexity online algorithms are proposed and simulation results demonstrate that the Lyapunov drift policy outperforms the Greedy policy and the max-ratio policy.

In [129], the autonomous maintenance problem in IoT networks is formulated as a partially observable MDP. The authors numerically show that the utilized DRL algorithm is effective for training intelligent agents to determine whether or not maintenance is required.

The authors in [130] aim to minimize AoI in multi-channel IoT networks under dynamic channel access attacks (DCAAs). The distributed channel access problem is then formulated, and a reinforcement learning algorithm is implemented based on the theoretical outcomes of game theory. A utility maximization problem for each sensor node's average AoI is formulated under DCAAs with probabilistic ACK feedback. The problem is then converted to two ordinary potential game (OPG) models, and a distributed learning algorithm is proposed to reach Nash equilibrium (NE). Lastly, the

efficacy of the proposed algorithm is validated by simulating with various variables. Table 2.1 shows a simple summary of learning-based approaches for AoI optimization.

2.5.2. AoI Based Trajectory Optimization for Data Collection Model

Owing to their adaptability and low deployment cost, UAVs are expected to play a crucial role in the collection of data from widely dispersed ground sensors over vast areas. In general, UAV-assisted IoT networks are considered, in which battery-limited UAVs are expected to collect physical observations captured by ground sensors and wirelessly transmit data to a monitoring station. Due to the energy constraints of ground nodes, a sensor is unable to transmit its physical observations unless a UAV is in close proximity to its region. Fig. 2.7 illustrates the significance of the UAV's flight path. In order to minimize the weighted sum of AoI, [131] considers the joint optimization of scheduling policy on ground nodes and flight trajectory of the UAV under battery constraints. The problem is formulated as a MDP with finite state and action spaces and a finite horizon. Due to the high dimensionality of the state space, an age-optimal policy-finding algorithm based on DRL is proposed. Simulations indicate the proposed deep RL method outperforms distance-based and random walk policies. Pre-trained deep neural network is used to accelerate the learning process as explained in [75].

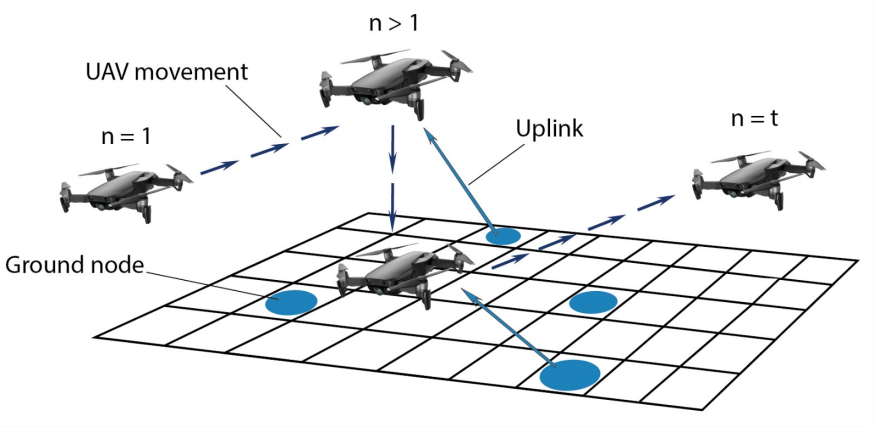


Figure 2.7. An example of UAV data collection model.

In [132], the flight trajectory of an energy-constrained UAV is optimized using a DRL method to decrease the average AoI at ground sensor nodes (SNs) while maintaining the lowest possible packet drop rate. A sample-and-replace policy refers to SNs that sample observations with a defined or random frequency and cache the sampled packet updates. The researchers define the data gathering problem as an MDP with finite state and action spaces and a finite horizon. Simulations demonstrate that the proposed technique outperforms the benchmark greedy strategy.

The same network architecture for UAV-assisted data collection and path planning is considered in IoT networks [133]. A dynamic programming approach is presented to optimize the UAV path planning and data acquisition problem by jointly considering the average AoI, sensor energy consumption, and sampling mode selection. Simulations reveal that the proposed method is competitive compared to the traditional flying time-optimal strategy.

In [134], a UAV-assisted wireless powered IoT system is explored, where it is assumed that the time required for sensors' energy harvesting (EH) from UAVs is nonlinear due to the nonlinear behavior of the EH circuit. The authors jointly optimize the UAV's trajectory, harvesting time, and data collection procedure at each sensor in order to minimize the average AoI of the data collected from all sensors. This nonconvex problem is divided into two subproblems: a combined energy transfer and time allocation for data acquisition problem and a UAV path planning problem. After showing the convexity of the first problem, Karush-Kuhn-Tucker (KKT) conditions are employed to identify the optimal solution, which serves as an input for the second subproblem. Simulation results reveal that DP outperforms ant colony heuristic algorithms in the solution of the defined problem.

The goal of [135] is to minimize the weighted sum of average AoI, energy required from UAVs and transmission energy at sensors and to this end, UAV flight speed, altitude of UAV and bandwidth allocation policy for data collection are jointly considered. The problem is modeled as a MDP. Twin-delayed deep deterministic (TD3)

policy gradient-based UAV path planning algorithm is proposed via deep neural network. Simulations demonstrate that the proposed technique outperforms the deep Q-network and actor–critic-based algorithms.

The authors in [136] assume that UAVs can update the sampling policy of IoT devices. On the other hand, UAV stores all past IoT device experience. A new algorithm is presented for lifetime reinforcement learning that utilizes these accumulated experiences. Simulations reveal that the proposed algorithm can converge more quickly than a baseline gradient algorithm. Table 2.2 summarizes the methods which are used for trajectory optimization in UAV-assisted IoT networks.

Table 2.2. Methods to optimize UAVs trajectory.

Deep Reinforcement Learning	[131], [132]
Dynamic Programming	[133], [134]
Gradient-based Neural Network	[75], [135]
Lifetime Reinforcement Learning	[136]

Different from the previous studies, data collecting in the ocean using autonomous underwater vehicles (AUVs), which is modelled in Fig. 2.8 and proposed in [137]. Due to the vast area and large-scale deployment cost of sensor nodes in the Internet of underwater things (IoUT), heterogeneous multi-AUV-assisted underwater data collection is modeled to optimize the peak AoI under time-varying underwater environment, energy and mobility constraints of AUV. In addition, the process of information exchange is modeled using the limited service M/G/1 vacation queueing model. The authors also provide a low-complexity adaptive algorithm for adjusting the upper limit of the queueing length. Simulations demonstrate that the proposed strategy is superior to conventional methods for minimizing peak AoI.

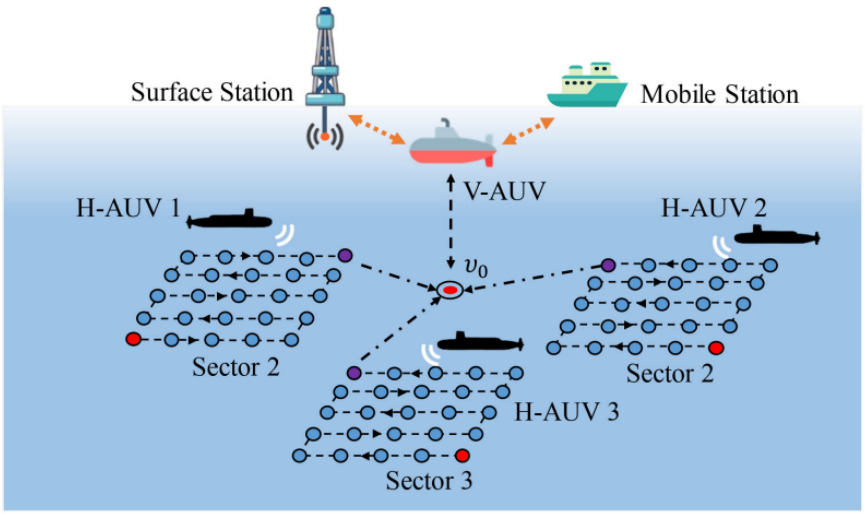


Figure 2.8. An example of AUV-assisted data collection model [137], © 2021 IEEE.

2.6. Scheduling and Sampling Issues for AoI in IoT

One of the important research areas on AoI is designing schedulers to minimize AoI for information sources in a given network. The network may consist of a single-link, multi-link and may be cellular or a multi-hop network. Sophisticated design of a scheduler to minimize AoI is crucial as it must address the requirements on AoI metric depending on the criteria of the applications. To this end, we consider the works addressing the scheduling problem in IoT networks in this section.

2.6.1. Scheduling Optimization

Scheduling plays a crucial role in AoI optimization, determining when and how data updates are transmitted. Effective scheduling algorithms are required to reduce information staleness and optimize network performance while considering dynamic network conditions, varying data arrival patterns, energy constraints, and diverse application needs.

The authors of [138] consider a scenario in which several terminals share a single channel for transmitting or receiving status update packets. The scheduling is modeled

as a restless multi-armed bandit problem inspiring from and exploiting the Whittle's index policy to incorporate stochastic packet arrivals, a decentralized status updating technique is proposed that provides universally near-optimal AoI performance.

To enable devices to share limited communication resources, [139] proposes centralized and distributed resource allocation techniques. The transmissions are intended to be completed in a timely manner, with AoI capturing the measurement of timeliness. Adopting nonlinear aging functions that are reliant on device type and message content, as opposed to the majority of previous works, implies the inherent heterogeneity of the network.

Trying to achieve a high data rate in the network may result in congestion, hence diminishing the update's freshness at the monitoring device. On the other hand, a low data rate probably results in degraded freshness due to the increasing arrival periods between status updates. In [90], a novel transport layer protocol is proposed to handle such a trade-off. The aforementioned protocol adjusts the update data rate to reduce the average AoI of updates at the monitor.

In [140], a dynamic IoT network is considered and a distributed scheduling technique is proposed to optimize the AoI in the network.

Unlike majority of the previous works, the authors in [141] focus on devices where typically support monitoring applications. Due to the computational limitations of such devices, simple first-in-first-out (FIFO) queueing discipline is presumed. Consideration is given to the problem of optimal status update generation when a source consisting of a FIFO queue serves two applications, one of which is sensitive to AoI and the other of which is not.

In [142], a UAV-assisted IoT network is considered where, IoT devices periodically sample a stochastic process and transmit updates to a BS. Due to their limited powers, IoT devices may require intermediary nodes to forward update packets to BS.

Considering this fact, UAVs with virtual queues are deployed as intermediary nodes between IoT devices and the BS in order to transmit information updates over unreliable channels. In the absence of channel conditions, the optimal online scheduling method is examined with the purpose of minimizing the expected weighted sum AoI for IoT devices. In a similar work [143], various scheduling policies for AoI minimization in two-hop UAV-relayed IoT networks are presented and compared. Maximum AoI first policy for sampling of IoT devices at UAV (hop 1) and maximum AoI difference policy for updating sampled packets from UAV to the BS (hop 2) outperform the others in terms of scaling to larger networks.

2.6.2. Impact of Varying Packet Size on Scheduling

Due to the dynamic nature of real-world physical processes, the sizes of status updates are typically not constant, resulting in varying sizes and consequently varying numbers of transmission slots for distinct updates. Therefore, there is a need for AoI optimization that take such facts into consideration. In [144], multi-time slot transmissions with non-uniform sizes of the status updates are considered as in a realistic scenario and the problem of joint device scheduling and status sampling is investigated with the objective of minimizing the average AoI at the destination. Similarly, an AoI-aware scheduling scheme for sensors is proposed to send status updates to cloud data center-based caches in [145].

In [146], heterogeneous sampling behaviors among source nodes with varying sample size and transmission slots are considered, and theoretical results for AoI are obtained. In addition, a near-optimal scheduling algorithm with low-complexity is proposed to minimize the AoI. The impact of stochastic arrivals on the peak AoI of NOMA-based IoT networks is investigated in [147] where the authors propose a method for determining the queue service rate and derive the analytical expression for the distribution of peak AoI. In this scheme, NOMA outperforms OMA standard in terms of network average peak AoI.

The multicast transmission of an access point to multiple IoT devices is examined in [148], where a deadline is enforced for the service time of each multicast status update, and the update is dropped when the deadline is reached. The authors evaluate the timeliness of status updates in this multicast scheme and show that enforcing deadlines reduces the network's average and peak AoI.

2.6.3. Sampling Optimization

Sampling rate in IoT is crucial for AoI optimization, reducing AoI with higher rates due to more frequent updates. However, this comes with increased sensor energy consumption, affecting overall efficiency. Striking the right balance between AoI reduction and energy usage is vital for sustainable IoT deployment. Optimization of sampling techniques, such as sleep mode selection and retransmission enhancements, is actively investigated to enhance data collection efficiency while conserving energy resources especially in industrial IoT networks. Optimization problems under energy constraints, RF-powered wireless energy transfer, and signaling overhead are also considered.

A typical industrial IoT application where multiple sensors monitor time-varying physical processes and send the information packets to a central BS through unreliable channels is given in [149]. Contrast to other approaches, each sensor may switch to sleep mode to save energy. For the setup, the authors formulate an AoI penalty minimization problem to schedule sensors' transmissions. After establishing a theoretically necessary condition for AoI penalty to converge to a finite value, a maximum weight based scheduling strategy is developed and its optimality is proved for symmetric networks with error-free channels. The primary goal of [150] is to covertly collect data from sensors in order to evade enemies. Retransmission is employed on the transmitter side to improve AoI in the event of a failed packet transmission, hence increasing the likelihood of adversary detection. A time-restricted retransmission policy is proposed to exploit the AoI security trade-off, as longer retransmission durations enhance the system's susceptibility.

In [151], the authors consider the joint status sampling and updating problem in which IoT devices sample a physical process with a sampling cost and deliver the samples to a particular destination with an updating cost. The objective is to minimize the average AoI subject to an average energy cost constraint at each device. An online learning method that may be implemented locally on each IoT device and converges almost surely is proposed. Similar research can be found in [152], which investigates the optimal online sampling strategy that minimizes the average AoI when the source node is considered to be powered by wireless energy transfer from the destination. The difference between age-optimal and throughput-optimal approaches is derived analytically.

In [153], an RF-powered source sends status updates to a destination node and the main goal is to minimize the average AoI at the destination subject to time and energy costs of generating status updates at the source.

Since blockchain is a promising technology for ensuring the transparency, reliability and traceability of networked information exchange, its usage has extended beyond cryptocurrencies. In [154], the sampling rate optimization problem is solved for IoT sensors which employ blockchain and tangle technologies for their transmission, with the aim of minimizing AoI experienced by end-users susceptible to processing and networking resources.

In [155], the authors investigate the optimal sampling policy which jointly optimizes wireless energy transfer and scheduling of status updates with the objective of minimizing the weighted sum-AoI in an environment in which IoT sensors sense potentially different physical processes and update a destination node via status updates.

In [156], randomly generated status updates are transmitted to a destination node over an unreliable link with the goal of minimizing the average AoI subject to limited average transmission power at the source.

In [157], the primary objective is to obtain a scheduling policy with minimal AoI and minimal signaling exchange overhead and the optimal policy is shown to be a round-robin policy with the most recent packet, while the others are dropped; the optimality of this policy is proven using a generalized Poisson-arrival-see-time-average theorem.

2.7. Experimental and Application Based Research

Finding a parking spot during rush hour in a congested city is challenging. The problem is that information regarding available parking places is not always readily accessible. IoT and advanced data analytics have paved the way for smart parking solutions to handle the problem of detecting available parking spaces in cities and garages. A smart parking IoT application utilizes data from IoT devices to recommend an available parking space in the vicinity of the target location.

According to [158], 21.4% of parking sensors generate data that is several hours old in real-world conditions. In a case study of a smart parking IoT application, it is demonstrated that an age of data (AoD) model may be used by the application to generate AoD-aware recommendations. A parking data set analysis from the real life is also presented to demonstrate the applicability of the proposed method.

In [159], the AoI behavior on experimental implementations of packet flows between IoT nodes and via the Internet, including the IoT access link, and end-to-end UDP and TCP flows is explored from the standpoint of computer applications. Practical considerations are examined such as synchronization and hardware selection, as well as the transport protocol and its effect on AoI. The findings shed light on application and transport layer techniques that can be used to optimize AoI in actual networks.

As a result of recent improvements in blockchain technology, there has been a rise in interest in blockchain-based application development. The values stored in a blockchain can be withdrawn or altered while the data remains intact. It is not al-

ways clear whether the discovered data from the blockchain storage is relevant for blockchain-based applications' real-time decision-making processes. Consequently, the issue of data freshness is crucial, particularly in dynamic networks that process data in real time. According to [160], transactions to refresh data require additional processing time within the blockchain network, which is known as ledger-commitment delay. Therefore, some users may receive obsolete information. Since it is critical to determine if blockchain technology is suitable for supplying real-time data services, the authors of [160] define the AoI of blockchain-enabled networks and explore the factors that affect the AoI.

In [54], a system in which external requests for status updates of a remote source are received and monitored by an EH sensor is considered. Based on a probabilistic approach, an external request for a status update may be satisfied by retrieving either a freshly generated update from an EH sensor or a previously cached update from an aggregator. The average AoI is calculated as a function of several system characteristics, and the findings reveal how the energy delivery procedure and the status updating rule affect the system's information freshness.

In [161], a novel strategy for reducing storage costs known as generalized deduplication (GD) was proposed. Recently, this method has been extended to provide distributed, multisource lossless compression in sensor networks, hence lowering the total number of bits transferred. Considering the scenario in which one source node receives one symbol/sample per unit time and transmits bits to the sink node, the authors demonstrate how GD can provide instant decoding of data, thereby reducing the average age, and demonstrate that the proposed solution reduces the AoI by 25% for the standard version and 36% for the instantly decodable version when using real-world data sets.

2.8. AoX in IoT

Researchers have introduced novel metrics in the exploration of AoI in IoT, such as age upon decisions (AuD), age of changed information (AoCI), age of processing (AoP), and spatially temporally correlated mutual information (STI). These metrics address different aspects of data freshness and processing in IoT systems. Additionally, CVaR is proposed for risk-aware AoI minimization in real-time IoT applications. Coage of information (CoI) considers the maximum AoI of constituent IoT devices, leading to CoI-aware policies for data transmissions. These advancements offer valuable tools for optimizing IoT systems and decision making processes.

AuD is proposed which represents the time difference between the timestamp of the usage of last received information to make a decision (e.g., estimations, inferences, and controls) and the timestamp of its generation in [162] and [163], respectively. In these works, an IoT network based on the premise is considered, in which sensor updates have exponential service times and FCFS discipline. Scheduling schemes that adjusts the status update and decision-making process are proposed in order to improve the timeliness of updates at decision epochs. Three key theoretical results are demonstrated. The first is that the average AuD is minimized when the arrival process is periodic and the decision process is Poisson. The second result is that the average AuD is greater when both the choice process and the arrival procedure are periodic. Finally, by adjusting the offset between the two periodic processes, the average AuD can be minimized when both processes are periodic. Periodic arrival and random decision are suggested for the use in actual IoT systems.

In [164], minimization of the weighted sum of AoS of serverless services is considered for the problem of information update delivery and acquisition (IUDA) in IoT networks. The model consists of a set N of IoT devices and the execution of a set F of serverless functions is illustrated in Fig. 2.9. An offline scheduling approach is proposed after the occurrences of serverless function arrivals are known. In addition, the authors analyze the scenario in which there is no knowledge regarding the arrival

of serverless functions and propose an online algorithm. Simulations reveal that the proposed algorithms outperform baseline solutions regarding the AoS performance.

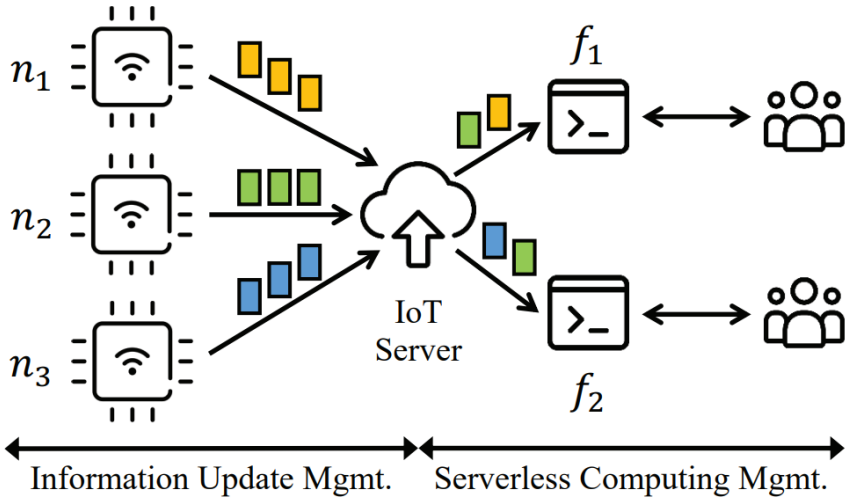


Figure 2.9. An IoT system with three IoT devices and two serverless functions [164].

In [165], the authors criticize the minimization of AoI's expected value due to its risk-neutral behavior. CVaR is proposed for the minimization of AoI in real-time IoT applications and an optimal status update mechanism is designed by jointly considering the AoI, the CVaR of the AoI, and the energy cost. Using dynamic programming, the optimal stationary policy is proposed after reducing the risk-aware MDP to a standard MDP.

AoCI is a novel metric defined in [166] that takes into account both the freshness over time and freshness over changed content. The authors model the process as a discrete time MC and formulate an infinite horizon average cost MDP in order to minimize the weighted sum of the AoCI and update cost. The optimal policy under a variety of circumstances are investigated and closed-form expressions are derived for the optimal solution.

In intelligent IoT applications such as video surveillance, the status update is generated after computation-intensive data processing. The authors in [167] define a

novel metric, AoP, to account for these time-consuming processing activities, which influence the freshness of the status. The procedure is illustrated in Fig. 2.10.

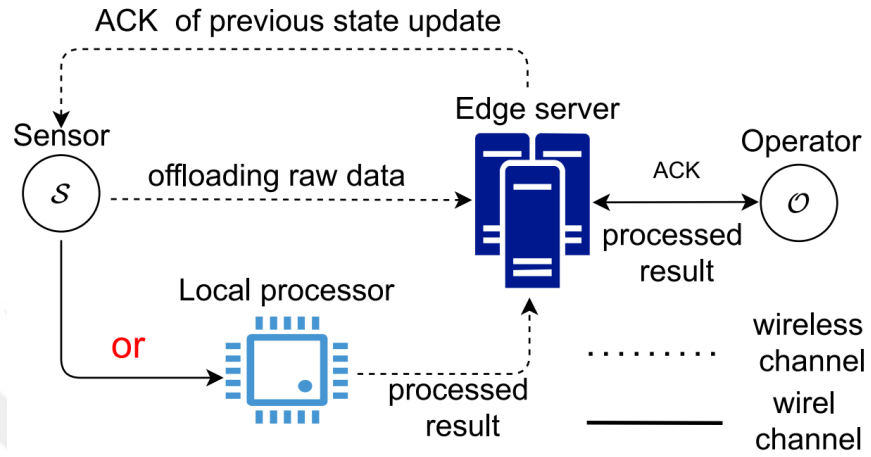


Figure 2.10. An example of status sampling and processing procedure [167], © 2021 IEEE.

The authors of [168] propose a novel metric, STI, by considering the correlation between the current status and the most recent update message. After formulating the problem as an MDP with potentially infinite state space, a policy is proposed to periodically update in order to maximize the average STI in satellite-integrated IoT networks. In conclusion, it is shown by simulations that the proposed policy outperforms both slotted ALOHA and threshold-ALOHA.

According to [169], CoI (or information cofreshness) is determined by the maximum AoI of the constituent IoT device. The authors address the problem of grant assignment and transmission scheduling (GATS) and propose the CoI-aware age block elimination policy to minimize a weighted sum of CoI.

2.9. Conclusion and Discussion

In the literature review of this thesis, we have first presented the basics of AoI and IoT, then we have presented the studies in which AoI is used as a timeliness metric

in various types of IoT networks. Furthermore, we investigated optimization problems, learning-based algorithms, and scheduling policies. Then, we stated the application-based AoI studies. Finally, we defined the term AoX which stands for some additional constraints on the freshness of timeliness.

3. AGE OF INFORMATION MINIMIZING DYNAMIC USER PAIRING IN DOWNLINK NOMA SYSTEMS

3.1. Introduction

Notice that in conventional NOMA systems two main factors need to be optimized: i) user clustering/pairing and ii) power allocation among users and clusters. Since complex power allocation strategies may not be feasible in many practical deployment scenarios, often pre-defined power allocation strategies such as the one given in [170] are employed, leaving the user clustering/pairing optimization as the only main problem to be solved. The significance and the possible gains implied by user clustering/pairing have been shown in prior works such as [7, 171, 172]. As mentioned above, a small number of works investigating NOMA systems from an AoI perspective such as [18–20] exist but none have focused on the user grouping/pairing problem depending on user rate requirements.

We consider a downlink communication system employing power domain NOMA with two users in each resource block and with given variable user rate requirements. The goal is to minimize the average AoI in the network by dynamically pairing the users according to varying user rate demands. Since the original minimization problem is NP-hard and thus infeasible to be solved in real life applications, a graph-theoretical dynamic pairing solution, where the original average AoI minimization is transformed to overall transmission success probability maximization, is proposed. Owing to the claw-freeness property of the conflict graph in this average AoI optimization setting, the maximum weighted independent set (MWIS) can be found in polynomial time using the algorithm given in [173]. The resulting MWIS provides the average AoI minimizing pairing as confirmed by the simulation results which show a significant decrease in the average AoI compared to standard OMA model and the benchmark pairing models proposed in [170, 172].

Notice that in [7] and [174] cellular NOMA networks and wireless ad hoc networks are considered, respectively where graph theoretical solutions are devised to sum-rate maximization problems. This work differs from these earlier works in the sense that average AoI minimization is considered as the primary objective function which is transformed into a conflict graph and solved using a MWIS approach.

The rest of this chapter is organized as follows: In Section 3.2 we present the conventional NOMA user pairing approaches used as benchmark in this work followed by the system model in Section 3.3. Then in Section 3.4 we show how average AoI minimization problem can be reformulated as that of finding MWIS on a conflict graph and demonstrate the solution methodology. Simulation results are presented in Section 3.5 before the concluding remarks in Section 3.6.

3.2. User Pairing Benchmarks in NOMA

Due to the NP-hardness implied by the nature of the user pairing optimization, the original problem requires exponential time in terms of the number of users in the network. A number of heuristic solutions, which are considered as benchmarks here, are presented in [170, 172] to this problem.

In [172], a practical method is proposed where a user with a very good channel is paired with one that has a moderate channel condition and a user with a moderate channel state is paired with another that has a very bad channel, etc. This pairing strategy can be written as

$$p_{k,m} = \begin{cases} 1, & \text{if } |m-k| = \dfrac{N}{2} \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

where $p_{k,m} = 1$ implies the pairing of users k and m assuming the channel coefficients $|h_k| \geq |h_m|$ for $k < m$ where $k, m \in \{1, 2, ..., N\}$ and N denotes the number of users.

Another approach in [170] considers the same optimization problem from a geometrical perspective. A disc is formed at the center of the cell and a ring is formed at

the outer part. For each pair, one user is sampled from the disc and another is sampled from the ring in order to guarantee the pairing of a near-user that has a good channel condition with a far-user that has a sufficiently worse channel condition.

3.3. System Model

We consider a cellular downlink communication scenario where a BS, which is located at the center of a disc having a radius r, serves randomly located N users inside the disc. The generate-at-will status update model is adopted so that a new status update for each user is generated at the BS at the beginning of each time slot. The BS aims to deliver these generated packets to all users in a timely manner. Each user is assumed to send back an acknowledgment (ACK) packet in an instantaneous and error-free manner when it receives a status update packet.

3.3.1. AoI Model

The main goal is to minimize the average AoI in the considered cell in order to guarantee the best achievable freshness of information in the network. The AoI can easily be measured as

$$\Delta_k(t) = \begin{cases} \Delta_k(t-1) + 1, & \text{if } r_k(t-1) = 0 \\ 1, & \text{if } r_k(t-1) = 1 \end{cases} \tag{3.2}$$

where $\Delta_k(t)$ is the AoI of user k at time t and $r_k(t) = 1$ denotes the presence of a successfully received packet for user k at time t and $r_k(t) = 0$ denotes the absence. In other words, AoI for a user becomes 1 when it successfully receives a status update packet and increments by 1 in each time slot if it does not receive any update. Then, the average AoI in the network, with equal importance for all users, is given by

$$\Delta = \lim_{T\to\infty} \frac{1}{NT} \mathbb{E}\left[\sum_{t=1}^{T}\sum_{k=1}^{N} \Delta_k(t)\right]. \tag{3.3}$$

3.3.2. NOMA System Model and the Outage Probabilities

In the considered scenario, there are $N/2$ sub-carriers, with N being an even number, and each sub-carrier is assigned to a pair of users where more than two users in a cluster are not allowed. Instantaneous channel coefficients between users and BS are denoted by $h_k = \sqrt{d_k^{-\gamma}} g_k$ where d_k is the distance between BS and user k, γ is the path-loss exponent and g_k represents the small-scale fading coefficient which is assumed to be Rayleigh distributed. It is assumed that the BS is able to estimate the distances of users by exploiting the multiple channel coefficients it received from each and every user.

In this scenario, the BS is assumed to know the exact distances of users whereas the exact channel coefficients are unknown since they can instantaneously change due to fading and shadowing effects. Knowing the exact distances, estimates of channel coefficients are calculated at BS to be used in the rate calculations of users as seen in equations (3.5) and (3.6).

In the next step of the system model, power allocation scheme is important to continue to solve the optimization problem. A pre-defined power allocation scenario as proposed in [170], where the BS transmits signals with an aim that equal received powers at user sides are maintained with the goal of maximizing the fairness in the system, is considered. Therefore, power allocation coefficients are assigned as $a_k = d_k^{\gamma} / \sum_{i=1}^{N} d_i^{\gamma}$, $\forall k \in \{1, 2, ..., N\}$.

In this case, the signal received by user k, which is paired with user m in the same sub-carrier, can be mathematically given as

$$y_k = h_k \sqrt{a_k P_{tot}} s_k + h_k \sqrt{a_m P_{tot}} s_m + n_k \tag{3.4}$$

where P_{tot} is the total transmit power of the BS, s_k denotes the message sent to user k and n_k denotes the additive white Gaussian noise (AWGN) with zero mean and variance $N_0/2$.

Assuming $|h_k| \geq |h_m|$ for $k < m$ throughout the chapter, after receiving the signal, user k performs SIC in order to decode the message destined to user m, having a poorer channel condition, and then removes this message from the overall signal to decode its own message. On the other hand, user m directly decodes its own message by treating the message destined to user k as noise. At the end of decoding operations, given that SNR denotes the transmit signal-to-noise ratio of the BS, the data rates of users m and k are respectively expected to be mathematically as

$$R_m = \log_2 \left(1 + \frac{a_m |h_m|^2}{a_k |h_m|^2 + \frac{1}{SNR}}\right) \tag{3.5}$$

$$R_k = \log_2 \left(1 + a_k |h_k|^2 SNR\right). \tag{3.6}$$

Assuming that an outage occurs for user m when the available data rate, R_m, is below its target rate R_m^T, i.e., $R_m < R_m^T$, the outage probability, $P_{out,m}$, can mathematically be given as

$$\begin{aligned} P_{out,m} &= Pr\left(R_m < R_m^T\right) \\ &= 1 - Pr\left(R_m \geq R_m^T\right) \\ &= 1 - Pr\left(\log_2\left(1 + \frac{a_m |h_m|^2}{a_k |h_m|^2 + \frac{1}{SNR}}\right) \geq R_m^T\right) \\ &= 1 - Pr\left(|h_m|^2 \geq \frac{2^{R_m^T} - 1}{SNR\left(a_m - a_k\left(2^{R_m^T} - 1\right)\right)}\right) \\ &= 1 - \exp\left(-\frac{d_m^\gamma \left(2^{R_m^T} - 1\right)}{SNR\left(a_m - a_k\left(2^{R_m^T} - 1\right)\right)}\right) \end{aligned} \tag{3.7}$$

since user m only needs to decode its own message in the given NOMA network setup. The target rate is chosen as R_m^T to satisfy $\left(a_m - a_k\left(2^{R_m^T} - 1\right)\right) > 0$ because the power allocation coefficients are constant and are specified beforehand in the considered scenario. By maintaining these constant power allocation coefficients, the network can achieve efficient and reliable communication tailored to the needs of each user. Furthermore, this predetermined setup helps in simplifying the overall network design and reducing computational overhead.

On the other hand, the outage probability for user k, P_k^{out} is given by

$$\begin{aligned}
P_{out,k} &= Pr\left(\left(R_{k(m)} < R_m^T\right) \vee \left(R_k < R_k^T\right)\right) \\
&= 1 - Pr\left(\left(R_{k(m)} \geq R_m^T\right) \wedge \left(R_k \geq R_k^T\right)\right) \\
&= 1 - Pr\left(\left(\log_2\left(1 + \frac{a_m|h_k|^2}{a_k|h_k|^2 + \frac{1}{SNR}}\right) \geq R_m^T\right) \wedge \right. \\
&\left. \left(\log_2\left(1 + a_k|h_k|^2 SNR\right) \geq R_k^T\right)\right) \\
&= 1 - Pr\left(\left(|h_k|^2 \geq \frac{2^{R_m^T} - 1}{SNR\left(a_m - a_k\left(2^{R_k^T} - 1\right)\right)}\right)\right. \\
&\left. \wedge \left(|h_k|^2 \geq \frac{2^{R_k^T} - 1}{a_k SNR}\right)\right) \\
&= 1 - \exp\left(-\max\left\{\frac{d_k^\gamma\left(2^{R_m^T} - 1\right)}{SNR\left(a_m - a_k\left(2^{R_k^T} - 1\right)\right)}, \right.\right. \\
&\left.\left. \frac{d_k^\gamma\left(2^{R_k^T} - 1\right)}{a_k SNR}\right\}\right)
\end{aligned} \tag{3.8}$$

where $R_{k(m)}$ denotes the decoding rate of message for m at user k since user k first decodes the message destined to user m by using SIC and then decodes its own message. The rate of user k in the benchmark OMA system can be written as

$$R_k = \log_2\left(1 + 2a_k|h_k|^2 SNR\right) \tag{3.9}$$

if user k is assigned a resource block for the considered time slot. Note that twice as much power as that of the NOMA system is allocated to user k in the OMA setup since only half of the users are served in each time block. The outage probability of user k, assuming user k is served in the considered time slot, in the OMA scheme can be obtained as

$$\begin{aligned}
P_{out,k} &= Pr\left(R_k < R_k^T\right) \\
&= 1 - Pr\left(R_k \geq R_k^T\right) \\
&= 1 - Pr\left(\log_2\left(1 + 2a_k|h_k|^2 SNR\right) \geq R_k^T\right) \\
&= 1 - Pr\left(|h_k|^2 \geq \frac{2^{R_k^T} - 1}{2a_k SNR}\right) \\
&= 1 - \exp\left(-\frac{d_k^\gamma\left(2^{R_k^T} - 1\right)}{2a_k SNR}\right).
\end{aligned} \tag{3.10}$$

As mentioned before, it is assumed that the power allocation is done in advance so as to maximize the fairness among users, we only consider the optimum user pairing between users on each resource block. The transformation of this original problem to its graph-theoretical counterpart is explained in the following section.

3.4. Problem Formulation and Graph-Theoretical Approach to AoI

In previous works, most researchers approach the AoI minimization problem by modeling users' (devices') states as a MC as seen in [18, 102, 175]. However, the MC model is insufficient for the average AoI optimal user pairing problem that is NP-hard as all possible pairings need to be tried. Note that the original problem, aiming to obtain an optimal user pairing, can mathematically be given by the expressions below.

Note that there are two constraints tied to the objective as

$$\begin{aligned} \arg\min_{\pi} \quad & \Delta(\pi) \\ \text{s.t.} \quad & \sum_{i=1}^{N} C_{i,b} = 2, \forall b \in \{1, ..., \frac{N}{2}\} \\ & \sum_{b=1}^{\frac{N}{2}} C_{i,b} = 1, \forall i \in \{1, ..., N\} \end{aligned} \tag{3.11}$$

where $C_{i,b} = 1$ means user i belongs to cluster b and $C_{i,b} = 0$, otherwise. Objective aims at finding the average AoI minimizing pairing in the system. The first constraint is used in order to guarantee that each cluster contains exactly 2 users and the second constraint ensures that each user can only be assigned to one cluster.

Due to the NP-hardness of the original problem, a transformation to a graph-theoretical counterpart is proposed. The original average AoI minimization problem is transformed to the problem of minimization of the total outage probability, $P_{out}(\pi)$, in the network to preserve the timeliness of information. Since $P_{out}(\pi) = 1 - P_{suc}(\pi)$, where $P_{suc}(\pi)$ is the error-free transmission probability from BS to all users in a NOMA scheme per time slot with a user pairing π, the problem is further converted to the

maximization of P_{suc} in the overall network such that

$$\pi^* = \arg\min_{\pi} \Delta(\pi) \rightarrow \arg\min_{\pi} P_{out}(\pi) \rightarrow \arg\max_{\pi} P_{suc}(\pi). \tag{3.12}$$

The error-free transmission probability of BS to user k in a pairing π is denoted as $P_{suc,k}(\pi)$. Also, $P_{suc,k,m}(\pi) = P_{suc,k}(\pi) * P_{suc,m}(\pi)$, where $P_{suc,k,m}(\pi)$ is the error-free transmission probability from BS to both users k and m in a pairing π. Since $P_{suc}(\pi) = \prod_{i=1}^{N} P_{suc,i}(\pi)$, it is feasible to transform the problem of maximization of $P_{suc}(\pi)$ into the maximization of $\log_2 P_{suc}(\pi)$ such that

$$\begin{aligned} \pi_T^* &= \arg\max_{\pi} \log_2 P_{suc}(\pi) \\ &= \arg\max_{\pi} \sum_{i=1}^{N} \log_2 P_{suc,i}(\pi) \end{aligned} \tag{3.13}$$

where π_T^* is the optimal pairing according to the new optimization problem. As seen from equation (3.13), the problem is now transformed to that of finding the optimal pairing that maximizes the sum of the logarithms of the error-free transmission probabilities of each user. Fortunately, this new optimization problem can be solved in polynomial time owing to the claw-freeness of the user pairing conflict graph as explained in next subsections. Notice that a target rate dependent user pairing strategy is employed, i.e., user pairings change with respect to the target rates of users.

In the following subsections, the necessary definitions are provided and the graph-theoretical solution is explained in detail.

3.4.1. Construction of the Conflict Graph

Definition 3.1. (Conflict graph) *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a conflict graph if there is an edge between two vertices when at most one of the variables represented by the vertices can equal one in an optimal solution [176].*

Definition 3.2. (Claw-free graph) *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is claw-free if none of its vertices has three pairwise non-adjacent neighbors [177].* An example claw may be seen in Fig. 3.1.

Definition 3.3. (Independent Set) *Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a subset of vertices $\mathcal{S} \subseteq \mathcal{V}$ is an independent set if $\forall i, j \in \mathcal{S}$, $\{i, j\} \notin \mathcal{E}$ is satisfied [178].*

Definition 3.4. (Independence Number) *Independence number of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the size of the largest independent set in the graph [179].*

Definition 3.5. (Maximum Weighted Independent Set) *Given an undirected and weighted graph $\mathcal{G} = (\mathcal{V}, w_{\mathcal{V}}, \mathcal{E})$, where the weights are assigned on vertices $w_{\mathcal{V}} : \mathcal{V} \to \mathbb{R}$, a subset of vertices $\mathcal{S} \subseteq \mathcal{V}$ constitute a maximum weighted independent set if $\forall i, j \in \mathcal{S}$, $\{i, j\} \notin \mathcal{E}$ and there is no other independent set having a greater weight sum $\sum_{i \in \mathcal{S}} w_i$ [178].*

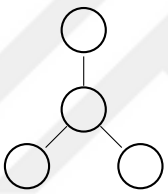


Figure 3.1. A claw.

The user pairing conflict graph is the graph of possible pairings among users. In the construction of the conflict graph $\mathcal{G} = (\mathcal{V}, w_{\mathcal{V}}, \mathcal{E})$, $\mathcal{V}$ is the set of possible pairings, $w_{\mathcal{V}}$ is the set of weights belonging to $\mathcal{V}$ and $\mathcal{E}$ is the set of conflicts. The vertices, weights and edges of the conflict graph are found step-by-step as follows:

1) $\mathcal{V}$: Given that there are N users, indexed from 1 to N, in a cellular network, each vertex v of the conflict graph corresponds to a pair of users (k, m), where $1 \leq k < N$ and $k < m \leq N$. There must be $N(N-1)/2$ vertices in total.

2) $w_{\mathcal{V}}$: The weight of a vertex is defined as the log of the error-free transmission probability of BS to that pair, i.e. $\log_2 P_{suc,k,m}$. Note that $\log_2 P_{suc,k,m} = \log_2 P_{suc,k,m}(\pi)$, i.e. $\log_2 P_{suc,k,m}$ does not depend on overall pairing scheme since pairing of users k and m is given and no further pairing information is needed to calculate

the combined error-free transmission probability. The unprocessed weight of the vertex containing the pairing (k, m), $w^u_{(k,m)}$, is calculated as

$$\mathrm{w}^u_{(k,m)} = \log_2 P_{suc,k,m} = \log_2 P_{suc,k} + \log_2 P_{suc,m} \tag{3.14}$$

assuming that they are NOMA pairs. Then the weight is specified as

$$\mathrm{w}_{(k,m)} = w^u_{(k,m)} + |\min_{i,j} w_{(i,j)}| + \epsilon \tag{3.15}$$

in order to guarantee the numerical stability of the MWIS algorithm where $\epsilon > 0$ and $k \in \{1, 2, ..., N-1\}$, $m \in \{k+1, k+2, ..., N\}$, $i \in \{1, 2, ..., N-1\}$, $j \in \{i+1, i+2, ..., N\}$. This modification in vertices' weights are done in order to make all weights positive and this in turn helps to choose as many vertices as possible in the MWIS algorithm.

3) $\mathcal{E}$: The vertex containing (k, m) pair must have edges with all vertices containing $(k, .)$ and $(m, .)$ pairings due to the fact that a user cannot be paired with two different users simultaneously. As a note, a self-loop, an edge between a vertex and itself, is not meaningful and not present in the conflict graph, i.e, $\forall v \in \mathcal{V}$, $\{v, v\} \notin \mathcal{E}$.

An example of user pairing conflict graph for a 4-user NOMA scheme can be seen in Fig. 3.2. Notice that this construction turns the error-free transmission probability maximizing user pairing problem into that of finding the MWIS over a conflict graph.

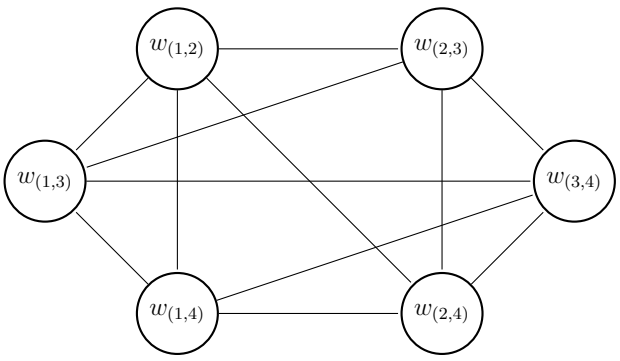


Figure 3.2. User pairing conflict graph for a 4-user NOMA scheme.

3.4.2. The MWIS Solution

Given the construction of the conflict graph as explained in Section 3.4.1, the following theorem can be used to get the optimal solution.

Theorem 3.1. *Given N users and $\frac{N}{2}$ resources in a NOMA system, i) the proposed conflict graph is claw-free, ii) all users are guaranteed to be paired in the MWIS solution, and iii) the overall transmission success probability maximizing user pairing is found in polynomial time.*

Proof. i) Claw-freeness: In the conflict graph, an arbitrarily chosen vertex (k, m) must be connected to all vertices in the set $(k, .)$, which constitute the set of all possible pairings of k with other users, since if k is paired with m, then any other pairing is not possible for k. Since all vertices in the set $(k, .)$ are connected to each other leading to a complete graph, the independence number of the set $(k, .)$ is 1. The same logic applies to the set $(m, .)$ leading to a total independence number of 2 for the sub-graph of vertices connected to the central vertex (k, m). A third independent vertex is not available to set up a claw. Therefore, the user pairing conflict graph is claw-free.

ii) Pairing of all users: Consider the case that the output of the MWIS algorithm has a cardinality $|\mathcal{W}|$ and $|\mathcal{W}| \geq N/2 + 1$. This is not valid since there are no $N + 2$ or more users in the system. Considering the case $|\mathcal{W}| \leq N/2 - 1$, we have $N - 2$ or less users in the final set, however such sets cannot be the ones having the maximum weight since it is always possible to add at least one more vertex to get an independent set having a greater weight. Therefore, $|\mathcal{W}| = N/2$ is satisfied and no user remains unpaired in the solution of the MWIS algorithm.

iii) Conclusion of the proof: The user pairing conflict graph is claw-free, hence the MWIS can be found in polynomial time guaranteeing all users being paired. Since it is possible to achieve the maximum overall transmission success probability when the MWIS is found, the AoI minimizing pairing can be obtained in polynomial time. □

3.4.2.1. The Computational Complexity. The overall computational complexity of the graph-theoretical method is dominated by the MWIS algorithm given in [173], which has a complexity of $\mathcal{O}(|\mathcal{V}|(|\mathcal{E}| + |\mathcal{V}|log|\mathcal{V}|))$. Since $|\mathcal{V}| = N(N-1)/2$ and $|\mathcal{E}| = N(N-1)(N-2)/2$ in the conflict graph, the overall complexity of user pairing is given by $\mathcal{O}(N^5)$, i.e., a polynomial time complexity.

3.5. Simulation Results

The simulation results have been obtained by randomly distributing N users in a cell having $r = 1$ km radius and repeated for 500 times for each γ and N for $T = 1000$ time slots to obtain the average AoIs where user target rates are specified as the user rates achieved by NOMA. Two different simulation results are given in Fig. 3.3 and 3.4 where we compare the average AoI performances of the standard OMA setup, proposed graph-theoretical pairing strategy used in NOMA and the benchmarks obtained from some recent NOMA user pairing research papers seen in [170, 172]. In the figures, Pairing-1 and Pairing-2 denote the benchmark user pairing approaches proposed in [170, 172], respectively.

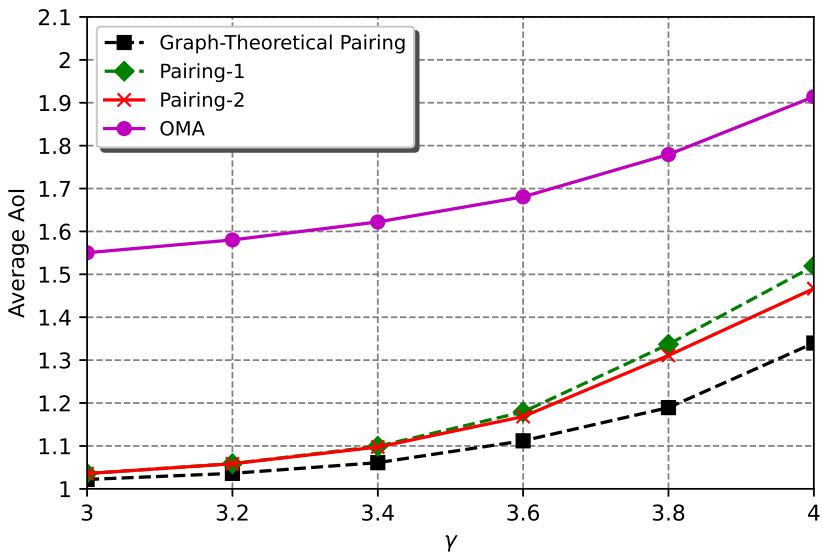


Figure 3.3. Average AoI comparison of pairing strategies ($N = 10$).

In Fig. 3.3, the average AoI performances are given where path-loss exponent γ values are chosen between 3 and 4, which model typical environments ranging from having better line-of-sight (LOS) to relatively lossy ones for suburban areas as may be seen in [180]. The proposed graph-theoretical method clearly outperforms the other approaches, by a greater performance gain with increasing path-loss exponent γ. For $\gamma = 4$, an approximately 10% lower average AoI is observed compared to the best of the benchmark approaches.

In Fig. 3.4, the average AoIs are evaluated with respect to N, the number of users in the network. Path-loss exponent γ has been chosen as 3.6 for this figure since greater values lead to a tendency in AoI values to oscillate more for high number of users. The proposed graph-theoretical strategy outperforms the benchmarks introduced in Section 3.2 where the average AoI of our method is approximately 5% lower than the closest rival's.

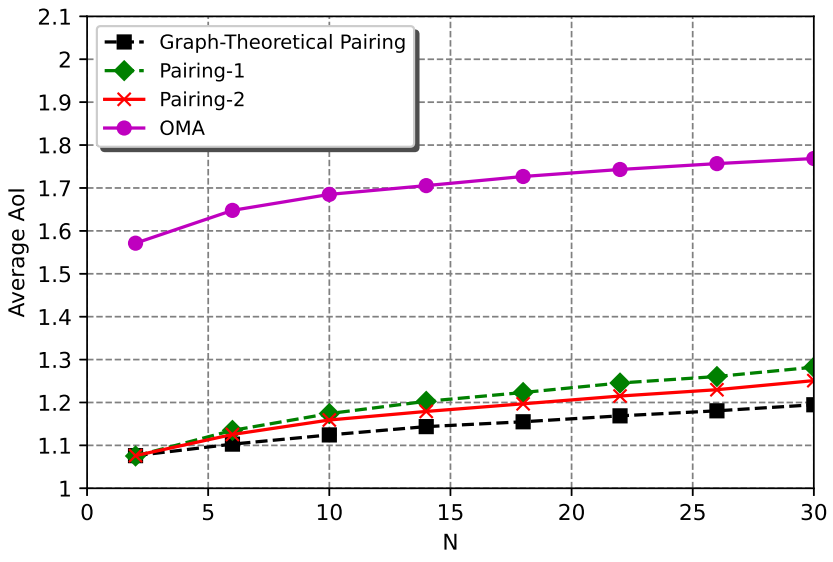


Figure 3.4. Average AoI comparison of pairing strategies ($\gamma = 3.6$).

The computational complexities of both benchmark approaches [170, 172] are dominated by sorting algorithms, with a time complexity of $\mathcal{O}(N \log N)$ using Merge sort. In contrast, the proposed algorithm has a significantly higher complexity of

$\mathcal{O}(N^5)$. This creates a trade-off between AoI and computational complexity. If a network has a large number of users and experiences high mobility, necessitating frequent updates of user pairings, the proposed solution may not be feasible. Otherwise, opting for the proposed strategy is advised.

Lastly, note that the lowest AoI that OMA can achieve is 1.5 where NOMA has a far greater potential as it can achieve 1. The gain of NOMA over OMA is splendid due to the scarcity of orthogonal resources to support all users in the same time slot for OMA.

3.6. Conclusion

In this chapter, we have considered the user pairing problem in a downlink NOMA system with a goal of minimizing the average AoI in a cellular network. To this end, a graph-theoretical dynamic pairing strategy has been proposed depending on the user rate requirements where the dependency comes from the weighting scheme in the conflict graph which, in turn, comes from the calculated outage probabilities of the users supported by NOMA scheme. The proposed method includes a transformation of the original problem to a manageable counterpart in a conflict graph and a polynomial time solution, as a result of the claw-freeeness of the conflict graph, is attainable. In the end, the simulation results show the superiority of the proposed method compared to benchmarks.

4. A NOVEL SCHEDULING STRATEGY IN PRIORITY-AWARE IOT NETWORKS FOR AGE OF INFORMATION OPTIMIZATION

4.1. Introduction

The problem of scheduling is inherently a decision-making problem as shown in some related works such as [181–183]. In [181, 182], multi-armed bandits (MAB) formulation has been employed to handle the presence of multiple actions and uncertain rewards, allowing for efficient information gathering techniques to minimize the disparity between AoI and PAoI limits. In [183, 184], on the other hand, Whittle's index, a well-known technique for solving the restless MAB problem, has been employed because it has been shown to provide an asymptotically optimal solution to the average AoI minimization problem. However, both MAB and Whittle's index policies, fall short to capture the dynamics of the scheduling problem for the following reasons. Firstly, the MAB approach aims to strike a balance between exploration and exploitation for scheduling sensors. However, it is possible to find the transmission success probabilities accurately in some scenarios as will be shown in Section 4.2, without the need for exploration steps. Secondly, the Whittle's index policy is specifically designed to minimize the average AoI, where in priority-aware IoT networks the aim usually is to minimize the number of sensors violating their PAoI limits. In other words, the prioritization issue of sensors, which is the main theme of this work, is not addressed by Whittle's index.

A more sophisticated algorithm is required to satisfy the requirements of priority-aware IoT systems under various power and frequency resource constraints. The proposed solution is based on solving the $0-1$ knapsack problem, which is NP-hard, introduced in [185] and involves selecting a subset of items with maximum value while adhering to the weight constraints. It has various applications, including resource allocation in D2D communications [186] and scheduling in Long Term Evolution (LTE)

networks [187]. The most commonly used algorithm to tackle the knapsack problem is dynamic programming, with a pseudo-polynomial time complexity of $O(NW)$, where N is the number of items and W is the maximum weight capacity of the knapsack.

The primary contribution of this chapter is to introduce a novel scheduling algorithm designed to minimize AoI violations associated with sensors in IoT networks. Specifically, we have identified that the well-established knapsack approach, when appropriately customized, effectively caters to the needs of the defined problem. This strategic adaptation holds the potential to enhance QoS within IoT networks, particularly in domains like autonomous vehicles, remote surgery and collaborative mobile robotics.

The remainder of the chapter is structured as follows: In Section 4.2, the system model and problem formulation are presented. The proposed method and the corresponding benchmarks are explained in Section 4.3 and 4.4. In Section 4.5, simulation results and commentary discussions are presented, while Section 4.6 contains the conclusion.

4.2. System Model and Problem Formulation

We consider an IoT network where N sensors continuously generate status updates and send them to a central node S. The update packets are assumed to be immediately generated when a sensor has been scheduled to transmit. Due to the number of available frequency bands, not all N sensors are allowed to transmit status updates simultaneously. Each sensor has a different battery specification so that each of them has different peak and average transmission power constraints. A simplified path-loss model is adopted as in [188] given by

$$\mathcal{P}_S = \mathcal{P}_i K \left[\frac{d_0}{d_i}\right]^{\gamma} \tag{4.1}$$

where $\mathcal{P}_S$ is the received power at the central node, $\mathcal{P}_i$ is the transmit power of sensor i, K is a constant depending on the antenna characteristics, d_0 is a reference distance for the antenna far field, γ is the path-loss exponent and d_i is the distance between the

sensor i and the central node S. On the other hand, the distribution of the transmit power to received power $\phi = \mathcal{P}_i/\mathcal{P}_S$, with the log-normal shadowing also considered, as in [188], is given as

$$p(\phi) = \frac{10 \ln 10}{\sqrt{2\pi}\sigma_{\phi_{dB}}\phi} exp\left[-\frac{(10\log_{10}\phi - \mu_{\phi_{dB}})^2}{2\sigma^2_{\phi_{dB}}}\right] \tag{4.2}$$

where $\mu_{\phi_{dB}}$ is the mean of $\phi_{dB} = 10log_{10}\phi$ in dB and $\sigma_{\phi_{dB}}$ is the standard deviation of ϕ_{dB}. Omitting the extensive calculations and explanations which are available in [188], the outage probability of the central node S with respect to the transmission of sensor i is denoted as p_i^{out}. This probability, defined as the likelihood that the received power at S falls below $\mathcal{P}_{min}$, is determined as

$$p_i^{out} = 1 - Q\left(\frac{\mathcal{P}_{min} - (\mathcal{P}_i + 10\log_{10}K - 10\gamma\log_{10}(d_i/d_0))}{\sigma_{\phi_{dB}}}\right) \tag{4.3}$$

under the combined path-loss and shadowing model where $Q(\cdot)$ denotes the well-known Q-function. In the subsequent sections, we demonstrate how the outage probability, derived from the path loss model, is customized for the purpose of scheduling sensors within the system. In addition to the outage probability, several other factors, including the chosen AoI model, frequency band and battery constraints, play a pivotal role in determining which sensors are to be scheduled.

Employing a discrete count model, the AoI is measured as

$$\Delta_i(t) = \begin{cases} \Delta_i(t-1) + 1, & \text{if } r_i(t-1) = 0, \\ 1, & \text{if } r_i(t-1) = 1. \end{cases} \tag{4.4}$$

$\Delta_i(t)$ is the AoI of sensor i measured in the central node S at time t. $r_i(t) = 1$ denotes the presence of a successfully sent packet for sensor i at time t and $r_i(t) = 0$ denotes its absence. In other words, the AoI for a sensor becomes 1 when it successfully sends a status update to the central node S with probability of success $1 - p_i^{out}$ given sensor i is scheduled, and it increments by 1 in each time slot if it does not send any update or the central node experiences an outage with probability p_i^{out}.

The main objective is to minimize the total number of sensors whose AoI values exceed their PAoI limits and the difference between AoI values and PAoI limits of these

sensors in order to guarantee the QoS in time-sensitive prioritization important IoT networks. The corresponding optimization problem can formally be defined as follows

$$\min \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \mathbb{1} \left\{ \Delta_i(t) > \Delta_i^{\max} \right\}, \tag{4.5a}$$

$$\min \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \left\{ \Delta_i(t) - \Delta_i^{\max} \right\} \forall i \in M, \tag{4.5b}$$

$$\text{s.t.} \sum_{i\in U} w_i \leq W, \tag{4.5c}$$

$$\mathcal{P}_i(t) \leq \mathcal{P}_i^{\text{peak}}, \forall t, \forall i \in U, \tag{4.5d}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathcal{P}_i(t) \leq \mathcal{P}_i^{avg}, \forall i \in U, \tag{4.5e}$$

where $\mathbb{1}$ is the indicator function, $M = \{i | \Delta_i(t) > \Delta_i^{\max}\}$, U is the subset of the scheduled sensors and $\Delta_i^{\max}$ is the PAoI limit of the sensor i. The number of available frequency bands in the network is denoted as W and constrained in inequality (4.5c) where w_i represents the number of frequency bands required by sensor i to transmit its update. Furthermore, the limited battery resources restrain the peak and average transmission powers of sensors, the corresponding constraints are provided in inequalities (4.5d) and (4.5e), respectively.

As can be seen in the next section, each sensor is assigned a reward value, which is determined by a function of excess AoI over its limit per time slot, i.e., the scheduling is mainly done prioritizing sensors with the highest exceeded AoI levels, being fair among all, not always favoring a group of sensors, and will optimize (4b) as much as (4a). So, these two objectives are heuristically highly correlated even if they do not always have the same optimal solution. In addition, there is a restricted number of available frequency bands, denoted as weights. This situation closely resembles the well-established knapsack problem. In the following section, we introduce an adaptation of the widely recognized dynamic programming algorithm for the knapsack problem to address the considered problem.

4.3. Knapsack Strategy

In general, knapsack problem can be expressed simply as

$$\text{maximize } \sum v_i \text{ subject to } \sum w_i \leq W \text{ such that } i \in N, \tag{4.6}$$

where N represents the set of elements, with each element i has a corresponding value v_i and weight w_i, and W denotes the maximum weight limit. The goal is to maximize the total value without exceeding the weight limit.

In the adapted model, v_i represents the reward associated with sensor i in the relevant time slot. The reward of the sensor i increases over time as long as it remains unscheduled or makes unsuccessful transmissions. As a result, the algorithm demonstrates a preference for sensors which exhibit a growing AoI over time. In this context, the weight w_i corresponds to the number of frequency bands required by sensor i to initiate an update transmission during a certain time slot. The sum of w_i, $i \in U$, is constrained by the total number of available frequency bands denoted as W. The values of v_i are adjusted using a modified version of the penalty function derived from [189], which provides diverse penalty functions for AoI. In the proposed model, the exponential penalty function is chosen as

$$v_i(t) = e^{\Delta_i(t) - \Delta_i^{max}}. \tag{4.7}$$

This function enables the exponential growth of reward values for each sensor i during the time slots in which a successful update transmission does not occur. This accounts for the prioritized scheduling of the sensors whose AoI exceeds their PAoI limits, ensuring prompt attention to such sensors.

It is important to note that, prior to initiating a sensor's transmission, the estimation of the outage probability, as described in (4.3), is used to make an informed decision regarding the potential success or failure. Both average power and peak power constraint are also considered. The average power constraint is enforced by monitoring the power levels using a sliding time window.

The width of this sliding window is specified similar to given in [190] and can be mathematically determined as

$$\tau(t, \alpha) = \lceil \min \{t, \lambda t^{\alpha}\} \rceil. \tag{4.8}$$

The width of the sliding window can be tuned with the parameters λ and α based on the environment characteristics where $\alpha \in (0, 1]$.

Algorithm 1 Knapsack dynamic programming algorithm

0: **for** $t \in T$ **do**
0: Find all sensors $i \in U$ with backtracking by knapsack dynamic approach
0: **if** $p_{discrete}^{out} = 0$ for all sensor $i \in U$ **then**
0: Keep all sensors $i \in U$ as chosen
0: **else**
0: **while** $p_{discrete}^{out} \neq 0$ for all sensors $i \in U$ **do**
0: **for** $i \in U$ **do**
0: **if** $p_{discrete}^{out} = 0$ **then**
0: Keep the sensor i
0: **else**
0: Find $P_i^{required}$ to turn $p_{discrete}^{out} = 0$
0: **if** $P_{required} > P_i^{peak}$ or $\frac{1}{\tau}\sum_{t=1}^{\tau} \mathcal{P}_i(t) > P_i^{avg}$ **then**
0: Leave the sensor i and find another subset U without the sensor i by knapsack dynamic approach
0: **else**
0: Keep sensor i and set $P_i^{required}$ as P_i
0: **end if**
0: **end if**
0: **end for**
0: **end while**
0: **end if**
0: Update Δ_i's
0: **end for**=0

After finding the set U, in each time slot, for every sensor $i \in U$, the power required ($P_i^{required}$) to converge p_i^{out} to zero is computed. If $P_i^{required} > P_i^{peak}$ or $\frac{1}{\tau}\sum_{t=1}^{\tau}\mathcal{P}_i(t) > P_i^{avg}$, then sensor i is assumed to be unavailable for transmission, $i \notin U$, and is replaced with the most appropriate sensor that satisfies the constraints. The extended algorithm is given in Algorithm 1. Note that $p_{discrete}^{out} = \text{binornd}(1, p_i^{out})$ where binornd(1,p) is a Matlab function which generates a random number $\{0, 1\}$ from the binomial distribution having success probability p. The time complexity of Algorithm 1 is O(NW), N being the size of the input (number of sensors) and W being the number of available frequency bands, leading to a pseudo-polynomial time complexity.

An alternative approach involves the use of the fully polynomial time approximation scheme (FPTAS) coupled with the dynamic programming algorithm. FPTAS is employed when generating an exact solution to an optimization problem is computationally complex, but an approximate solution is deemed acceptable. It runs in polynomial time relative to both the input size and the inverse of the desired accuracy factor. The FPTAS assignment scheme produces a solution within a factor of $(1+\epsilon)$ of the original solution, where $0 < \epsilon < 1$ is provided as an input to the algorithm. The algorithm's time complexity becomes polynomial in N as O(N^3/ϵ) [191]. The main solution remains identical to Algorithm 1, with the sole distinction that v_is are assigned as in Algorithm 2. However, the FPTAS assignment scheme may not necessarily improve the time complexity, it does depend on the parameters of the considered problem.

Algorithm 2 FPTAS assignment scheme for knapsack algorithm

0: P = max (v_i) for $i = 1, 2, \ldots, N$

0: $K := \epsilon\frac{P}{N}$

0: **for** $i = 1, 2, \ldots, N$ **do**

0: $\quad v_i' := \left\lfloor \frac{v_i}{K} \right\rfloor$

0: **end for**=0

4.4. Benchmarks

4.4.1. Epsilon-Greedy Algorithm

Scheduling IoT network sensors resembles a MAB problem with power and frequency constraints. The epsilon-greedy algorithm achieves this balance by initially exploring to learn rewards before exploiting the best choice. In this customized approach, multiple sensors are selected without collisions, considering power and frequency band constraints for each sensor individually.

The value is denoted as $Q_t(s_t, a_t)$ where t denotes the current time slot, s_t is the set of scheduled sensors at $t-1$ and a_t is the set of sensors to be scheduled in time slot t. In other words, $Q_t(s_t, a_t)$ is the value function that represents the reward of choosing option a_t at time t where the set of scheduled sensors at time $t-1$ is s_t. Notice that the value function for Q-learning denoted as $Q_t(\cdot, \cdot)$ is different from the tail distribution function of the standard normal distribution denoted by $Q(\cdot)$ in Section 4.2.

The values of Q matrix are calculated by

$$\begin{aligned} Q_t\left(s_t, a_t\right) = & Q_{t-1}\left(s_t, a_t\right) \\ & + \alpha\left(r_t + \gamma \max_{a'} Q_t\left(s'_{t+1}, a'\right) - Q_{t-1}\left(s_t, a_t\right)\right) \end{aligned} \tag{4.9}$$

where α denotes the learning rate, which determines the degree of the acceptance of new values over past ones and γ denotes the trade-off between the present and future rewards. The reward refers to the gain obtained by selecting the option a_t.

Non-stationary rewards are considered as in (4.7). The sensors leading to the maximum value of Q_t can be chosen as

$$a = \arg\max_{a} Q_t\left(s_t, a\right). \tag{4.10}$$

The workflow is modified similar to Algorithm 1, wherein the collisions are prevented and the selection of sensors is based on epsilon-greedy logic instead of knapsack problem solution. The computation of the outage probability involves the utilization of the

binomial distribution function, while the monitoring of average and peak power is accomplished through a sliding window mechanism as explained in the previous section. The width of this sliding window is determined based on the adapted algorithm, which takes into account the characteristics of the environment and tuning parameters.

4.4.2. Whittle's Index Algorithm

Another benchmark algorithm is based on Whittle's index, which has been proven to be asymptotically optimal in terms of average AoI according to [183]. Each sensor's index captures both the reward associated with it and the level of uncertainty. Sensors with the highest indices are selected for scheduling. The Whittle's index is defined as

$$W(h) = hf(h+1) - \sum_{j=1}^{h} f(j) \tag{4.11}$$

where $W(h)$ represents the Whittle index in state h and f is the cost function as derived in [184]. In this setup, f denotes the reward function given by equation (4.7). In the benchmark, the sensors with the highest Whittle's Indices are selected in a descending order, ensuring the sum of the frequency bands they require satisfies $\sum_{i \in U} w_i \leq W$.

This work omits two optimization methods from benchmarking: the optimal stopping problem model [192], which maximizes rewards by optimizing selection time, and the search model [193], which optimizes selection among options with random rewards. These models were excluded because the proposed model's reward growth is exponential until sensor selection, making them unsuitable benchmarks. Additionally, the Gittins index [194], which calculates expected rewards over all states, was excluded due to its computational complexity.

4.5. Simulation Results

The simulations are performed for a total of $T = 10000$ iterations, considering a set of $N = 100$ users. The number of available frequency bands varies within the range of $W \in [25, 245]$ and is represented as an integer value by noting that the widths

of the frequency bands vary among networks. For each user, the required number of frequency bands for transmission, denoted as $w_i \in [1, 25]$, is sampled uniformly random as an integer value in the given interval. The values chosen for the parameters are $\lambda = 12.3$ and $\alpha = 4 \times 10^{-4}$, which are utilized in the computation of the sliding window's width. AoI values of each sensor are initialized with 1. The PAoI limits of sensors, $\Delta_i^{max} \in [5, 8]$, are sampled uniformly random in the specified interval. Average power is chosen as 43 dB and peak power is given as 1.2 times this value. In real time applications, all these parameters can be set by extracting the needs of each sensor.

Fig. 4.1 illustrates the number of sensors surpassing their PAoI limits. The dynamic programming approach for the knapsack model demonstrates a 5.5% improvement over the Whittle's index and a 3.82-fold improvement over the epsilon-greedy model on average, specifically for $W > 140$. This improvement signifies a reduction in the number of sensors exceeding PAoI limits, thereby enhancing the QoS in the network. The most striking outcome is that none of the sensors within the network violate their PAoI limits after $W > 250$ when the proposed approach is employed for scheduling.

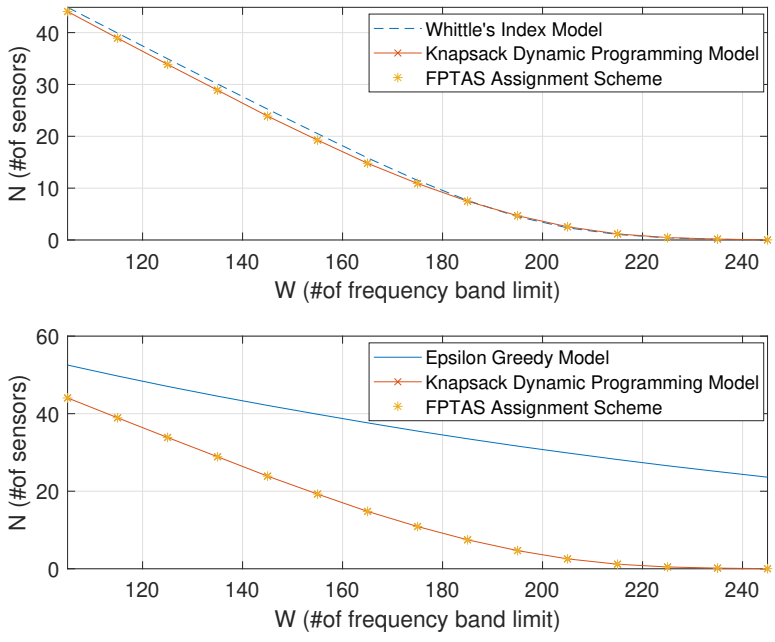


Figure 4.1. Average number of sensors exceeding their PAoI limits.

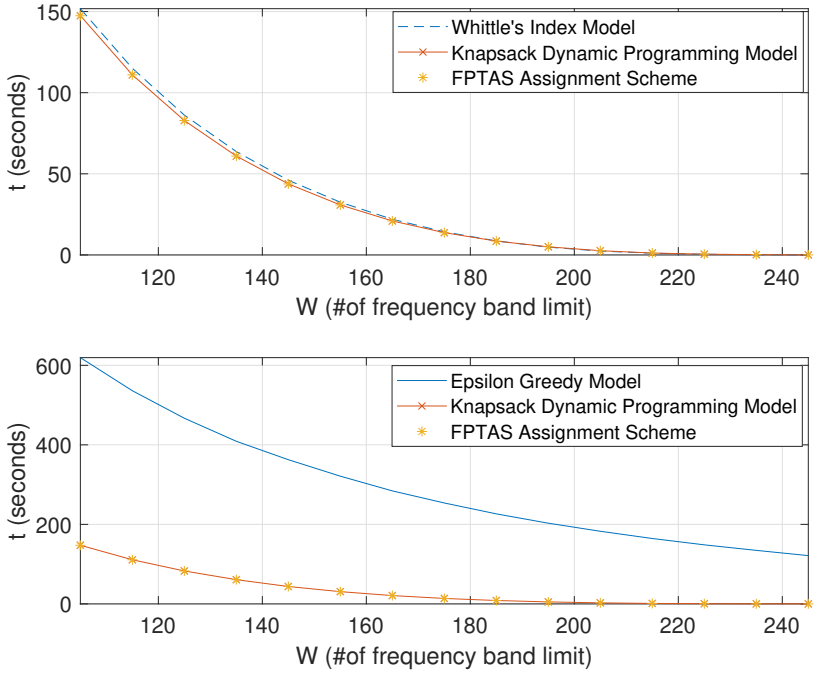


Figure 4.2. Average sum of AOIs exceeding PAoI limits.

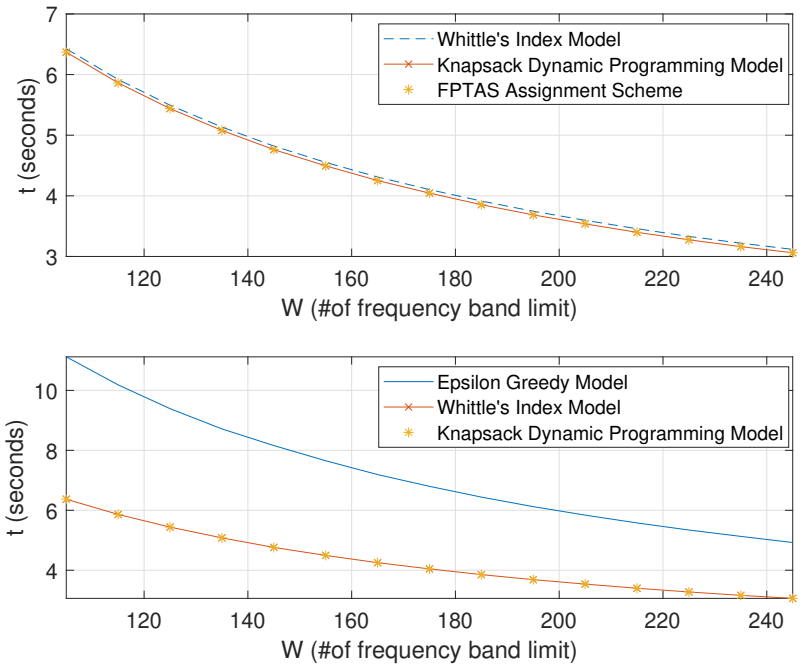


Figure 4.3. Average AOIs.

Additionally, the total sum of AoI values exceeding PAoI limits serves as another QoS metric. Fig. 4.2 presents the comparison, where the dynamic knapsack algorithm outperforms Whittle's index scheduling by 5.3% and the epsilon-greedy model by 17.32 times in average for $W > 140$. Furthermore, Fig. 4.3 showcases the average AoI comparison between the dynamic knapsack approach, Whittle's index, and the epsilon-greedy models. For $W > 140$, the knapsack model outperforms the epsilon-greedy model by 66.3% and Whittle's index by 1.6% in terms of average AoI. Despite Whittle's index being asymptotically optimal for average AoI, the knapsack solution enhances AoI-related metrics without any loss in average AoI.

The FPTAS performs similarly to dynamic programming but excels in reducing time complexity and improving running time while maintaining comparable QoS. It transforms the original pseudo-polynomial time complexity problem to a polynomial time complexity one with input size N, from $O(NW)$ to $O(N^3/\epsilon)$. On the other hand, Whittle's Index is calculated for N sensors and there are h addition operations for each calculation then the overall complexity becomes $O(Nh)$ where h is the lookback period specified in (4.11).

In summary, the dynamic programming approach with FPTAS outperforms Whittle's index scheduling in key metrics. It demonstrates a substantial 5.5% reduction in the number of sensors exceeding PAoI limits, a significant 5.3% decrease in the total sum of AoI values exceeding PAoI limits, and 1.6% advantage over Whittle's index in terms of average AoI, where Whittle's index is considered asymptotically optimal. The results, coupled with complexity analyses, establish that the proposed approach is the best choice for small to moderate IoT networks and systems with a large lookback period T. Conversely, in very large networks with limited computational resources, the proposed method may become impractical, making Whittle's index scheduling a more viable choice.

4.6. Conclusion

Satisfying PAoI limits for certain sensor nodes is crucial in dynamic environments where up-to-date information is essential, ensuring timely decision-making, optimizing performance, enhancing user experience and preventing accidents. For this reason, a priority-aware age of information optimization problem has been proposed and reformulated as a knapsack problem where it has been shown that the dynamic programming algorithm outperforms epsilon-greedy and Whittle's index algorithms in terms of reducing the number of sensors whose AoI exceeds the PAoI constraint, while also minimizing the difference between their AoIs and the PAoI constraints in a resource-constrained environment.

5. IMPACT OF BLOCK CODING ON AGE OF INFORMATION

5.1. Introduction

In this work, we delve into quantifying and comparing the AoI bounds associated with block coding and, as a subcase, polar coding as polar codes are esteemed in next-generation communication systems for their capacity-achieving properties, efficient algorithms, and hardware-friendly nature, finding adoption in standards like 5G NR. Discrete memoryless BSC and BEC are considered in an IoT network and evaluations focus on the coding schemes rather than transmissions strategies which is another task addressed in [32]. The BSC and BEC serve as valuable tools for grasping the fundamentals of channel coding as many IoT communication systems rely on wireless technologies prone to issues like packet loss and interference, akin to the behaviors of BSC and BEC. Owing to its optimality, maximum-likelihood decoding (MLD) error bound provided in [195] is employed even though the complexity of MLD grows exponentially with the block-length of the code, leading to an NP-hard problem to solve as shown in [196]. As a practically realizable block coding scheme, polar coding with successive-cancellation (SC) decoding introduced in [197] is considered. While SC decoding may not fall under the category of MLD, it is an essential part of the discussion as it represents a prominent decoding technique employed in practical implementations of polar codes. Thus, this helps us to compare the theoretical AoI limits of block codes to a practical application of it which is polar coding with SC decoding to observe the gap between the theoretical limit of a wider set (block coding with MLD) and the practical limit of a scheme inside this set (polar coding with SC decoding). The key contribution of this chapter is twofold: i) the general lower bound and upper bounds on the average AoI in the IoT network are derived for MLD in the general class of block codes and SC decoding in polar codes, and ii) AoI gap of SC decoded polar code and MLD block code with no coding scenarios, are evaluated with respect to changes in IoT network parameters.

The rest of the chapter is organized as follows. The system model is presented in Section 5.2. Block decoding error probabilities and related AoI expressions are derived in Sections 5.3 and 5.4. Numerical experiments are illustrated in Section 5.5 before the concluding remarks in Section 5.6.

5.2. System Model

We consider an IoT network where N sensors continuously generate status updates and send them to a central node S where status updates are integrated to be evaluated. The status update of each sensor is l_i bits, $i \in \{1, 2, ..., N\}$ where l_i does not change since the contents of the updates are pre-defined and do not vary. Update packets are assumed to be immediately generated when a sensor has been scheduled to transmit. Due to scarcity of resources, M of N sensors are allowed to transmit status updates simultaneously to central node S, being $M < N$. The transmissions are done such that the generated status updates are encoded at the transmitter sensor using block coding and sent through discrete memoryless BSC or BEC. BSC with a crossover probability ϵ is denoted as BSC(ϵ). The input X and output alphabets Y are $X = Y = \{0, 1\}$ and the conditional channel probability can be written as

$$P_{Y|X}(y|x) = \begin{cases} 1-\epsilon, & y = x, \\ \epsilon, & y \neq x. \end{cases} \tag{5.1}$$

BEC with an erasure probability γ is denoted as BEC(γ). The input X and output alphabets Y are $X = \{0, 1\}$ and $Y = \{0, 1, e\}$ where the conditional channel probability is mathematically described as

$$P_{Y|X}(y|x) = \begin{cases} 1-\gamma, & y = x, \\ \gamma, & y = e, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

Sensor i needs a transmit time of $t_i = l_i/R_i$, where R_i denotes the transmission rate of sensor i, in order to complete its update transmission to the central unit. Each status update is assumed to be correctly decoded if all the corresponding code block transmissions are correctly decoded. Noting that erroneous decoding of the incoming

signals at the central entity will result in an understated average AoI, only correctly decoded packets are considered to calculate the AoI values in the central node. The main objective is to guarantee that S receives the freshest data by minimizing the average AoI which is continuously measured at S. Therefore, for the scheduling task to decide which M users should transmit at any moment, Whittle's index policy has been utilized since it has been proved to be asymptotically optimal in terms of average AoI in [183]. It is a decision-making algorithm which dynamically assigns index values to actions based on their expected reward and cost, striking a balance between exploration and exploitation to optimize resource allocation.

A continuous AoI model for each sensor i, $i \in \{1, 2, ..., N\}$ at the central node S has been employed. The average AoI, which is denoted as $\bar{\Delta}$, at the central node can mathematically be written as

$$\bar{\Delta} = \frac{1}{N} \sum_{i=1}^{N} \limsup_{T \to \infty} \frac{1}{T} \int_{t=0}^{T} \Delta_i(t). \tag{5.3}$$

In the next section, the block-error probabilities are derived for block coding with MLD and polar coding with SC decoding as those expressions are required to finalize AoI analysis.

5.3. Block-Error Probability Derivations

5.3.1. Block-Error Probability for Block Codes

Theorem 5.1. *In [195], it is shown that, for any discrete memoryless point-to-point channel with transition probability $P_{Y|X}(y|x)$, the block coding block-error probability under maximum-likelihood decoding becomes*

$$P_e \leq e^{-N_b \frac{(C-R_c)^2}{8/e^2 + 2(\log_e |\mathcal{Y}|)^2 - 2R_{cr}^2}} \tag{5.4}$$

where N_b is the block-length, R_c is the coding rate, C represents the channel capacity, $|\mathcal{Y}|$ is the size of the output alphabet and $R_c \in [\max\{0, C - (\frac{4}{e^2} + \log_e^2 |\mathcal{Y}| - R_{cr}^2)\}, C]$. Here, $R_{cr} \triangleq E_0'(1, P_X)$ is the critical rate defined in Eq. (5.6.30) of [198] where P_X

stands for input distribution and $E_0'(1, P_X)$ represents the derivative of $E_0(\rho, P_X)$ with respect to ρ at $\rho = 1$. $E_0(\rho, P_X)$ is given as

$$E_0(\rho, P_X) \triangleq -log_e \sum_y \left[\sum_x P_X(x) P_{Y|X}(y|x)^{1/(1+\rho)} \right]^{1+\rho} \tag{5.5}$$

and is employed in the calculation of the upper bound for the ensemble average probability of decoding error using MLD.

Proof. The derivations can be found in [198] and [195]. □

Corollary 5.2. *In the presence of block coding in a BSC(ϵ), the block-error probability under MLD is expected to be upper-bounded by*

$$P_e \leq e^{-N_b \psi} \tag{5.6}$$

where ψ is given as

$$\psi = \frac{(R_c + (\epsilon - 1)\log_e(1-\epsilon) - \epsilon \log_e(\epsilon) - \log_e 2)^2}{-2\omega^2 + \frac{8}{e^2} + 2\log_e^2(2)} \tag{5.7}$$

and ω is given as

$$\omega = \log_e\left(\sqrt{1-\epsilon} + \sqrt{\epsilon}\right) - \frac{\sqrt{1-\epsilon}\log_e(4-4\epsilon) + \sqrt{\epsilon}\log_e(4\epsilon)}{2\left(\sqrt{1-\epsilon} + \sqrt{\epsilon}\right)}. \tag{5.8}$$

Proof. For BSC(ϵ) with input and output alphabets $X = Y = \{0, 1\}$, $|\mathcal{Y}| = 2$ and the channel capacity C_{BSC} is

$$C_{BSC} = \log_e 2 + \epsilon \log_e \epsilon + (1-\epsilon)\log_e 1 - \epsilon \text{ nats.} \tag{5.9}$$

Having $P_X = (\frac{1}{2}, \frac{1}{2})$,

$$\begin{aligned} E_0(\rho, P_X) &= \rho \log_e 2 \\ &-(1+\rho)\log_e\left[(1-\epsilon)^{1/(1+\rho)} + (\epsilon)^{1/(1+\rho)}\right] \end{aligned} \tag{5.10}$$

and the derivative with respect to ρ is found as

$$\begin{aligned} \frac{\partial E_0(\rho, P_X)}{\partial \rho} &= \log_e 2 \\ &+ \frac{\frac{1}{(1+\rho)}\epsilon^{1/(1+\rho)}\log_e \epsilon + \frac{1}{(1+\rho)}(1-\epsilon)^{1/(1+\rho)}\log_e(1-\epsilon)}{\epsilon^{1/(1+\rho)} + (1-\epsilon)^{1/(1+\rho)}} \\ &- \log_e(\epsilon^{1/(1+\rho)} + (1-\epsilon)^{1/(1+\rho)}). \end{aligned} \tag{5.11}$$

Then by substituting $\rho = 1$, we get the critical rate for BSC as

$$\begin{aligned} R_{cr} &= \left.\frac{\partial E_0(\rho, P_X)}{\partial \rho}\right|_{\rho=1} \\ &= \frac{\sqrt{1-\epsilon}\log_e(4-4\epsilon) + \sqrt{\epsilon}\log_e(4\epsilon)}{2\left(\sqrt{1-\epsilon}+\sqrt{\epsilon}\right)} \\ &\quad - \log_e\left(\sqrt{1-\epsilon}+\sqrt{\epsilon}\right) \end{aligned} \tag{5.12}$$

As the values of C_{BSC}, $|\mathcal{Y}|$ and R_{cr} are plugged into (5.4), (5.6) is obtained. □

Corollary 5.3. *In the presence of block coding in a BEC(γ), the block-error probability under MLD is expected to be upper-bounded by*

$$P_e \leq e^{-N_b \xi} \tag{5.13}$$

where ξ is mathematically found as

$$\xi = \frac{(R_c + (\gamma-1)\log_e 2)^2}{-2\left(\frac{(1-\gamma)log_e(2)}{(\gamma+1)}\right)^2 + \frac{8}{e^2} + 2\log_e^2(3)}. \tag{5.14}$$

Proof. Steps are similar to the proof of Corollary 5.2, for BEC(ϵ) with input $X = \{0, 1\}$ and output alphabets $Y = \{0, 1, e\}$, $|\mathcal{Y}| = 3$ and the channel capacity C_{BEC} is

$$C_{BEC} = (1-\gamma)\log_e 2 \text{ nats.} \tag{5.15}$$

Having $P_X = (\frac{1}{2}, \frac{1}{2})$, $E_0(\rho, P_X)$ becomes

$$E_0(\rho, P_X) = -\log_e\left(2^{-\rho}(1-\gamma) + \gamma\right) \tag{5.16}$$

and the related derivative is found as

$$\frac{\partial E_0(\rho, P_X)}{\partial \rho} = \frac{(1-\gamma)\log_e 2}{(2^\rho - 1)\gamma + 1}. \tag{5.17}$$

Then by substituting $\rho = 1$, we get the critical rate for BEC as

$$R_{cr} = \left.\frac{\partial E_0(\rho, P_X)}{\partial \rho}\right|_{\rho=1} = \frac{(1-\gamma)\log_e 2}{\gamma+1}. \tag{5.18}$$

As the values of C_{BEC}, $|\mathcal{Y}|$ and R_{cr} are plugged into (5.4), (5.13) is obtained. □

5.3.2. Block-Error Probability for Polar Coding

Polar codes are the first capacity-achieving linear channel codes with low-complexity construction, encoding, and decoding algorithms [197]. A polar code $\mathcal{P}(N, K, \mathcal{A})$ is

characterized by its power of two block-length N, code dimension K and the information index set $\mathcal{A}$ with $|\mathcal{A}| = K$. Given the information set $\mathcal{A}$, the *rate-profiling* block of the encoder embeds the data $\mathbf{d} \in \{0,1\}^K$ into the carrier vector $\mathbf{u} \in \{0,1\}^N$ such that $\mathbf{u}_{\mathcal{A}} = \mathbf{d}$ and $\mathbf{u}_{\mathcal{A}^c} = \mathbf{0}$. That is, the data is embedded to $\mathbf{u}$ at indices $\mathcal{A}$, while the rest of the indices are filled with 0. The encoder then computes the polar transform. The polar transform can be mathematically provided as

$$\mathbf{x} = \mathbf{u}\,\mathbf{P}_n\,,\ \mathbf{P}_n = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes n} \tag{5.19}$$

where $n = \log_2 N$, and $\otimes$ denotes the Kronecker product. This transformation polarizes $\mathbf{u}$ in the sense that as the block-length approaches infinity, a $C(W)$ fraction of bits becomes ideal and a $1 - C(W)$ fraction becomes completely unreliable where $C(W)$ denotes the channel capacity.

The choice of $\mathcal{A}$ is called the *code construction*, and it can be performed in $\mathcal{O}(N\text{poly}(\log N))$ steps [199–201]. The encoder and the initially proposed successive cancellation (SC) decoder both have $\mathcal{O}(N \log N)$ complexity [197]. In [202], it is shown that, for any binary-input discrete memoryless channel W with symmetric capacity $I(W)$ and any rate $R < I(W)$, the polar-coding block-error probability under SC decoding satisfies the upper-bound

$$P_e \leq 2^{-N_b^{\beta}} \tag{5.20}$$

for any $\beta < \frac{1}{2}$ when the block-length N_b is large enough.

The parameter β represents a scaling factor that influences the decoding performance and, consequently, the AoI characteristics of the system. By adjusting the value of β, one can effectively control the level of error correction achieved by the polar code decoder, thereby impacting the overall AoI performance. Practically, the value of β can be adjusted through various means, such as modifying the decoding algorithm parameters, optimizing code construction techniques, or adapting the code rate based on specific application requirements.

5.4. AoI Analysis

A status update is assumed to be correctly decoded if all the codeword transmissions regarding it are correctly decoded. We define the number of blocks in the transmission as $n_i = \lceil \frac{l_i}{N_b} \rceil$ where $\lceil x \rceil$ represents the ceiling function that rounds x to the nearest integer greater than or equal to x. Then, P_e^i, probability of decoding error for sensor i's update, is given by

$$P_e^i = 1 - (1 - P_e)^{n_i}. \tag{5.21}$$

Whittle's index policy chooses M sensors having greatest W_is where

$$W_i = \frac{\Delta_i(\Delta_i - 1)p_i}{2} + \Delta_i - \Delta_i(1 - p_i)^{L-\Delta_i} \tag{5.22}$$

with $p_i = 1 - P_e^i$ being the successful transmission probability of sensor i and L is the maximum value for any Δ_i.

Theorem 5.4. *The average AoI for any block-coded transmission scheme is mathematically described as*

$$\bar{\Delta}_{general} = \frac{N + 2M}{2MN^2} \left(\sum_{i=1}^{N} \frac{l_i}{R_i} \right) \sum_{i=1}^{N} (1 - P_e)^{-n_i}. \tag{5.23}$$

Proof. Note that for sensor i, the number of blocks needed to be transmitted per update is given by n_i. The probability of success per update transmission is given as

$$p_i = 1 - P_e^i = (1 - P_e)^{n_i}. \tag{5.24}$$

Since p_i values are very close to 1, $p_i \lesssim 1$, $\forall i$, Whittle's index policy tends to prioritize the sensors having greater AoI values. Note that in this case, Whittle's index calculation for sensor i becomes

$$W_i = \frac{\Delta_i(\Delta_i + 1)}{2}. \tag{5.25}$$

According to scheduling rule of Whittle's index policy, at any time, M sensors having the greatest AoIs should be scheduled. In the beginning, the scheduled ones should have the highest M transmission times t_is since an increase in higher AoI levels adds up more to the average AoI in the network. After the sensor, that is scheduled but having the smallest transmission time among scheduled ones, completes its transmission, the

sensor that has the largest transmission time among remaining sensors will be scheduled to that idle channel. Using the same strategy, each channel will have the same expected number of sensors until the end of the transmissions of all sensors and there will be $\frac{N}{M}$ sensors per each channel in every cycle of scheduling. In expectation, the AoI cycle begins with $\mu_{\mathbf{t}}$ and ends with $\mu_{\mathbf{t}} + \frac{N}{M}\mu_{\mathbf{t}}$ resulting in an average as

$$\bar{\Delta}_{No-error} = \left(\frac{N}{2M} + 1\right)\mu_{\mathbf{t}} \tag{5.26}$$

given all the decoding operations are successful where $\mathbf{t}$ is the vector of transmission periods per update for sensors, i.e.,

$$t_i = \frac{l_i}{R_i} \tag{5.27}$$

and $\mu_{\mathbf{t}}$ can be mathematically provided as

$$\mu_{\mathbf{t}} = \frac{\sum_{i=1}^{N} \frac{l_i}{R_i}}{N}. \tag{5.28}$$

In fact, not all the decoding operations are guaranteed to be successful. The expected number of trials for sensor i to successfully transmit its update can be found as

$$\mathbb{E}[n_{trial,i}] = \frac{1}{p_i} = (1 - P_e)^{-n_i} \tag{5.29}$$

where $n_{trial,i}$ denotes the number of trials for sensor i. It is possible to obtain the average AoI by using the formula

$$\bar{\Delta}_{general} = \mu_{\mathbb{E}[\mathbf{n_{trial}}]}\bar{\Delta}_{No-error} \tag{5.30}$$

where $\mathbb{E}[\mathbf{n_{trial}}]$ denotes the vector of expected number of trials for sensors to successfully transmit a status update and $\mu_{\mathbb{E}[\mathbf{n_{trial}}]}$ is the mean of the vector. As a result, the average AoI can be found as

$$\bar{\Delta}_{general} = \frac{N + 2M}{2MN^2}\left(\sum_{i=1}^{N} \frac{l_i}{R_i}\right)\sum_{i=1}^{N}(1 - P_e)^{-n_i}. \tag{5.31}$$

□

Corollary 5.5. *Lower bound for the average AoI for any block-coded transmission scheme can be given as*

$$\bar{\Delta}_{No-error} = \frac{N + 2M}{2MN}\left[\sum_{i=1}^{N} \frac{l_i}{R_i}\right]. \tag{5.32}$$

Proof. The result follows by assigning $P_e = 0$ in equation (5.23). □

Corollary 5.6. *MLD upper bound for the average AoI for block coded transmissions in BSC(ϵ) is given as*

$$\bar{\Delta}_{ML,BSC} \leq \frac{N+2M}{2MN^2}\left(\sum_{i=1}^{N}\frac{l_i}{R_i}\right)\sum_{i=1}^{N}\left(1-e^{-N_b\psi}\right)^{-n_i} \tag{5.33}$$

where ψ is given in equation (5.7).

Proof. The result follows as $P_e \leq e^{-N_b\psi}$, which is provided in inequality (5.6), is plugged into the equation (5.23). □

Corollary 5.7. *MLD upper bound for the average AoI for block coded transmissions in BEC(γ) is found as*

$$\bar{\Delta}_{ML,BEC} \leq \frac{N+2M}{2MN^2}\left(\sum_{i=1}^{N}\frac{l_i}{R_i}\right)\sum_{i=1}^{N}\left(1-e^{-N_b\xi}\right)^{-n_i} \tag{5.34}$$

where ξ is given in equation (5.14).

Proof. The result follows as $P_e \leq e^{-N_b\xi}$, which is provided in inequality (5.13), is plugged into the equation (5.23). □

Corollary 5.8. *SC decoding upper bound for the average AoI for polar coded transmissions is given as*

$$\bar{\Delta}_{SC,Polar} \leq \frac{N+2M}{2MN^2}\left[\sum_{i=1}^{N}\frac{l_i}{R_i}\right]\sum_{i=1}^{N}\left(1-2^{-N_b^{\beta}}\right)^{-n_i}. \tag{5.35}$$

Proof. The result follows as $P_e \leq 2^{-N_b^{\beta}}$, which is provided in inequality (5.20), is plugged into the equation (5.23). □

5.5. Simulation Results

In numerical experiments, the system parameters are set as follows: $N = 50$, $M = 10$, $l_i = 5$ Mb $\forall i \in \{1, ..., N\}$ and $R_i = 1$ Mbps $\forall i \in \{1, ..., N\}$, i.e., all IoT

nodes are assumed to be equivalent. In Fig. 5.1, upper bounds for the average AoI in case of MLD on a BSC(ϵ), $\bar{\Delta}_{ML,BSC}$, MLD on a BEC(γ), $\bar{\Delta}_{ML,BEC}$ and SC decoding of polar codes, $\bar{\Delta}_{SC,Polar}$ are normalized with respect to average AoIs of the uncoded transmissions in which the bit error rate equals 3×10^{-7}. The sample calculation for BSC case can be given as

$$||\bar{\Delta}_{ML,BSC}(\epsilon)|| = \frac{(\bar{\Delta}_{ML,BSC}(\epsilon) - \bar{\Delta}_{uncoded})}{\bar{\Delta}_{uncoded}}. \tag{5.36}$$

$||\bar{\Delta}_{ML,BEC}(\gamma)||$ and $||\bar{\Delta}_{SC,Polar}(\gamma)||$ can be obtained by plugging the upper bounds in (5.34) and (5.35), respectively, in (5.36). The normalized average AoI expressions have been evaluated with respect to changes in block-length, N_b and ϵ for BSC, γ for BEC and β for polar coding. The uncoded transmissions can achieve similar AoI performance to channel coded counterparts when bit error probabilities are extremely lower (redundant bits are excluded so that number of bits to transmit is lower than channel coded updates), accounting for the superiority of channel coding in unreliable channels. The normalized average AoI floors being approximately -0.2 indicate the lowest achievable AoI values with block coding are 20% lower than those without.

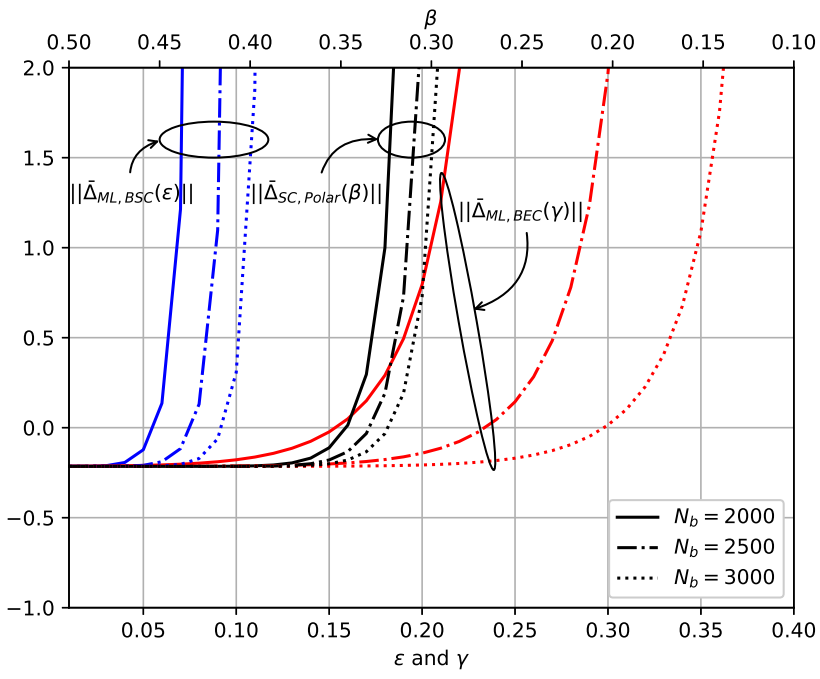


Figure 5.1. Upper bounds for normalized average AoIs.

As seen from the results, increasing the block-length decreases AoI values by delaying the AoI take off and rapid increase for all of the cases as expected since increasing block-length helps to enhance the efficiency of coding and decoding operations. In case of $\bar{\Delta}_{ML,BSC}$, $\epsilon \lessapprox 0.03$ is enough to satisfy an average AoI that is nearly equal to the lowest achievable average AoI. Performing a similar experiment on $BEC(\gamma)$, $\gamma \lessapprox 0.06$ is enough to nearly equalize the theoretical achievable upper bound and lower bound for the average AoI. However, it is important to note that an increase in block-length has a larger impact on BEC case compared to BSC as the AoI take off is further delayed in BEC. As for the AoI performance of polar codes in case of SC decoding, it is practically achievable and as seen in Fig. 5.1, for $\beta \gtrapprox 0.375$, it is possible to obtain nearly optimal AoI performance in the network. Also, when $\beta \lessapprox 0.325$, the AoI floor significantly increases even for higher block-lengths. β is known to be less than $\frac{1}{2}$, and it is asymptotically possible for β to get arbitrarily close to $\frac{1}{2}$. However in the practical scenario, this is not possible and we conclude that one needs to sustain a $\beta \gtrapprox 0.375$, to get a superb AoI performance.

Increasing β in polar coding while maintaining a constant data rate involves optimizing code construction, adjusting error-correction capability, and exploring advanced coding techniques. These strategies aim to enhance the efficiency and reliability of data transmission. However, these enhancements may lead to increased computational complexity due to more intricate encoding and decoding processes. Therefore, it is essential to strike a balance between β enhancement and computational feasibility. For instance, in IoT communication systems, adjusting the error-correction capability of polar codes can improve reliability without altering the data rate, but this must be carefully managed to ensure computational resources remain viable for IoT devices.

The gap between the AoI of the general class of block codes with MLD and the AoI of the practical subcase of polar codes with SC decoding may be found by calculating the differences between the equations (5.33), (5.34) and (5.35) given the network parameters and also illustrated in Fig. 5.1. As seen, $\bar{\Delta}_{ML,BSC}$ tends to converge the AoI lower bound for $\epsilon <= 0.03$ whereas $\bar{\Delta}_{ML,BEC}$ converges for $\gamma <= 0.06$ given

$N_b > 2000$. For the same interval of N_b, $\bar{\Delta}_{SC,Polar}$ performs similarly given $\beta >= 0.375$, as illustrated in Fig. 5.1. In addition, $||\bar{\Delta}_{ML,BEC}(\gamma)||$ is the most susceptible case in terms of block-length of coding while $||\bar{\Delta}_{SC,Polar}(\beta)||$ represents the least susceptible case.

5.6. Conclusion

The effect of block coding on AoI is considered. The general lower and upper bounds on the average AoI of the network are derived for MLD in block codes and SC decoding in polar codes. In addition, AoI gap, between SC decoded polar code and ML decoded block code with no error scenarios is evaluated with numerical experiments where we conclude that β in block-error probability of polar codes is a key to maintain a satisfying freshness of update packets in SC decoding of polar codes.

6. IMPACT OF NETWORK CODING ON AGE OF INFORMATION

6.1. Introduction

Ensuring data freshness has become a significant challenge for communication networks due to the massive amounts of data generated by IoT devices and the time-sensitive nature of some applications. It is crucial to evaluate the performance of potential network deployment schemes in terms of age of information (AoI), a key metric proposed in [2], for next generation wireless communication systems. AoI at a destination is defined as the elapsed time since the source generated the most recent received information update packet [2]. Prior works, including [14], have revealed that maximizing the sum-rate or minimizing the delay are not comparable to minimizing the AoI. Therefore, AoI optimization/minimization in wireless networks has become a significant concern and has been addressed in various contexts, such as IoT networks [43, 151, 203], UAV-Assisted IoT networks [204, 205], broadcast wireless networks [206, 207], energy harvesting sensor networks [84,208], and multi-state time varying channels [16].

In [155], the importance of AoI is emphasized in an IoT network where devices observe potentially diverse physical processes and are expected to regularly communicate the state of these processes at a destination node. The authors of [189] develop efficient algorithms for determining the average AoI optimal update generation policy and investigate the optimality of a zero-wait policy for a model in which updates are transmitted from a source node to a destination over a channel. Moreover, in [151], the joint design of device scheduling and status sampling is investigated with the aim of minimizing the average AoI for a real-time IoT monitoring system comprised of a large number of IoT devices. In [209], a low-complexity scheduling algorithm based on Whittle's index is proposed to minimize the AoI in the network, taking a stochastic arrival model and various IoT devices into consideration. In [183], the asymptotic optimality

of Whittle's index policy on AoI minimization for selecting which sensors to broadcast in a given time slot is demonstrated. In contrast to previous research, transmission flaws in the network are taken into account in [210, 211] since a lost packet does not improve the AoI performance at the receiver. The multicast transmission of an access point to multiple IoT devices is investigated in [148], where a deadline for the service time of each multicast status update is enforced, and when the deadline is reached, the update is dropped. The authors evaluate the timeliness of status updates in this multicast system and demonstrate that enforcing deadlines decreases the average and peak AoI of the network.

In [90, 139, 140], centralized and decentralized approaches to AoI are examined, and optimization problems are addressed in accordance with these approaches. Due to the importance of energy efficiency, works such as [107, 108, 208] analyze the AoI and energy trade-off and propose optimized update strategies that account for this. The authors consider and solve sampling optimization problems in various network deployment scenarios in [149, 150, 153, 154, 212] because the sampling pattern of sensor updates is highly determinative of AoI behavior in a network. [151] and [152] focus on sampling optimization through the application of learning algorithms. In [52], the trade-off between timeliness and distortion is highlighted, wherein transmitting updates more frequently can reduce the AoI at the expense of distortion, and vice versa. AoI behaviors are examined in [51, 70, 77, 118–121] where status updates are pre-processed. Because raw data must be processed, optimization strategies which reduce the impact of preprocessing on AoI are available. In [144–146], the minimum AoI has been analyzed and optimized for networks with variable packet sizes and nonperiodic scheduling.

Employing network coding, the benefits are increased throughput and network robustness, as demonstrated in a number of works, including [33–37]. Such advantages incur an increase in the computational complexity of network nodes. At the heart of this chapter's discussion is the trade-off between such a computational burden and the AoI gain resulting from the need for fewer transmissions over unreliable channels.

The AoI performance of network-coded flows is investigated in [38–41]. Similarly, we concentrate on the AoI performance characteristics of network coded flows employed in multicast transmissions versus uncoded flows in this study. However, none of the aforementioned works take into account the computational burden that network coding imposes due to operations involving the combination of various data packets. The computational complexity is a non-negligible parameter because AoI is a performance metric designed specifically for IoT networks, where sensors may have limited computational capabilities that have a negative impact on coding scenarios. In addition, generalized model of multicast transmissions has been considered instead of considering only a toy scenario such as two-way relay networks. This chapter's primary contribution is an examination of the AoI effect of multicasting network-coded packets in wireless networks. Specifically, our most significant contributions are as follows:

- The exact derivations of peak and average expected AoIs are provided for two-way relay networks with and without network coding where computational burden to combine packets is taken into account.
- The exact derivations of peak and average expected AoIs are provided for general multicast transmissions with and without network coding where computational burden to combine packets is taken into account.
- Behavioral analyses of proportional AoI gain with respect to network parameters are given where the importance of the computational capacity and transmission rates of nodes come out to be the key determinants of the advantage/disadvantage of network coding over uncoded scenario.

The remainder of the chapter is structured as follows: In Section 6.2, we provide a preliminary discussion on network coding principles. In Section 6.3, the system model is presented. In order to facilitate the reading and comprehension, the peak and average expected AoIs of a simple relay network are mathematically derived in Section 6.4. In Section 6.5, the general multicast model is examined, and the network's peak and average expected AoIs are mathematically derived. Section 6.6 covers the behavioral analyses of proportional AoI gain expressions with respect to network parameters. In

Section 6.7, simulation results and commentary discussions are presented, while Section 6.8 contains the conclusion. This chapter is finalized by redoing the same AoI analysis in an energy harvesting enabled environment as seen in Section 8.

6.2. Preliminaries on Network Coding

The combination of packets is typically assumed to be an XOR operation, though this is not required. The operation may result in a linear combination of different packets, as illustrated in [35], or it may be a non-linear combination as in [213], with the objective of maximizing the achievable rate region in discrete two-way relaying channels. Packets are not required to be the same length. Consider a scenario in which the first packet, P_1, is l_1 bits in length and the second packet, P_2, is l_2 bits in length. Assuming without loss of generality that $l_1 \geq l_2$, we pad the second packet with $l_1 - l_2$ zeros so that their lengths become identical. After the padding operation, a linear coding can be implemented, i.e. additions and multiplications can be performed between the bits of P_1 and P_2 in a finite field. The length of the resulting packet will be l_1. However, as stated, coding options are not limited to linear operations, so any type of non-linear operation can be employed; the choice must be case dependent. We stick to linear network coding along the chapter and the following subsections are formed accordingly.

6.2.1. Encoding

Consider a scenario where a node linearly combines n packets to 1. The packets are combined using the operation

$$Y = \sum_{i=1}^{n} f_i M_i \tag{6.1}$$

where $f = (f_1, f_2, ..., f_n)$ are the combination coefficients, M_i is the ith packet to be combined and Y is the encoded vector. The summation is performed bitwise and inside a finite field F. If we restrict the field to F_2, then the operation becomes an addition in F_2, which is an XOR. An important reminder is that the cardinality of field must be big

enough to guarantee an unambiguous recovery of the original signals $M_1, M_2, ..., M_n$. Assume we have α such encoded packets, $Y_1, Y_2, ..., Y_\alpha$, recursive encoding is possible so that the encoded packets $Y_1, Y_2, ..., Y_\alpha$ can be further encoded as

$$Z = \sum_{i=1}^{\alpha} g_i Y_i \tag{6.2}$$

where $g = (g_1, g_2, ..., g_\alpha)$ are the new combination coefficients and Y_i is the ith encoded packet and Z is the encoded packet of encoded packets.

A packet must contain the coefficients relative to the original packets and the encoded data in order for the perfect recovery of the original packets to be possible. For instance, a node that encodes Y from the original packets $M_1, M_2, ..., M_n$, must transmit a packet containing (f, Y) set. In the case of recursive encoding, Z will be encoded by packets including (f^i, Y_i) set, where f^i represents the coefficients of the ith packet to be encoded. Thus, the overall combination will be transmitted as (h, Z) where h can mathematically be described as

$$h_t = \sum_{i=1}^{\alpha} g_i f_t^i. \tag{6.3}$$

6.2.2. Decoding

Consider an instant where a node received packets containing (f^1, Y_1), (f^2, Y_2),..., (f^κ, Y_κ), so the node has the knowledge of κ different coefficient sets for κ corresponding encoded Ys. As indicated by the equation (6.1), the number of original message packets is n. Decoding is therefore equivalent to solving a linear system containing κ equations and n unknowns. Clearly, $\kappa \geq n$ is required in order to recover the original messages. This condition is necessary, but it is not sufficient because the equations may have a linear dependence. To ensure sufficiency, a number of linearly independent equations equal to or greater than n is required.

6.2.3. Packet Combining

We have previously explained how to perform linear combinations of packets. Now comes the question of how to choose the combining coefficients.

There is no unique solution to this problem. Nonetheless, the prevalent and widely accepted method is to choose the coefficients uniformly at random over a finite field F as in [214], where nodes are independent in this selection. The system is therefore decentralized. As noted previously, linearly dependent coefficient sets are undesirable. As demonstrated in [215], the probability of having linearly dependent coefficient sets decreases rapidly as the field cardinality increases and becomes nearly negligible for field F_{2^8}. This behavior is referred to in the literature as the curse of dimensionality. Therefore, random linear network coding appears to be a viable and promising technique.

Employing deterministic network codes is an alternative strategy. A deterministic network code is proposed in [216] for multicasting in polynomial time. As the coefficients of each node are predetermined, the packets that are forwarded need only carry the information and not the coefficient vectors.

6.3. System Model and Problem Formulation

Let us consider a general multicast scenario seen in Fig. 6.1. Node B is the sender and may either be considered as a bottleneck in a network or be considered as the source and it is assumed to have a status update, which is divided into n packets, in order to forward to k receivers denoted as $R_1, R_2, ..., R_k$. None of the receivers is assumed to have any of n packets in advance. When a receiver gets all n packets, it integrates them to get the update.

It is assumed that the update is taken when the last message packet constituting the update arrives. For simplicity, assume that a status update is constantly generated

at B whenever the last sent packet among all is decoded at the last receiving node. Additionally, dividing the update at the source and integrating the message packets at a receiver is assumed to immediately take place.

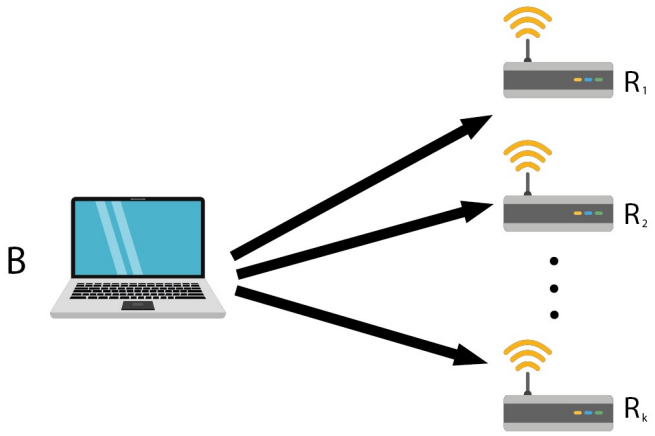


Figure 6.1. Multicast network.

As linear network coding is adopted, only linear combinations of the incoming messages are possible and this leads each message to be a linear combination of the message packets $M_1, M_2, ..., M_n$ released from the source s. Therefore, an encoded message forwarded by an intermediate node in the network can be described as

$$E_b = \sum_{i=1}^{n} f_i M_i \mod q \tag{6.4}$$

where f_i is the encoding coefficient for message M_i. The encoding coefficients can be chosen randomly [214] or in a deterministic way [216] as noted in Section 6.2.3. We stick to the random generation approach in this work.

Additionally, as previously explained in Section 6.2.1, it is important that a forwarded encoded packet must contain the encoding coefficients in order to allow for the sufficiency of the perfect recovery of the original packets. Thus, the form of the encoded message message is given as

$$E = [E_b, f_1, f_2, ..., f_n]. \tag{6.5}$$

Since network coded packets contain extra coefficients, coded packets are longer than the original packets. This increases the transmission times of encoded packets.

The computation time of a node j to mix different messages is denoted by $\tau_{j,comp}$ and it is dependent on the computational capacity of the node and the total number of operations. Let l_i denote the length of the message M_i. Computational capacity of node j is denoted by $C_{j,comp}$. It is assumed that the length of messages and computational capacity of nodes are equivalent, i.e., $l_i = l_j = l$ and $C_{i,comp} = C_{j,comp} = C_{comp}$ $\forall i, j \in \{1, ..., n\}$. Let the number of messages to be mixed at node j be $n_{j,mix}$. Then the number of operations can be given by $l(n_{j,mix} - 1)$ additions to be performed in $\mathcal{Z}_q$ and insertion of $n_{j,mix}$ coefficients to the mixed message for the perfect recovery in the receiving node. The total number of encoding operations becomes $l(n_{j,mix} - 1) + n_{j,mix}$ while we assume addition and insertion operations to be equivalent in terms of computational complexity. Let $2^m < q \leq 2^{m+1}$, m being a non-negative integer, i.e., $m \in \mathbb{Z}^*$, an element of $\mathcal{Z}_q$ field can be represented by $m+1$ bits. Thus, the number of operations in bits is found as $(m+1)(l(n_{j,mix} - 1) + n_{j,mix})$. The computation time, $\tau_{j,comp}$ can finally be given by

$$\tau_{j,comp} = \frac{(m+1)(l(n_{j,mix} - 1) + n_{j,mix})}{C_{comp}}. \tag{6.6}$$

The transmission time of a packet between two nodes is assumed to be dependent on the length of the packet and the transmission rate between the sending and receiving node and denoted by $\tau_{j\to k}^{M_i}$, and by $\tau_{j\to k}^{E_i}$ in case of an encoded packet. The transmission rate from node j to k is R_{jk}. The length of the message is l for original message packets and $l + n_{j,mix}$ for encoded ones. Thus, $l(m+1)$ bits for an uncoded packet and $(l + n_{j,mix})(m+1)$ bits for a coded packet are enough for the representation. The transmission times are given as

$$\tau_{j\to k}^{M_i} = \frac{l(m+1)}{R_{jk}} \tag{6.7}$$

$$\tau_{j\to k}^{E_i} = \frac{(l + n_{j,mix})(m+1)}{R_{jk}} \tag{6.8}$$

for an uncoded and an encoded message packet, respectively.

Noting that $n_{j,mix} = n$ throughout the chapter, message packets are mixed at source j and transmitted to receiver k. In such a case k needs at least n randomly coded successful transmissions. In order to derive original message packets, receiver k must perform the linear operation seen $\mathbf{M}_{n\times l} = \mathbf{f}_{n\times n}^{-1} \cdot \mathbf{E}_{n\times l}$ which can be seen as

$$\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix} \times \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1l} \\ M_{21} & M_{22} & \cdots & M_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & M_{nl} \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} & \cdots & E_{1l} \\ E_{21} & E_{22} & \cdots & E_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ E_{n1} & E_{n2} & \cdots & E_{nl} \end{bmatrix} \tag{6.9}$$

since the receiver has the encoding coefficients $\mathbf{f}$ and the encoded messages $\mathbf{E}$. To this end, a simple method of matrix inversion, which is well-known and widely used Gauss-Jordan Elimination, to get $\mathbf{f}_{n\times n}^{-1}$ has a computational complexity of $\mathcal{O}(n^3)$. Additionally, multiplication of $\mathbf{f}_{n\times n}^{-1} \cdot \mathbf{E}_{n\times l}$ requires $n^2l+(n-1)nl$ operations which add up to $\mathcal{O}(n^3)+n^2l+(n-1)nl$ in total to complete decoding operations at the receiver side. The decoding time can be given as

$$\tau_{k,decode} = \frac{\mathcal{O}(n^3) + n^2l + (n-1)nl}{C_{comp}}. \tag{6.10}$$

It is assumed that a message packet waits in the buffer of each and every node. In addition, the waiting time for receiver i, $\tau_{i,wait}$, is independent of the message and only dependent on the traffic at the sending node, where $\tau_{i,wait}$ follows the exponential distribution and can be given as

$$f(\tau_{i,wait}; \lambda) = \begin{cases} \lambda e^{-\lambda \tau_{i,wait}}, & \tau_{i,wait} \geq 0 \\ 0, & \tau_{i,wait} < 0. \end{cases} \tag{6.11}$$

Given the source multicasts to k receivers, and assuming that it is waiting for the ACK or NACK from all receivers to transmit the next packet, the expected waiting time after each multicast transmission can be modeled as $\tau_{wait} = \max(\tau_{1,wait}, ..., \tau_{k,wait})$ where $\tau_{1,wait}, ..., \tau_{k,wait}$ are independent and identically distributed random variables. Following the steps in [217], the expected waiting time is found as

$$\mathbb{E}[\tau_{wait}] = \frac{H_k}{\lambda} \tag{6.12}$$

where H_k denotes the kth harmonic number, i.e., $H_k = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{k}$. Lastly, there is a success probability for every packet's transmission that is assumed to be dependent

on the sending node j and receiving node k. This probability is denoted by p_{jk} where $p_{jk}, \forall j, k$ is in the interval $[0, 1]$ and once sampled, its value remains constant over time.

The AoI computation for each packet would involve one sensing operation of the current time and one subtraction operation of the update timestamp ($u(t)$) from the current time (t), i.e., $\Delta(t) = t - u(t)$. The computational complexity of this operation is $\mathcal{O}(1)$. Compared to the transmission time, decoding time and complexity, the complexity of AoI measurement and computation operations would be significantly less and thus they are omitted in computations.

In order to warm up for the general multicast model, we first introduce a simplified scenario where there are only two receivers and then generalize to the k user model as seen in the following sections.

6.4. Two-User Multicast Scheme

Consider the two-way relay network seen in Fig. 6.2 where receiver nodes X and Z regularly interchange status update packets. Status update packets sent from node X is denoted by a and the ones sent by Z is denoted as b. It is assumed that a new update is generated at X (or Z) when Z (or X) receives the formerly generated update and packets are retransmitted immediately in case of a transmission failure. The goal is to inspect the behavior of the expected AoIs in the network whether network coding is employed or not. Doing so, the transmission steps of uncoded flows are denoted by s_i, $i \in \{1, 2, 3, 4\}$ from top to bottom, and the steps of coded flows are represented by $s_{i,coded}$, $i \in \{1, 2, 3\}$, i.e., an extra subscript is employed to highlight the difference from uncoded ones. Message packets are assumed to be sampled from $\mathcal{Z}_q$ field, i.e., element-wise modulo q addition is assumed to be represented by $\oplus$ along the section.

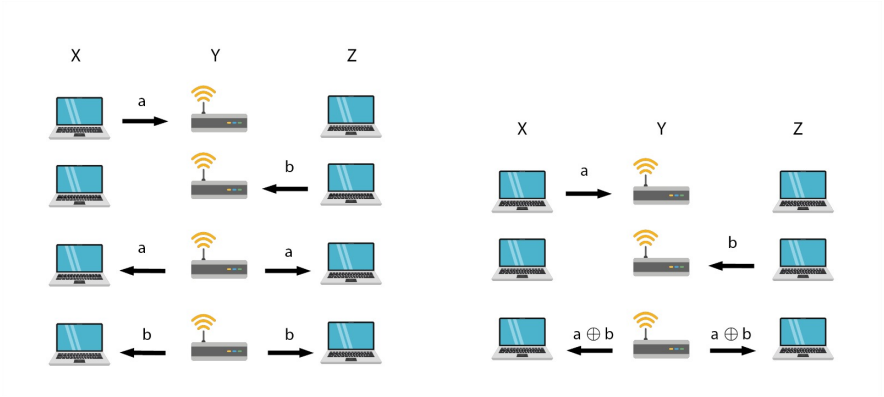


Figure 6.2. Transmissions without (left) and with (right) network coding.

6.4.1. Peak and Average Expected AoI without Coding

To begin with the case without coding seen in the left hand side of Fig. 6.2, the first two steps are relatively easy to assess and the expected completion times are given straightforwardly for the first step as

$$\begin{aligned}\mathbb{E}[t_{s_1}] &= \mathbb{E}[\tau^a_{X\to Y} + \tau_{wait}]\frac{1}{p_{XY}} \\ &= \left[\frac{l(m+1)}{R_{XY}} + \frac{1}{\lambda}\right]\frac{1}{p_{XY}}\end{aligned} \tag{6.13}$$

and for the second step as

$$\mathbb{E}[t_{s_2}] = \left[\frac{l(m+1)}{R_{ZY}} + \frac{1}{\lambda}\right]\frac{1}{p_{ZY}} \tag{6.14}$$

since $\mathbb{E}[\tau^a_{X\to Y}]$ or $\mathbb{E}[\tau^a_{Z\to Y}]$ are deterministic and $\mathbb{E}[\tau_{wait}] = 1/\lambda$. At steps s_3 and s_4, Y multicasts the messages where it is assumed that single-rate multicast scheme is adopted. Since X already has packet a and Z already has packet b, third step multicast rate is chosen as R_{YZ} and fourth step multicast rate is chosen as R_{YX}. Expected completion times of steps are then found for the third step as

$$\mathbb{E}[t_{s_3}] = \left[\frac{l(m+1)}{R_{YZ}} + \frac{1}{\lambda}\right]\frac{1}{p_{YZ}} \tag{6.15}$$

and for the last step as in

$$\mathbb{E}[t_{s_4}] = \left[\frac{l(m+1)}{R_{YX}} + \frac{1}{\lambda}\right]\frac{1}{p_{YX}}. \tag{6.16}$$

Theorem 6.1. *In the absence of coding, the peak expected AoI of X which is equal to that of Z and thus equal to the network's peak expected AoI, which can be expressed as*

$$\begin{aligned}\max_t \mathbb{E}[\Delta(t)] &= \max_t\{\mathbb{E}[\Delta_X(t)], \mathbb{E}[\Delta_Z(t)]\} \\ &= 2\left[\mathbb{E}[t_{s_1}] + \mathbb{E}[t_{s_2}] + \mathbb{E}[t_{s_3}] + \mathbb{E}[t_{s_4}]\right].\end{aligned} \tag{6.17}$$

Proof. Since the case is symmetric for X and Z, i.e., $\max_t\{\mathbb{E}[\Delta_X(t)]\} = \max_t\{\mathbb{E}[\Delta_Z(t)]\}$, let us consider the peak expected AoI of Z, i.e., $\max_t\{\mathbb{E}[\Delta_Z(t)]\}$. At any time when Z receives an update which is generated with a timestamp $u(t)$, $\mathbb{E}[\Delta_Z(t)]$ becomes $u(t) + \mathbb{E}[t_{s_1} + t_{s_2} + t_{s_3} + t_{s_4}] - u(t) = \mathbb{E}[t_{s_1} + t_{s_2} + t_{s_3} + t_{s_4}]$ and a new status update packet is generated at X. Until this newly generated packet arrives at Z, $\mathbb{E}[\Delta_Z(t)]$ increases to $2\,\mathbb{E}[t_{s_1} + t_{s_2} + t_{s_3} + t_{s_4}]$ and this sets the peak of the expected AoI in the network. □

Theorem 6.2. *When coding is not adopted, average expected AoI of X is equal to that of Z and thus equal to the average expected AoI of the network which can mathematically be expressed as*

$$\begin{aligned}< \mathbb{E}[\Delta(t)] > &= \frac{1}{2}\sum_{i\in\{X,Z\}} \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{E}[\Delta_i(t)]dt \\ &= 1.5\left[\mathbb{E}[t_{s_1}] + \mathbb{E}[t_{s_2}] + \mathbb{E}[t_{s_3}] + \mathbb{E}[t_{s_4}]\right].\end{aligned} \tag{6.18}$$

Proof. Since the case is symmetric for X and Z, let us consider the average expected AoI at Z, i.e., $\lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{E}[\Delta_Z(t)]dt$. Noting that the expected AoI behavior of Z represents a saw-tooth behavior where the low-end is at $\mathbb{E}[t_{s_1} + t_{s_2} + t_{s_3} + t_{s_4}]$ and high-end is at $2\,\mathbb{E}[t_{s_1} + t_{s_2} + t_{s_3} + t_{s_4}]$, it is straightforward to conclude that the average of $\mathbb{E}[\Delta_Z(t)]$ is given as $1.5\left[\mathbb{E}[t_{s_1}] + \mathbb{E}[t_{s_2}] + \mathbb{E}[t_{s_3}] + \mathbb{E}[t_{s_4}]\right]$ and this completes the proof. □

6.4.2. Peak and Average Expected AoI with Network Coding

As for the evaluation of the coded flows seen in Fig. 6.2, the first two steps are clearly equivalent to ones of uncoded flows and given as

$$\begin{aligned} \mathbb{E}[t_{s_{1,coded}}] &= \mathbb{E}[t_{s_1}] \\ \mathbb{E}[t_{s_{2,coded}}] &= \mathbb{E}[t_{s_2}] \end{aligned} \tag{6.19}$$

as there is no need for extra coefficients in message packets for recovery thanks to simple modulo q addition operation.

For the evaluation of the third step $s_{3,coded}$, note that Y will continue to retransmit $a \oplus b$ until both X and Z successfully receive the message packet. Let ζ be the number of transmissions until both X and Z get the coded packet. Expected number of transmissions until both nodes successfully receive the message packet, $\mathbb{E}[\zeta]$, is required to be known to find $\mathbb{E}[t_{s_{3,coded}}]$. Expected time for the completion of the third step, $\mathbb{E}[t_{s_{3,coded}}]$, may be given as

$$\begin{aligned} \mathbb{E}[t_{s_{3,coded}}] &= \mathbb{E}[\tau_{Y,comp}] + \mathbb{E}[\tau_{Y\to X}^{a\oplus b} + \tau_{wait}]\,\mathbb{E}[\zeta] \\ &\quad + \mathbb{E}[\tau_{X,decode}] \\ &= \frac{l(m+1)}{C_{comp}} + \left[\frac{l(m+1)}{R_M} + \frac{3}{2\lambda}\right]\mathbb{E}[\zeta] \\ &\quad + \frac{l(m+1)}{C_{comp}} \end{aligned} \tag{6.20}$$

assuming that step s_1 does not begin until the last successfully received message is decoded and R_M is the single-rate multicast transmission rate, being $R_M = \min\{R_{YX}, R_{YZ}\}$ and $\mathbb{E}[\tau_{wait}] = H_2/\lambda = 3/2\lambda$ due to multicasting to two receivers.

Noting that $\zeta = \max\{\zeta_X, \zeta_Z\}$ where ζ_i is the number of transmission until node i successfully receives the message packet and using the maximum-minimums identity [218], we get an equivalent expression as

$$\zeta = \max\{\zeta_X, \zeta_Z\} = \zeta_X + \zeta_Z - \min\{\zeta_X, \zeta_Z\}. \tag{6.21}$$

Taking the expectation of both sides, we have

$$\mathbb{E}[\zeta] = \mathbb{E}[\zeta_X] + \mathbb{E}[\zeta_Z] - \mathbb{E}[\min\{\zeta_X, \zeta_Z\}] \tag{6.22}$$

and then the last term $\mathbb{E}[\min\{\zeta_X, \zeta_Z\}]$ is found as

$$\begin{aligned}\mathbb{E}[\min\{\zeta_X, \zeta_Z\}] &= \sum_{i=0}^{\infty}[(1-p_{YX})(1-p_{YZ})]^i \\ &= \frac{1}{1-(1-p_{YX})(1-p_{YZ})}.\end{aligned} \tag{6.23}$$

Finally, $\mathbb{E}[\zeta]$ is expressed by

$$\mathbb{E}[\zeta] = \frac{1}{p_{YX}} + \frac{1}{p_{YZ}} - \frac{1}{p_{YX}+p_{YZ}-p_{YX}p_{YZ}}. \tag{6.24}$$

Theorem 6.3. *When coding is applied, the peak expected AoI of X is equal to that of Z and hence equal to the network's peak expected AoI, which may be written as*

$$\begin{aligned}\max_t \mathbb{E}[\Delta_{coded}(t)] &= \max_t\{\mathbb{E}[\Delta_X(t)], \mathbb{E}[\Delta_Z(t)]\} \\ &= 2\left[\mathbb{E}[t_{s_1}] + \mathbb{E}[t_{s_2}] + \mathbb{E}[t_{s_{3,coded}}]\right].\end{aligned} \tag{6.25}$$

Proof. The result can be found by following the same steps as in the proof of Theorem 6.1. □

Theorem 6.4. *When coding is performed, the average expected AoI of X is identical to that of Z and, thus, equal to the network's average expected AoI, which may mathematically be given as*

$$\begin{aligned}< \mathbb{E}[\Delta_{coded}(t)] > &= \frac{1}{2}\sum_{i\in\{X,Z\}} \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathbb{E}[\Delta_i(t)]dt \\ &= 1.5\left[\mathbb{E}[t_{s_1}] + \mathbb{E}[t_{s_2}] + \mathbb{E}[t_{s_{3,coded}}]\right].\end{aligned} \tag{6.26}$$

Proof. The result can be found by following the same steps as in the proof of Theorem 6.2. □

6.4.3. Comparison of Peak and Average Expected AoIs with or without Coding

The difference of peak expected AoIs can be given by

$$\max_t \mathbb{E}[\Delta(t)] - \max_t \mathbb{E}[\Delta_{coded}(t)] = 2\left[\mathbb{E}[t_{s_3}] + \mathbb{E}[t_{s_4}] - \mathbb{E}[t_{s_{3,coded}}]\right] \tag{6.27}$$

and the difference of average expected AoIs is

$$< \mathbb{E}[\Delta(t)] > - < \mathbb{E}[\Delta_{coded}(t)] >= 1.5 \left[\mathbb{E}[t_{s_3}] + \mathbb{E}[t_{s_4}] - \mathbb{E}[t_{s_{3,coded}}]\right]. \tag{6.28}$$

If the differences are positive, it can be concluded that network coding is more efficient, in terms of expected AoI in the network, compared to transmissions without employing coding.

We evaluate the following expression which we call proportional AoI gain, i.e.,

$$\gamma = \frac{\left[\mathbb{E}[t_{s_3}] + \mathbb{E}[t_{s_4}] - \mathbb{E}[t_{s_{3,coded}}]\right]}{\mathbb{E}[t_{s_{3,coded}}]} \tag{6.29}$$

representing the proportional time difference of uncoded multicast and coded multicast transmissions and the closed form expression can be seen as

$$\frac{\left[\mathbb{E}[t_{s_3}] + \mathbb{E}[t_{s_4}] - \mathbb{E}[t_{s_{3,coded}}]\right]}{\mathbb{E}[t_{s_{3,coded}}]} = \frac{\left[\frac{l(m+1)}{R_{YZ}} + \frac{1}{\lambda}\right]\frac{1}{p_{YZ}} + \left[\frac{l(m+1)}{R_{YX}} + \frac{1}{\lambda}\right]\frac{1}{p_{YX}}}{\left[\frac{l(m+1)}{R_M} + \frac{3}{2\lambda}\right]\left[\frac{1}{p_{YX}} + \frac{1}{p_{YZ}} - \frac{1}{p_{YX}+p_{YZ}-p_{YX}p_{YZ}}\right] + 2\frac{l(m+1)}{C_{comp}}} - 1. \tag{6.30}$$

Behaviors of γ with respect to network parameters are presented in Section 6.6.

In this scenario, note that the receivers X and Z already have one of the packets they need. What if they do not? What if there are k receivers instead of 2. Now, we address the general multicast question and observe the trade-off between coded and uncoded flows if the receivers do not have any of the packets they need.

6.5. General Multicast Scheme

We consider the scenario seen in Fig. 6.1 with the assumptions in Section 6.3.

6.5.1. Peak and Average Expected AoI without Coding

It is important to address the question of sending n packets successfully to k receivers in minimum number of transmissions in order to minimize the AoI, peak or average. First, considering the scenario without coding, it is necessary to obtain the

expected number of transmissions to deliver packet i successfully to all receivers, which is denoted as T_i, being $T = T_1 = T_2 = ... = T_n$. Using a similar notation to the previous section, let ζ be the number of transmissions until all k nodes receive the sent packet and ζ_i be the number of transmissions until R_i successfully receives the message packet.

The problem of finding T can be modeled as k independent Bernoulli trials with success probabilities are $p_1, ..., p_k$ respectively where p_i denotes the success probability of a transmission from B to R_i, $i \in \{1, 2, ..., k\}$. The aim is to find the expected number of transmissions where all k receivers have successfully got the packet at least once. For receiver R_i, the probability that the packet is not received after β transmissions is $(1 - p_i)^\beta$. In this case, probability of receiving the packet becomes $1 - (1 - p_i)^\beta$. The probability that transmissions to each receiver have succeeded at least once is given as

$$P^{suc}_{R_1,...,R_k}(\beta) = \prod_{i=1}^{k} \left[1 - (1 - p_i)^\beta\right] \tag{6.31}$$

whereas the probability that the packet has not been received by each and every receiver can be mathematically described as

$$P^{fail}_{R_1,...,R_k}(\beta) = 1 - P^{suc}_{R_1,...,R_k}(\beta). \tag{6.32}$$

As a result, the expected number of multicast transmissions, $T = T_i = \mathbb{E}[\zeta]$, required for all receivers to obtain a packet is given as

$$T = T_i = \mathbb{E}[\zeta] = \sum_{m=0}^{\infty} P^{fail}_{R_1,...,R_k}(\beta). \tag{6.33}$$

Theorem 6.5. *In the absence of coding, the peak expected AoI of the network is mathematically given as*

$$\begin{aligned} &\max_{i,t}\{\, \mathbb{E}[\Delta_i(t)]\} \\ &\quad = \max_i \left(\left[T\,(2n-1) - \frac{1}{p_i} \right] \left[\frac{l(m+1)}{R_M} + \frac{H_k}{\lambda} \right] + \frac{\left(\frac{2}{p_i} - 1\right) H_k + 1}{\lambda} \right) \end{aligned} \tag{6.34}$$

where $\max_{i,t}$ *is the maximum in time and among nodes. On the other hand, the average*

expected AoI of the network is found as

$$\frac{1}{k}\sum_{i=1}^{k}\lim_{T\to\infty}\frac{1}{T}\int_0^T \mathbb{E}[\Delta_i(t)]dt \\ = \frac{1}{k}\sum_{i=1}^{k}\left(\left[T\left(\frac{3n}{2}-1\right)-\frac{1}{2p_i}\right]\left[\frac{l(m+1)}{R_M}+\frac{H_k}{\lambda}\right]+\frac{\left(\frac{3}{2p_i}-1\right)H_k+1}{\lambda}\right). \tag{6.35}$$

Proof. In order to find the average expected AoI in the network, first consider an arbitrary receiver i. Assume that receiver i gets the last packet of an update at t_i. Then, the time it takes for source to generate a new update is

$$\Delta_{i,diff} = \frac{H_k-1}{\lambda}+\left[T-\frac{1}{p_i}\right]\left[\frac{l(m+1)}{R_M}+\frac{H_k}{\lambda}\right] \tag{6.36}$$

since the last packet must be arrived at all receivers. After the generation of the new update, the last packet of this update arrives at receiver i in

$$\Delta_{i,update} = T(n-1)\left[\frac{l(m+1)}{R_M}+\frac{H_k}{\lambda}\right]+\left[\frac{1}{p_i}-1\right]\frac{H_k}{\lambda}+\frac{1}{\lambda}, \tag{6.37}$$

i.e., when the last packet of update arrives at i, its age, hence the expected AoI of i becomes $\Delta_{i,update}$.

As the last packet of the last update has arrived at t_1, the next update's last packet arrives at $t_i+\Delta_{i,diff}+\Delta_{i,update}$ with an age $\Delta_{i,update}$. For receiver i, observe that the minimum of the expected AoI is $\Delta_{i,update}$ and the peak expected AoI is given as $\Delta_{i,diff}+2\Delta_{i,update}$, i.e., $\max_{i,t}\{\mathbb{E}[\Delta_i(t)]\}=\max_i\left(\Delta_{i,diff}+2\Delta_{i,update}\right)$ and this leads to the expression seen in equation (6.34) for the overall network. As the minimum expected AoI for receiver i is $\Delta_{i,update}$ and the peak expected AoI is $\Delta_{i,diff}+2\Delta_{i,update}$, the average results in $(\Delta_{i,diff}+3\Delta_{i,update})/2$. In total, the average expected AoI in the network is expressed as shown in equation (6.35). □

6.5.2. Peak and Average Expected AoI with Coding

As in the uncoded case, the significant problem is to send n packets successfully to k receivers in minimum number of transmissions in order to minimize the AoI. In this scenario, n packets are assumed to be coded employing random linear network

coding and the coded packet is multicasted to all receivers. When each and every receiver gets n coded packets with linearly independent encoding coefficient vectors, all receivers can successfully decode the packets. The most important question is how many coded packets a receiver needs for the perfect recovery of original ones.

Theorem 6.6. *The probability of perfect recovery of n original packets, $P_{pr,n}$ at receiver side given that n linearly coded packets have been received using randomly chosen coefficients from $\mathcal{Z}_q$, is given as $P_{pr,n} = \prod_{i=0}^{n-1}(1 - q^{i-n})$.*

Proof. Note that for the perfect recovery of n original packets, a node must receive at least n different coded versions of the original packets. Assuming random linear network coding and coefficients being sampled from a finite field $\mathcal{Z}_q$, being q prime, the question of perfect recovery is equivalent to the probability of an nxn matrix, where the entries are sampled from $\mathcal{Z}_q$, being full rank, i.e., having linearly independent columns.

Using induction, when we sample $\phi = 1$ vector, say v_1, and Ψ_i being the matrix of chosen first i vectors, the probability that matrix is full rank is

$$P(rank(\Psi_1) = 1) = 1 - q^{-n} \tag{6.38}$$

since the only problematic v_1 would be the all zero vector. Now, suppose we already have ϕ linearly independent vectors $\{v_i\}_{i=1}^{\phi}$, then the next vector $v_{\phi+1}$ to be chosen must not be in the span of the rest of ϕ vectors, $\{v_i\}_{i=1}^{\phi}$. The probability of choosing $v_{\phi+1}$ so that the linear independence is preserved is given as

$$P(rank(\Psi_{\phi+1}) = \phi + 1 | rank(\Psi_\phi) = \phi) = \frac{q^n - q^\phi}{q^n} = 1 - q^{\phi-n}. \tag{6.39}$$

Since $P_{pr,n} = P(rank(\Psi_n) = n)$ serves as the condition ensuring the perfect recovery of the original packets, it can be determined mathematically by evaluating

the conditional probabilities as

$$\begin{aligned} P_{pr,n} &= P(rank(\Psi_n) = n) = P(rank(\Psi_1) = 1) \\ &\times ... \times P(rank(\Psi_{\phi+1}) = \phi + 1 | rank(\Psi_\phi) = \phi) \\ &\times ... \times P(rank(\Psi_n) = n | rank(\Psi_{n-1}) = n - 1) \\ &= (1 - q^{-n}) \\ &\times ... \times (1 - q^{\phi - n}) \\ &\times ... \times (1 - q^{-1}) \\ &= \prod_{i=0}^{n-1} (1 - q^{i-n}). \end{aligned} \tag{6.40}$$

which provide insights into the likelihood of achieving a rank n for the received matrix Ψ_n under various transmission and channel conditions. □

The behavior of the probability of the perfect recovery (named as "Objective Value" in Fig. 6.3) with respect to the values of q and n is seen in Fig. 6.3 where $n \in [2, 100]$ and $q \in [16, 512]$. From the simulations, the most important thing to note is that probability of recovery insufficiency is below 10^{-3} when $q > 32$ and below 10^{-6} when $q > 100$ regardless of the value of n. We conclude, by suitably adjusting the cardinality, q, of the finite field $\mathcal{Z}_q$, it is observed that having n coded packets is enough for the perfect recovery of the original ones with very high probability, ~ 1, used in the remaining parts.

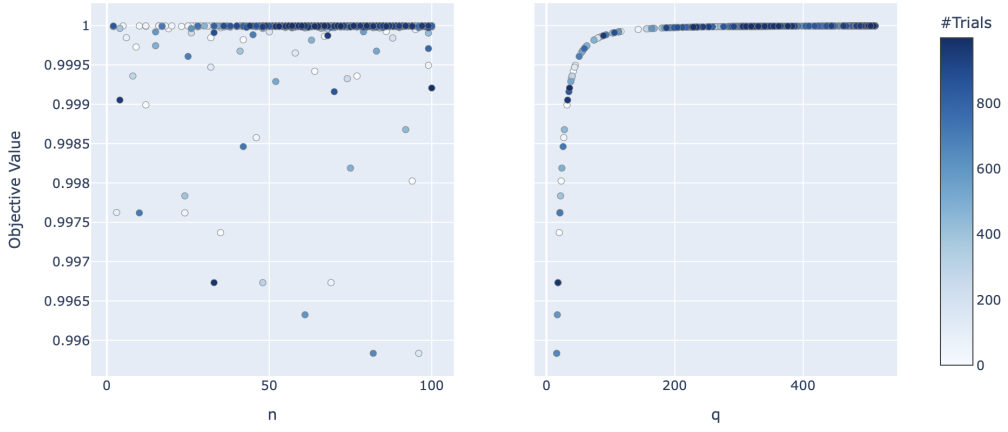


Figure 6.3. Probability of perfect recovery with respect to q and n.

In the case of coded flows, T_{coded} denotes the expected number of transmissions to deliver at least n coded packets to k receivers. As in the uncoded case, problem of finding T_{coded} can be modeled adopting k independent Bernoulli trials with success probabilities are $p_1, ..., p_k$ respectively where p_i denotes the success probability of a transmission from B to R_i, $i \in \{1, 2, ..., k\}$. The aim is to find the expected number of transmissions where all k receivers have successfully got at least n coded packets. For receiver R_i, the probability of successfully receiving at least n packets in β transmissions can be mathematically expressed as

$$P_{R_i}^{c-suc}(\beta) = \begin{cases} 1 - \sum_{\gamma=0}^{n-1} P(\gamma, \beta, p_i), & \text{if } \beta \geq n \\ 0, & \text{otherwise} \end{cases} \tag{6.41}$$

given that the expression $P(\gamma, \beta, p_i)$ is provided as

$$P(\gamma, \beta, p_i) = \binom{\beta}{\gamma} p_i^{\gamma} (1 - p_i)^{\beta - \gamma}. \tag{6.42}$$

For all receivers, the probability that at least n encoded packets have been successfully received in β multicast transmissions is given by

$$P_{R_1,...,R_k}^{c-suc}(\beta) = \begin{cases} \prod_{i=1}^{k} \left[1 - \sum_{\gamma=0}^{n-1} P(\gamma, \beta, p_i)\right], & \text{if } \beta \geq n \\ 0, & \text{otherwise} \end{cases} \tag{6.43}$$

whereas the probability of failure is found as

$$P_{R_1,...,R_k}^{c-fail}(\beta) = \begin{cases} 1 - P_{R_1,...,R_k}^{c-suc}(\beta), & \text{if } \beta \geq n \\ 1, & \text{otherwise.} \end{cases} \tag{6.44}$$

As a result, the expected number of multicast transmissions, T_{coded}, required for all receivers to obtain at least n coded packets becomes

$$\begin{aligned} T_{coded} &= \sum_{\beta=0}^{\infty} P_{R_1,...,R_k}^{c-fail}(\beta) \\ &= n + \sum_{\beta=n}^{\infty} P_{R_1,...,R_k}^{c-fail}(\beta) \end{aligned} \tag{6.45}$$

given that all the necessary variables to calculate T_{coded} are provided in advance.

Theorem 6.7. *When coding is implemented, the peak expected AoI of the network is given as*

$$\begin{aligned}&\max_{i,t}\{\mathbb{E}[\Delta_{i,coded}(t)]\}\\&=\max_{i}\Bigg(\left[T_{coded}+\frac{n}{p_i}-2\right]\left[\frac{(l(n-1)+n)(m+1)}{C_{comp}}+\frac{(l+n)(m+1)}{R_M}+\frac{H_k}{\lambda}\right]\\&+\frac{2\left(\mathcal{O}(n^3)+n^2l+(n-1)nl\right)}{C_{comp}}+\frac{H_k+1}{\lambda}\Bigg)\end{aligned}\tag{6.46}$$

where $\max_{i,t}$ *is the maximum in time and among nodes. On the other hand, average expected AoI of the network is found as*

$$\begin{aligned}&\frac{1}{k}\sum_{i=1}^{k}\lim_{T\to\infty}\frac{1}{T}\int_0^T\mathbb{E}[\Delta_{i,coded}(t)]dt\\&=\frac{1}{k}\sum_{i=1}^{k}\Bigg(\left[\frac{T_{coded}-3}{2}+\frac{n}{p_i}\right]\left[\frac{(l(n-1)+n)(m+1)}{C_{comp}}+\frac{(l+n)(m+1)}{R_M}+\frac{H_k}{\lambda}\right]\\&+\frac{3\left(\mathcal{O}(n^3)+n^2l+(n-1)nl\right)}{2C_{comp}}+\frac{H_k+2}{2\lambda}\Bigg).\end{aligned}\tag{6.47}$$

Proof. In order to find the average expected AoI in the network, first consider an arbitrary receiver i. Assume that receiver i gets the last packet of an update at t_i. Then, the time it takes for source to complete update's packets distribution to all receivers and then generate a new update is denoted as $\Delta_{i,diff}^{coded}$ with a closed form expression seen as

$$\begin{aligned}\Delta_{i,diff}^{coded}=&\left[T_{coded}-\frac{n}{p_i}\right]\left[\frac{(l(n-1)+n)(m+1)}{C_{comp}}+\frac{(l+n)(m+1)}{R_M}+\frac{H_k}{\lambda}\right]\\&+\frac{H_k-1}{\lambda}\end{aligned}\tag{6.48}$$

since receiver i needs $\frac{n}{p_i}$ transmissions in expectation to successfully get all n packets of the update and the length of a packet becomes $l+n$ due to insertion of coding coefficients and the term including C_{comp} resembles the network coding operation in the transmitter side as explained in Section 6.3. Note that in this step, decoding times for receiver i and the last receiver having the last update cancel each other. After the generation of the new update, the last packet of this update arrives and is decoded at

receiver i in an interval given as

$$\Delta_{i,update}^{coded} = \left[\frac{n}{p_i} - 1\right]\left[\frac{(l(n-1)+n)(m+1)}{C_{comp}} + \frac{(l+n)(m+1)}{R_M} + \frac{H_k}{\lambda}\right] + \frac{1}{\lambda} + \frac{\mathcal{O}(n^3) + n^2 l + (n-1)nl}{C_{comp}}, \tag{6.49}$$

when the last packet of update arrives at i, update's age, hence the expected AoI of i becomes $\Delta_{i,update}^{coded}$. Note that the last term provides the operational complexity of decoding of n message packets at the receiver side where $\mathcal{O}(n^3)$ accounts for the $n \times n$ matrix inversion with Gauss–Jordan elimination and $n^2 l + (n-1)nl$ accounts for the multiplication of matrices with $n \times n$ and $n \times l$ sizes.

As the last packet of the last update has arrived at t_i, the next update's last packet arrives at $t_i + \Delta_{i,diff}^{coded} + \Delta_{i,update}^{coded}$ with an age $\Delta_{i,update}^{coded}$. For receiver i, observe that the minimum of the expected AoI is $\Delta_{i,update}^{coded}$ and the peak expected AoI is given as $\Delta_{i,diff}^{coded} + 2\Delta_{i,update}^{coded}$, i.e., $\max_i \left(\Delta_{i,diff}^{coded} + 2\Delta_{i,update}^{coded}\right)$ which can be rewritten as in equation (6.46). As the minimum expected AoI for receiver i is $\Delta_{i,update}^{coded}$ and the peak expected AoI is $\Delta_{i,diff}^{coded} + 2\Delta_{i,update}^{coded}$, the average results in $(\Delta_{i,diff}^{coded} + 3\Delta_{i,update}^{coded})/2$. In total, the average expected AoI in the network is expressed as in equation (6.47). □

6.5.3. Comparison of Peak and Average Expected AoIs with or without Coding

We evaluate the behavior of proportional AoI gain, i.e.,

$$\gamma_k = \frac{\max_{i,t}\{\mathbb{E}[\Delta_i(t)]\} - \max_{i,t}\{\mathbb{E}[\Delta_{i,coded}(t)]\}}{\max_{i,t}\{\mathbb{E}[\Delta_{i,coded}(t)]} \tag{6.50}$$

representing the proportional time difference of uncoded multicast and coded multicast transmissions for k users with respect to parameters of concern. Behaviors of γ_k with respect to network parameters are presented in Section 6.6.

6.6. Behavioral Analyses

Lemma 6.8. *(Effect of Parameters): Proportional AoI gains for both 2-users and k-users, γ and γ_k are increasing in C_{comp}, whereas they are decreasing in R_τ. On the other hand, γ and γ_k are neither increasing nor decreasing in l, λ, q and p_τ.*

Proof. Taking the partial derivative of γ and γ_k with respect to l, λ, q and p_τ, the denominators are always positive whereas the numerators may either be positive or negative, i.e., γ and γ_k are neither increasing nor decreasing in l, λ, q and p_τ. For the ease of reading, these additional eight expressions are not provided here.

Taking the partial derivative of γ with respect to C_{comp}, we get

$$\frac{\partial \gamma}{\partial C_{comp}} = \frac{16R_\tau l\left(m+1\right)\left(p_\tau-2\right)^2 p_\tau \lambda\left(l\left(m+1\right)\lambda+R_\tau\right)}{\left(\left(2p_\tau-3\right)\left(2l\left(m+1\right)\lambda+3R_\tau\right)C_{comp}+4R_\tau l\left(m+1\right)\left(p_\tau-2\right)p_\tau\lambda\right)^2}. \quad (6.51)$$

Note that denominator is always positive. Also, we observe that numerator is always positive, so that $\frac{\partial \gamma}{\partial C_{comp}} > 0$, i.e., γ is always increasing in C_{comp}.

Taking the partial derivative of γ with respect to R_τ, we get

$$\frac{\partial \gamma}{\partial R_\tau} = \frac{-4C_{comp} l\left(m+1\right)\left(p_\tau-2\right)\lambda\left(4l\left(m+1\right)\left(p_\tau-2\right)p_\tau\lambda+2C_{comp}p_\tau-3C_{comp}\right)}{\left(\left(4l\left(m+1\right)\left(p_\tau-2\right)p_\tau\lambda+6C_{comp}p_\tau-9C_{comp}\right)R_\tau+2C_{comp}l\left(m+1\right)\left(2p_\tau-3\right)\lambda\right)^2}. \quad (6.52)$$

Note that denominator is always positive whereas numerator is always negative, so that $\frac{\partial \gamma}{\partial R_\tau} < 0$, i.e., γ is always decreasing in R_τ.

For the case involving k users, the expressions for the derivatives can be found respectively in

$$\frac{\partial \gamma_k}{\partial C_{comp}} = \frac{\frac{(m+1)(n+l(n-1))\xi_1}{C_{comp}^2}+\frac{\xi_3}{C_{comp}}}{\xi_1\xi_2+\xi_3+\xi_4} + \frac{\left(\frac{(m+1)(n+l(n-1))\xi_1}{C_{comp}^2}+\frac{\xi_3}{C_{comp}}\right)\left(\xi_5\xi_6+\xi_7-\xi_1\xi_2-\xi_3\right)}{\left(\xi_1\xi_2+\xi_3+\xi_4\right)^2}, \quad (6.53)$$

$$\frac{\partial \gamma_k}{\partial R_\tau} = \frac{\left(m+1\right)\left(n+l\right)\xi_1\left(\xi_5\xi_6+\xi_7-\xi_3-\xi_1\xi_2\right)}{\left(\xi_1\xi_2+\xi_3+\xi_4\right)^2 R_\tau^2} + \frac{\frac{(m+1)(n+l)(\xi_1)}{R_\tau^2}-\frac{l\cdot(m+1)\xi_5}{R_\tau^2}}{\xi_1\xi_2+\xi_3+\xi_4}. \quad (6.54)$$

These equations are highly intricate and difficult to interpret, presenting a significant challenge for anyone attempting to follow the detailed calculations. To facilitate both the comprehension and the presentation of these complex expressions, we have undertaken a reorganization of the variable symbols. The revised notation should help in reducing the cognitive load on the reader, allowing for a clearer understanding of the underlying mathematical concepts. The new assignments can be provided as

$$\xi_1 = \frac{n}{p_i} + T_{coded} - 2, \tag{6.55a}$$

$$\xi_2 = \frac{(m+1)\,(n + l\,(n-1))}{C_{comp}} + \frac{H_k}{\lambda} + \frac{(m+1)\,(n+l)}{R_\tau}, \tag{6.55b}$$

$$\xi_3 = \frac{2\,(n^3 + ln^2 + l\,(n-1)\,n)}{C_{comp}}, \tag{6.55c}$$

$$\xi_4 = \frac{H_k + 1}{\lambda}, \tag{6.55d}$$

$$\xi_5 = T\,(2n-1) - \frac{1}{p_i}, \tag{6.55e}$$

$$\xi_6 = \frac{H_k}{\lambda} + \frac{l\,(m+1)}{R_\tau}, \tag{6.55f}$$

$$\xi_7 = \frac{H_k\left(\frac{2}{p_i} - 1\right) - H_k}{\lambda}. \tag{6.55g}$$

The important thing here to note is that $i = \arg\max_i \mathbb{E}[\Delta_{i,coded}(t)]$ for equation (6.55a) and $i = \arg\max_i \mathbb{E}[\Delta_i(t)]$ for equations (6.55e) and (6.55g). $\frac{\partial \gamma_k}{\partial C_{comp}}$ and $\frac{\partial \gamma_k}{\partial R_\tau}$ are not straightforward to evaluate, however the interested reader can interpret these expressions to confirm $\frac{\partial \gamma_k}{\partial C_{comp}} > 0$ and $\frac{\partial \gamma_k}{\partial R_\tau} < 0$ similar to findings in case of 2-receiver model. □

Table 6.1. Simulation parameters of IoT networks.

	p_τ	R_τ	C_{comp}	l	λ	q
Household	$[0.7, 1]$	10^6	10^6	256	$[1, 10^3]$	256
UAV	$[0.7, 1]$	10^9	10^{10}	256	$[1, 10^3]$	256
Military	$[0.2, 0.5]$	10^6	10^{10}	256	$[1, 10^3]$	256

6.7. Simulation Results

Majority of the potential use cases of the discussed network coding model, which include but not limited to industrial control for manufacturing jet engines, remote control in mining, remote bus driving, augmented reality, and smart harbor, etc., may be classified as shown in [69] in order to address the time-sensitivity of the networks, i.e., data freshness. Numerous real-world examples may well be provided such as i) multimedia IoT networks collecting data from sensors located at traffic lights, vehicles, and traffic cameras in order to perform vehicle accident prediction in near-real time (e.g., within 30 ms) [63, 64], ii) distributed measurement systems for time, mission, or safety-critical data traffic for Industry 4.0 [24, 65] and iii) healthcare event predictions and warnings for timely intervention [21, 67]. Furthermore, AoI minimization in household IoT [219], UAV IoT [118] and military IoT [220] have been considered significant and thus have been investigated here in order to observe the impact of network coding extensively. In the proposed scenarios, a sensor that continuously sends its updates to multiple sinks that are processing the update information is considered in AoI sense.

Monte Carlo simulations are performed to observe the general pattern of AoI values where the number of trials for each run is set 10000. The relevant network parameters are uniformly and independently sampled inside specified intervals in each trial. Parameter intervals set for each scenario may be seen in TABLE 6.1. The computation rates for IoT devices may vary in a wide range, from a few megahertz to a few gigahertz, based on the specific use case of the devices [221]. In this work, we consider two types of sensors; low capability sensors that have a processing capacity of one megabit per second and high capability devices that have a processing capacity in the order of ten gigabit per second. In addition, transmission rate of sensors are case-dependent and how they are chosen is explained in next subsections depending on the protocol employed in the network. Note that it is assumed all nodes are equivalent with a universal transmission rate and transmission success probabilities as R_τ and p_τ, respectively. As a note, in the legend of the figures, the computational method of the related curves may be found.

6.7.1. UAV IoT

The data rates of protocols in UAV IoT vary based on the application and data type. Wireless protocols like Wi-Fi, cellular networks, and satellite communication are used to transmit data to a ground station or remote server. Data rates can range from a few kilobits per second to hundreds of megabits per second for real-time video streaming. Protocol and data rate selection depend on factors like UAV distance from the ground station, data amount, reliability, and security needs. The potential protocols that may be adopted are Wi-Fi, cellular networks, LoRa and Zigbee. Here, we consider a UAV IoT network with high computational capabilities employing IEEE 802.11n protocol [222], a system with relatively high data rates. IEEE 802.11n protocol [222] introduces multiple-input and multiple-output (MIMO) technology to improve signal quality and range where the supported data rates vary depending on the specific configuration and number of spatial streams used and generally are around hundreds of megabits per second to a few gigabits per second and chosen accordingly as seen in Table 6.1.

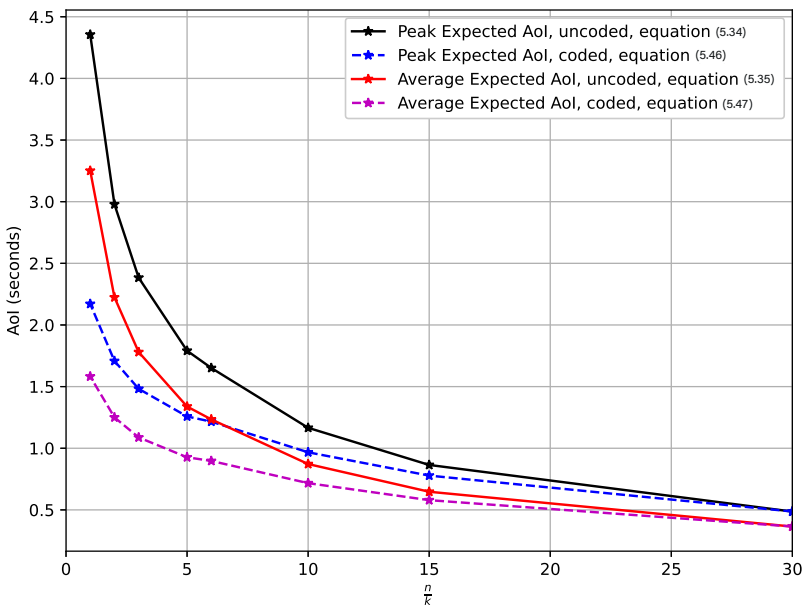


Figure 6.4. AoI measurements for a UAV IoT network ($n = 30$).

In Fig. 6.4, the peak and average expected AoI measurements are provided for uncoded and network coded cases with respect to $\frac{n}{k}$. Again, $n = 30$ is constant and k is the variable parameter. In this type of network, it is observed that network coding is always superior to no coding and they become approximately equal when there is only one receiver. This is not surprising as network coding exploits the multicast transmissions in case of multi receivers to outperform no coding in general. We conclude that network coding enhances the AoI performance, in general, in networks which consist of sensors with high computational and transmission rate capabilities.

6.7.2. Military IoT

In military IoT, low data rate protocols are combined with advanced algorithms and machine learning to process and analyze data locally on the device. This reduces transmitted data, improves network efficiency, and lowers latency. Military IoT devices can operate independently with local data processing, even in situations with communication disruptions. Low data rate protocols and high computational power allow for real-time and accurate performance of complex tasks such as object recognition, threat detection, and predictive maintenance. Here, we consider a military IoT network with high computational capabilities employing Bluetooth Low Energy (BLE) protocol [223, 224], a system with relatively low data rates. BLE sensors typically operate at a data rate of a megabit per second.

In Fig. 6.5, the peak and average expected AoI measurements are provided for uncoded and network coded cases with respect to $\frac{n}{k}$. As in prior simulations, $n = 30$ is constant and k is the variable parameter. Similar to the case of the UAV network, it is observed that network coding is always superior to no coding. In addition, it is clear that the AoI differences between coded and uncoded setups are even more greater, i.e., performing network coding decreases AoI measurements to the levels of one-third of uncoded flows. In short, network coding extremely improves the AoI performance, in general, in networks which consist of sensors with high computational resources and low transmission rate capabilities.

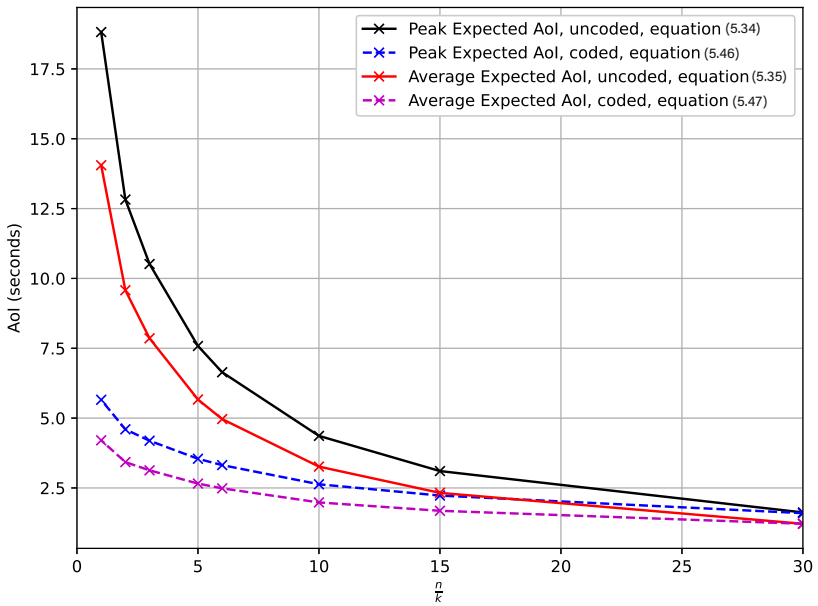


Figure 6.5. AoI measurements for a military IoT network ($n = 30$).

6.7.3. Household IoT

Household IoT enables remote control and monitoring of home features like lighting, temperature, security, and entertainment through smartphones or other devices. IoT protocols' data rates differ based on device and purpose. Smart thermostats and door locks may transmit small data volumes infrequently, while streaming media players and security cameras require higher data rates for continuous data flow. The protocols used to connect IoT devices to the internet, such as Wi-Fi, Bluetooth, Zigbee, and Z-Wave, also have different data rates and range capabilities [225]. Wi-Fi, for example, can provide high-speed data transmission but has a limited range, while Zigbee and Z-Wave have lower data rates but can transmit data over longer distances. The data rate requirements for household IoT devices depend on the specific use case and the desired level of performance and functionality. Here, we consider a household IoT network employing ZigBee protocol [226], a system with low data rates and low computational capabilities. Zigbee offers different data rates depending on the particular setup in use, but generally between a hundred of kilobits and a megabit per second.

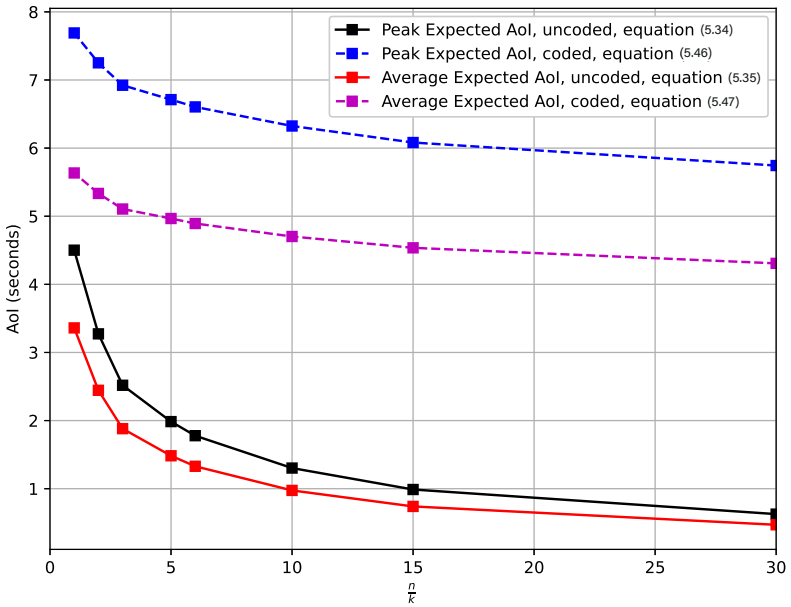


Figure 6.6. AoI measurements for a household IoT network ($n = 30$, $\frac{n}{k} \geq 1$).

In Fig. 6.6, the peak and average expected AoI measurements are provided for uncoded and network coded cases with respect to $\frac{n}{k}$, which represents the number of message packets to send per receiver. Here, $n = 30$ is constant and k is the variable parameter. In Fig. 6.6, it is seen that network coding is not necessary even though it outperforms no coding in UAV and military IoT networks. In general, it may be suitable for household IoT where the number of receivers (k) is relatively high and number of packets to be coded (n) are low. So it is in favor of network coding if $\frac{n}{k}$ gets smaller. In the case of household IoT, network coding starts to outperform no coding for $\frac{n}{k} < 0.2$ when $n = 30$ as observed in Fig. 6.7. Fig. 6.6 does not show the values for $\frac{n}{k} < 1$ since then it becomes unpleasantly dense. Hence, we address this in Fig. 6.7. Additionally, Fig. 6.8 shows that network coding starts to outperform no coding for $\frac{n}{k} \leq 0.5$ when $n = 5$. So, the main conclusion is network coding generally outperforms no coding but it is not always the case and it is parameter dependent especially when the computational power of sensors are low as may be in some household IoT networks.

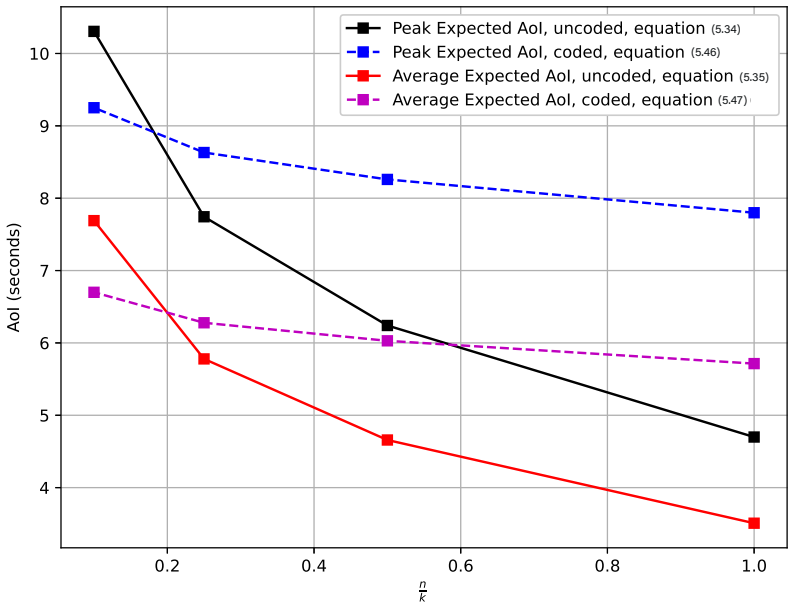


Figure 6.7. AoI measurements for a household IoT network ($n = 30$, $\frac{n}{k} \leq 1$).

6.7.4. A General Perspective

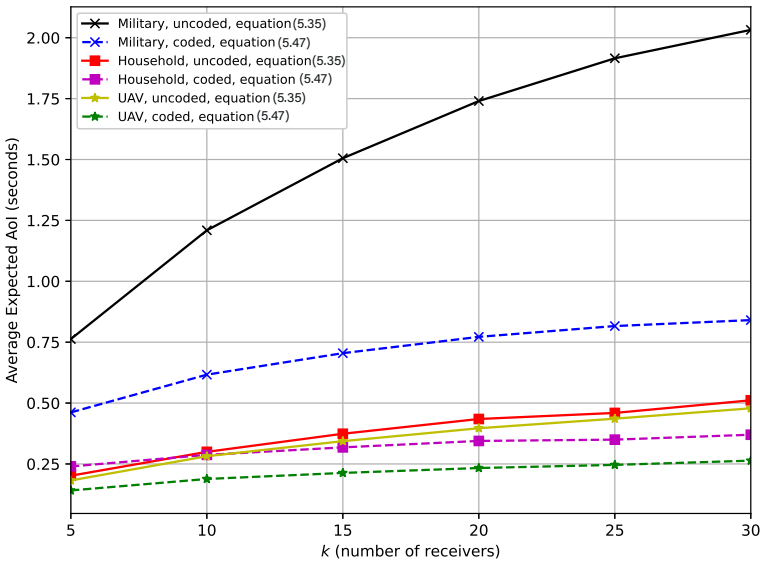
Fig. 6.8 and 6.9 provide us with a general understanding where the subfigures seen in Fig. 6.8 have k as the variable ($n = 5$) whereas the subfigures in Fig. 6.9 have n as the variable ($k = 5$). Fig. 6.8(a) and 6.9(a) present average expected AoI values, while Fig. 6.8(b) and 6.9(b) illustrate peak expected AoIs. In all plots, network coding helps to significantly lower AoI in UAV and military networks, where the improvement in the latter is obviously remarkable. However, in household IoT, while coding is the better option in terms of AoI for increasing k, the situation reverses when n is the increasing variable. This is because increasing n rapidly increases the computational complexity of dominant decoding operations and household sensors' scarce computational capacities immediately fall short to satisfy this rapidly increasing demand. In general, for all types of networks, note that increasing k favors network coding whereas increasing n favors no coding. It is crucial to consider these dynamics when designing and optimizing network protocols for different applications and network configurations.

Lastly, along the chapter, it is clear that the transmit power, which we do not consider in this chapter directly, should vary. A network coded packet needs a higher transmission power since it contains extra decoding coefficients leading to a higher number of bits to be transmitted. On the other hand, the total number of transmissions to convey all the update information is expected to be lower for a network coded system, i.e., $T_{coded} < nT$, where the values of variables T and T_{coded} can be calculated via equations (6.33) and (6.45). Therefore, even though packet-wise transmit power is expected to increase for the network coded case, the total transmit power is expected to decrease per update owing to the lower number of packets to be transmitted when coding is employed.

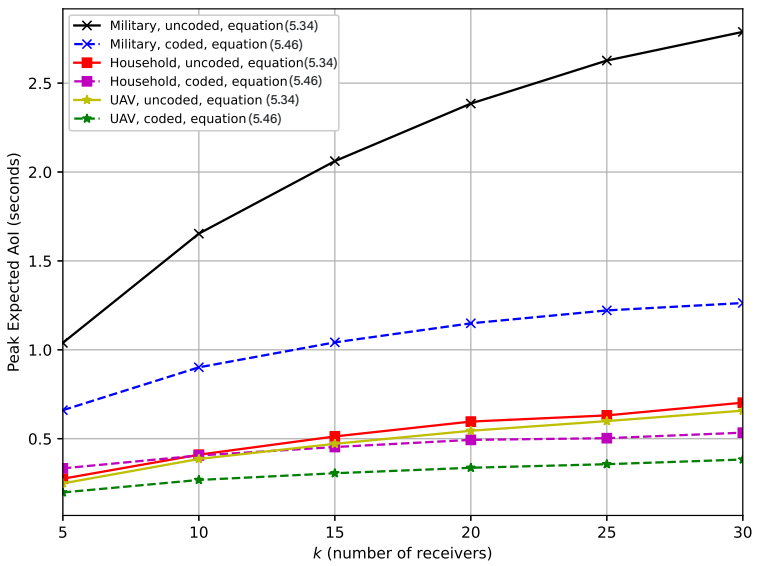
6.8. Conclusion

Previous works examining the impact of network coding on AoI did not consider the computational workload that arises from combining multiple data packets at sensor nodes. As AoI is tailored for IoT networks that may have sensors with limited computational resources, the computational complexity is an important factor to consider and the analyses neglecting it fall short to capture the overall effect of network coding.

In this chapter, we focused on multicast transmissions and quantified the peak and average expected AoI expressions for both uncoded and network coded transmissions where the computational burden of coding has been taken into acccount. Via extensive simulations, the key observations are twofold: i) network coding improves the AoI performance in networks which consist of sensors with high enough computational resources, and the advantage of network coding disappears if the computational capabilities of sensors in the considered network are very low, and ii) while increasing number of receivers favors the network coding to decrease the AoI levels, increasing number of packets to code has an opposite effect and increases the AoI measurements in considered networks.

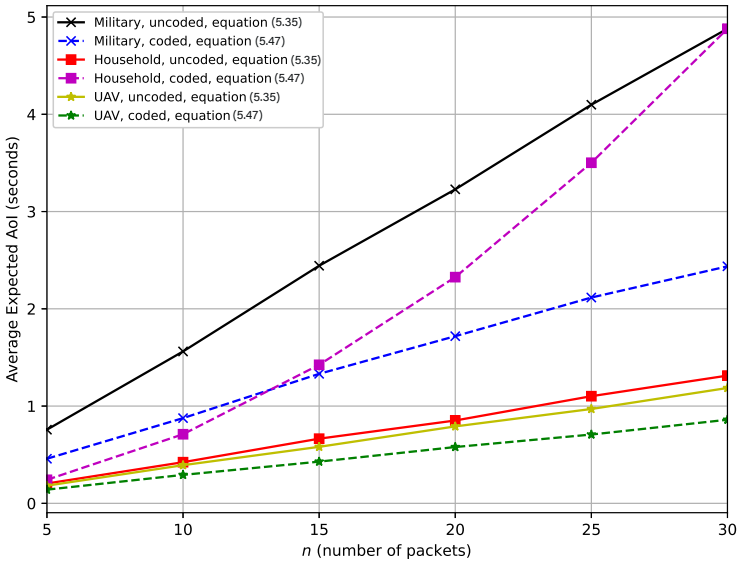


(a) Average AoI measurements with respect to number of receivers.

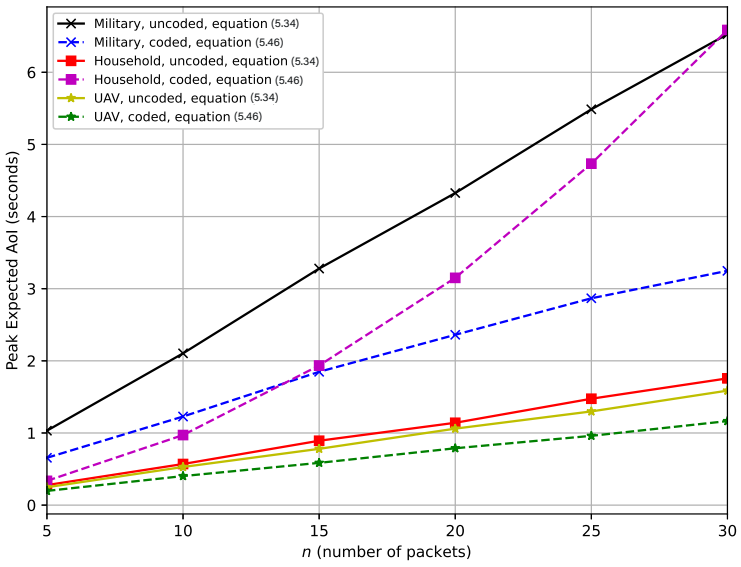


(b) Peak AoI measurements with respect to number of receivers.

Figure 6.8. AoI measurements for all scenarios ($n = 5$) where subfigure 6.8(a) shows average AoI values and subfigure 6.8(b) shows peak AoI values.



(a) Average AoI measurements with respect to number of packets.



(b) Peak AoI measurements with respect to number of packets.

Figure 6.9. AoI measurements for all scenarios ($k = 5$) where subfigure 6.9(a) shows average AoI values and subfigure 6.9(b) shows peak AoI values.

7. IMPACT OF NETWORK CODING ON AGE OF INFORMATION IN MULTI-SOURCE MULTI-HOP IOT NETWORKS

7.1. Introduction

In multi-hop and multi-source network model, data packets are delivered to the destination by being carried over multiple intermediate nodes or routers. In [48], the authors explore a discrete-time queueing model for multisource IoT systems, investigating AoI and peak AoI distributions for various queueing disciplines. In [111], the focus shifts to multi-packet reception scenarios, presenting a Whittle index-based scheduling policy that outperforms existing strategies. A similar network architecture is examined in [112], addressing AoI-aware multi-source information updates through efficient optimization algorithms. Additionally, joint AoI and energy consumption reduction for IoT device scheduling is considered in [113] by identifying optimal threshold-based policies where multiple information packets are necessary for a single status update.

In [227], the impact of network coding on AoI is investigated for a single source where the source is assumed to have a status update, which is divided into a number of packets, in order to forward to a set of monitors where multiple sources or transmission over multiple hops are not taken into account. However, in majority of the dense IoT deployments as given in [94], there are also multiple sources and multiple hops. Therefore the primary contribution of this chapter is to extend the analysis methodology of [227] to a generalized IoT multicast scenario with multiple sources, multiple hops and multiple receivers and to present analytical derivations for the AoI performance with and without network coding. The specific contributions can be summarized as follows:

- The derivations of average AoIs are provided for multi-source and multi-server first stage of transmissions where the transmissions are not coded.

- The derivations of average AoIs are provided for multi-server and multi-monitor second stage of transmissions. The comparison of AoIs with and without network coding is given. For both of the transmission stages, probabilistic open expressions are given instead of closed AoI expressions due to the complexity of the network model.
- Extensive numerical experiments are conducted and are shown to corroborate the theoretical findings which demonstrate significant AoI gains offered by network coding.

The rest of this chapter is organized as follows. In Section 7.2, system model and problem formulation are provided. In Section 7.3, AoI calculations are considered from simple network to a more general network. Furthermore, theoretical results and simulation results are presented for each hop. Simulation results and commentary discussions are provided in Section 7.4. Final remarks and open problems are discussed in Section 7.5.

7.2. System Model

Fig. 7.1 demonstrates a general multi-source and multi-hop network model consisting of sources (transmitters), servers, and monitors (receivers). Information sources I_i $(i = 1, 2, \ldots, N_t)$ make real-time observations from the environment. These independent observations are transmitted to the monitors M_k $(k = 1, 2, \ldots, N_r)$ through intermediate servers S_j $(j = 1, 2, \ldots, N_s)$ which are assumed to cooperate with each other in order to eliminate redundancy. Moreover, it is assumed that a status update is generated when all monitors receive all the packets from all information sources. If an observation has been successfully sent to a server, only this server transfer this observation to the monitors. Thus, the power consumption due to the successful transmission of the information from a source to all servers will be prevented.

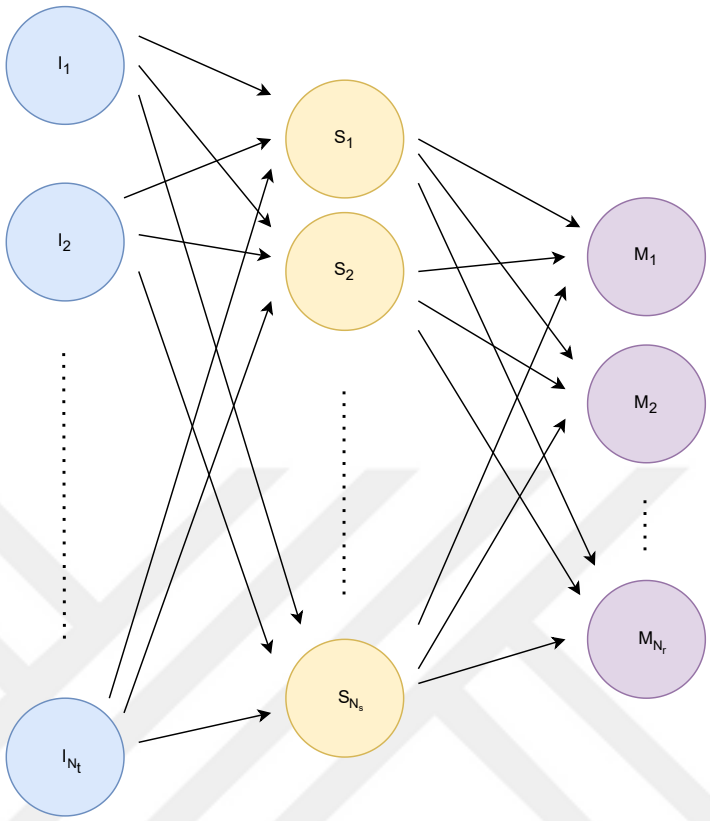


Figure 7.1. Multi-hop, multi-source network model.

There is a transmission success probability from the i-th source to the j-th server which is denoted by p_{ij}. Moreover, waiting time for a packet transmission from the i-th source to the j-th server follows exponential distribution with mean λ which is mathematically expressed as

$$f(t_{wait};\lambda) = \begin{cases} \lambda e^{-\lambda t_{wait}} & t_{wait} \geq 0, \\ 0 & t_{wait} < 0. \end{cases} \tag{7.1}$$

Similarly, transmission success probability from the j-th server to the k-th monitor is denoted by q_{jk} and arrival time for a successfully transmitted packet from the j-th source to monitor follows exponential distribution. In this work, success probabilities for all pairs are assumed to be identical for simplicity, *i.e.*, $p_{ij} = q_{jk} = p$ for $1 \leq i \leq N_t$, $1 \leq j \leq N_s$ and $1 \leq k \leq N_r$.

The transmission process from sources to monitors is assumed to occur in two stages. In the first stage, sources send their observations to the servers. In order to complete the first stage, all information must be sent successfully to one of servers. In the second stage, the servers start sending their packets to the monitors. After all the packets are transmitted successfully to the monitors, a status is updated. Then, the sources make observations and the first stage is started immediately.

A set, denoted as $P = \{p_1, p_2, \ldots, p_{N_t}\}$, is defined as the collection of all source packets. Also, the sets $P_1, P_2, \ldots, P_{N_s}$ are defined such that P_j contains the source packets that the j-th server will send to the monitor. Note that these subsets are disjoint, some of them might be empty sets and $\cup P_j = P$. For j-th server, the packets can be sent sequentially or network coded message can be sent. In this chapter, we will compare linear network coded transmission to uncoded transmission in terms of AoI performance at the monitor. For the j-th server, the encoded message can mathematically be expressed as

$$E_j = \sum_{i=1}^{s_j} f_i M_i \mod q \tag{7.2}$$

where s_j is the number of packets at the j-th server, f_i is encoding coefficient for packet p_i, $M_1, M_2, \cdots, M_{s_j}$ are information packets that will be sent from the j-th server. Encoding coefficients are randomly generated as stated in [214]. To decode encoded packets, the encoding coefficient should be known at the receiver side. Therefore, the encoding coefficients are embedded at the end of the message which is given as

$$E = [E_j, f_1, f_2, \cdots, f_{s_j}]. \tag{7.3}$$

In order to decode message with matrix operations, at least s_j number of encoded messages should be received successfully.

For non-negative integer m where $2^m < q \leq 2^{m+1}$, elements of $\mathcal{Z}_q$ field can be represented by $m + 1$ bits. In this work, we assume that the number of bits for the packets at each information source is identical and represented by l. Moreover, the transmission rate between any node is also identical and represented by R.

When the transmission of a packet is considered to be dependent on the packet's length and transmission rate, the required time for packet transmission is expressed as

$$\begin{aligned} t_\tau &= \frac{l(m+1)}{R}, \\ t_\tau^{NC} &= \frac{(l+n)(m+1)}{R}, \end{aligned} \tag{7.4}$$

where n is number of packets to be encoded, t_τ and t_τ^{NC} are transmission times for the strategies uncoded and network coded, respectively. Note that the final message is longer than the original packets. However, transmission rate is assumed to be sufficiently high, therefore, the transmission time increment due to the packet combination coefficients is negligible. Moreover, computational capacities of the servers and monitors are assumed to be sufficiently high. Therefore, the computation time for encoding and decoding is also negligible with respect to transmission time and waiting time. Thus, the main issue in this system model is decreasing the total number of transmission.

To compare the network coded and uncoded strategies, the average AoI is used as a performance metric and expressed as

$$\Delta = \frac{1}{T} \lim_{T \to \infty} \int_0^T \Delta(t) dt. \tag{7.5}$$

Note that the $\Delta(t)$ graph behavior appears as a sawtooth, therefore the average AoI is the middle point of the low-end and the high-end. Therefore, the average AoI can be mathematically re-formulated as

$$\Delta = 3 \frac{E[T_{stage_1}] + E[T_{stage_2}]}{2}. \tag{7.6}$$

7.3. AoI Analyses

It is already stated that the average AoI can be calculated by the passing time on the first and the second stages. In this section, the expected passing time for the first and second stages with and without network coding are calculated separately.

7.3.1. Transmission From Sources to Servers

To find the expected passing time for the first stage, three cases are investigated step-by-step as follows: 1) $N_t = N_s = 1$. 2) $N_t = 1$, $N_s > 1$. 3) $N_t, N_s > 1$.

7.3.1.1. $N_t = N_s = 1$. The probability of successful transmission at step k is mathematically expressed by the following

$$p(k) = (1-p)^{k-1}p. \tag{7.7}$$

At step k, the total waiting time is a random variable $T = \sum_{i=1}^{k} T_i$ where $T_i \sim \text{Exp}(\lambda)$ is the waiting time at step i. Sum of k independent exponential distribution with the same parameters is known as Erlang distribution and expressed as

$$f(t; k, \lambda) = \lambda^k \frac{t^{k-1}}{(k-1)!} e^{-\lambda t} \quad t, \lambda \geq 0. \tag{7.8}$$

As a result, the probability density function (PDF) of the total waiting time follows exponential distribution which is expressed as

$$\begin{aligned} f_T(t) &= \sum_{k=1}^{\infty} p(k) f(t; k, \lambda) \\ &= \sum_{k=1}^{\infty} (1-p)^{k-1} p\lambda^k \frac{t^{k-1}}{(k-1)!} e^{-\lambda t} \\ &= p\lambda e^{-\lambda t} \sum_{k=1}^{\infty} \frac{[(1-p)\lambda t]^{k-1}}{(k-1)!}. \end{aligned} \tag{7.9}$$

The summation corresponds the Taylor series expansion of $e^{(1-p)\lambda t}$. Then the PDF of the total waiting time is expressed as

$$f_T(t) = p\lambda e^{-\lambda t} e^{(1-p)\lambda t} = p\lambda e^{-p\lambda t}. \tag{7.10}$$

As a result, the total waiting time also follows an exponential distribution, *i.e.*, $T \sim \text{Exp}(p\lambda)$. Then, the expectation can be straightforwardly given as

$$E[T] = \frac{1}{p\lambda} \tag{7.11}$$

and when transmission time is included, the expected total passing time is given by

$$E[T] = \frac{1}{p}\left[\frac{1}{\lambda} + t_\tau\right]. \tag{7.12}$$

7.3.1.2. $N_t = 1$, $N_s > 1$. In this network model the source initiates multicast transmission of the message to all servers. The source then waits to receive an ACK or a NACK message from each server. If the source receives the NACK message from all servers, it restarts the multicast transmission. However, once the source receives an ACK message from any server indicating successful receipt of the packet, the source stops the multicast transmission. In this context, a random variable, denoted by T_i, is defined as the the total waiting time for the transmission from a source to all servers at the i-th try. Then, $T_i = \max(t_1, \cdots, t_j, \cdots, t_{N_s})$ where $t_j \sim \text{Exp}(\lambda)$. Then, the CDF of the random variable T_i can be expressed as

$$F_{T_i}(t, N_s) = P(t \geq t_1, t \geq t_2, \cdots, t \geq t_{N_s}), \tag{7.13}$$

$$F_{T_i}(t, N_s) = [F_T(t)]^{N_s}. \tag{7.14}$$

Here, $F_T(t) = 1 - e^{-\lambda t}$ is the CDF of the waiting time between each source and server pair. Because T_i is a non-negative random variable, the expected value of T_i can be calculated from its CDF as

$$E[T_i] = \int_0^\infty (1 - F_{T_i}(t, N_s))\, dt = \frac{H_{N_s}}{\lambda} \tag{7.15}$$

where $H_{N_s} = \sum_{k=1}^{N_s} \frac{1}{k}$ is the N_s-th harmonic number. Then, the expected passing time for this stage is given by

$$\begin{aligned} E[T] &= \sum_{k=1}^{\infty} p(k) k \left[E[T_i] + t_\tau \right] \\ &= \sum_{k=1}^{\infty} [(1-p)^{N_s}]^{k-1} k [1 - (1-p)^{N_s}] \left[\frac{H_{N_s}}{\lambda} + t_\tau \right]. \end{aligned} \tag{7.16}$$

On the other hand, for $0 < x < 1$

$$\sum_{k=1}^{\infty} x^{k-1} k = \frac{d}{dx}\left(\sum_{k=0}^{\infty} x^k\right) = \frac{d}{dx}\left(\frac{1}{1-x}\right) = \frac{1}{(1-x)^2}. \tag{7.17}$$

By substituting $x = (1-p)^{N_s}$, equation (7.17) can be written as

$$\sum_{k=1}^{\infty} [(1-p)^{N_s}]^{k-1} k = \frac{1}{[1-(1-p)^{N_s}]^2}. \tag{7.18}$$

Finally, by using equation (7.18), the total expected passing time for this stage on equation (7.16) is expressed as

$$E[T] = \frac{1}{1-(1-p)^{N_s}} \left[\frac{H_{N_s}}{\lambda} + t_\tau \right]. \tag{7.19}$$

7.3.1.3. $N_t, N_s > 1$. For a single source and multiple server case, the CDFs of the waiting time and passing time at step i are

$$\begin{aligned} F_{T_i}(t) &= (1 - e^{-\lambda t})^{N_s} \quad t \geq 0, \\ F_{T_i}(t) &= (1 - e^{-\lambda (t - t_\tau)})^{N_s} \quad t \geq t_\tau, \end{aligned} \tag{7.20}$$

respectively. Then, the PDF of the passing time for step i is expressed as

$$f_{T_i}(t) = N_s (1 - e^{-\lambda (t - t_\tau)})^{N_s - 1} \lambda e^{-\lambda (t - t_\tau)} \quad t \geq t_\tau. \tag{7.21}$$

Considering that the process ends at step k, the total passing time for the first source can be easily quantified as

$$T_{I_1,k}(t, k) = T_1 + T_2 + \cdots + T_k. \tag{7.22}$$

If the process is finished at step k, the PDF of the total passing time for the first source is convolution of k independent and identically distributed (i.i.d.) random variables which can be represented as

$$f_{T_{I_1},k} = f_{T_i}(t) * f_{T_i}(t) * \cdots * f_{T_i}(t) \tag{7.23}$$

where $*$ denotes the convolution operation. The PDF of the total passing time for the first source is mathematically found as

$$f_{T_{I_1}}(t) = \sum_{k=1}^{\infty} p(k) f_{T_{I_1},k}(t, k) \tag{7.24}$$

where $p(k) = (1 - p)^{N_s(k-1)}(1 - (1 - p)^{N_s})$. Equation (7.24) is valid for all sources because they are all identical with the same parameters.

A new random variable for the total passing time in the first stage is defined as the maximum transmission duration among all sources, i.e., $T_{stage_1} = \max(T_{I_1}, T_{I_2}, \cdots, T_{I_{N_t}})$.

The probability density function of the total passing time for the first source is mathematically calculated as

$$f_{T_{I_1}}(t) = \sum_{k=1}^{\infty} p(k) f_{T_{I_1},k}(t, k), \tag{7.25}$$

and the CDF of T_{stage_1} is expressed as

$$F_{T_{stage_1}}(t) = [F_{T_{I_1}}(t)]^{N_t}. \tag{7.26}$$

Because T_{stage_1} is a nonnegative random variable, the expected value of the total passing time for the first stage can be found as

$$E[T_{stage_1}] = \int_0^\infty (1 - F_{T_{stage_1}}(t))\, dt. \tag{7.27}$$

7.3.2. Uncoded Transmission From Servers to Monitors

To find the expected passing time for the second stage, two cases are investigated step-by-step as follows: 1) $N_s = 1, N_r > 1$. 2) $N_s, N_r > 1$.

7.3.2.1. $N_s = 1, N_r > 1$. Similar to first stage, when there is a single server and the number of monitor is N_r, the CDF and PDF of the passing time at step i are given as

$$\begin{aligned} F_{T_i}(t) &= (1 - e^{-\lambda(t-t_\tau)})^{N_r} \quad t \geq t_\tau, \\ f_{T_i}(t) &= N_r(1 - e^{-\lambda(t-t_\tau)})^{N_r-1}\lambda e^{-\lambda(t-t_\tau)} \quad t \geq t_\tau, \end{aligned} \tag{7.28}$$

respectively. If the process is finished at step k, the total passing time for the first server can be expressed as

$$T_{S_1,k}(t,k) = T_1 + T_2 + \cdots + T_k. \tag{7.29}$$

The PDF of the total passing time for the first server is the convolution of k i.i.d. random variables which is expressed as

$$f_{T_{S_1},k} = f_{T_1}(t) * f_{T_1}(t) * \cdots * f_{T_1}(t). \tag{7.30}$$

If a monitor has a correct packet, then the server is in a "success state". Conversely, if a monitor has no correct packet, then the server is said to be in a "failure state". In order for all servers to be in a success state at the k-th step, there should be some monitors which are in a failure state at the $(k-1)$-th step. To finish the multicasting at the k-th step, the probability should be such that

$$\begin{aligned} p(k) &= \sum_{\alpha=1}^{N_r} \binom{N_r}{\alpha} \left[(1-p)^{k-1}\right]^\alpha \left[1 - (1-p)^{k-1}\right]^{N_r-\alpha} p^\alpha \\ &= \sum_{\alpha=1}^{N_r} \binom{N_r}{\alpha} \left[(1-p)^{k-1}p\right]^\alpha \left[1 - (1-p)^{k-1}\right]^{N_r-\alpha}. \end{aligned} \tag{7.31}$$

The PDF of the total passing time for the first server is calculated as

$$f_{T_{S_1;1}}(t) = \sum_{k=1}^{\infty} p(k) f_{T_{S_1},k}(t,k) \tag{7.32}$$

where $p(k)$ is expressed in equation (7.31).

7.3.2.2. $N_s, N_r > 1$. The PDF of the total passing time for a single packet under a single server and multiple monitor scenario has been calculated in the subsection 7.3.2.1. Considering that the number of packets to be transmitted is s_1 for the server S_1, the random variable for total passing time for the first server is expressed as

$$T_{S_1;1:s_1} = T_{S_1;1} + T_{S_1;2} + \cdots + T_{S_1;s_1}. \tag{7.33}$$

Because the process is independent and identical for all packet transmission, the PDF of the total waiting time for the first server and s_1 packet can be calculated by the convolution of the s_1 i.i.d. random variables, which can be expressed as

$$f_{T_{S_1;1:s_1}}(t) = f_{T_{S_1;1}}(t) * f_{T_{S_1;1}}(t) * \cdots * f_{T_{S_1;1}}(t). \tag{7.34}$$

Similar calculations can be generated for the PDF of the total passing time at server S_i, *i.e.*, $f_{T_{S_i;1:s_i}}(t)$. Then the random variable for the total passing time for a given number of packets is expressed as

$$T_{stage_2,\mathrm{UC}} = \max(T_{S_1;1:s_1}(t), \cdots, T_{S_{N_s};1:s_{N_s}}(t)). \tag{7.35}$$

Then, the CDF of the total passing time for a given number of packets is found as

$$F_{T_{\mathrm{stage}_2},\mathrm{UC};s_1,\cdots,s_{N_s}}(t) = \prod_{i=1}^{N_s} F_{T_{S_i;1:s_i}}(t). \tag{7.36}$$

The expected total passing time for a given number of packets for uncoded transmission is calculated as

$$E[T_{\mathrm{stage}_2,\mathrm{UC};s_1,\cdots,s_{N_s}}] = \int_0^{\infty} (1 - F_{T_{\mathrm{stage}_2},\mathrm{UC};s_1,\cdots,s_{N_s}}(t))\, dt \tag{7.37}$$

due to the fact that the random variable is non-negative. Then, the expected total passing time for the second stage can be calculated as

$$E[T_{\mathrm{stage}_2,\mathrm{UC}}] = \sum_{\sum s_i = N_t} \frac{N_t!}{\prod s_i!} \frac{1}{{N_s}^{N_s}} E[T_{\mathrm{stage}_2,\mathrm{UC};s_1,\cdots,s_{N_s}}]. \tag{7.38}$$

7.3.3. Network Coded Transmission From Servers to Monitors

Network coding changes the transmission time for a single packet due. Therefore, when there is a single server and the number of monitor is N_r, the CDF and PDF of the passing time at step i are given as

$$\begin{aligned} F_{T_i}(t) &= (1 - e^{-\lambda(t-t_\tau^{NC})})^{N_r} \quad t \geq t_\tau^{NC}, \\ f_{T_i}(t) &= N_r(1 - e^{-\lambda(t-t_\tau^{NC})})^{N_r-1}\lambda e^{-\lambda(t-t_\tau^{NC})} \quad t \geq t_\tau^{NC}, \end{aligned} \tag{7.39}$$

respectively. To decode information packets successfully, at least s_1 correct packets should be received from all monitors. If a monitor receives s_1 correct packets, it is in the success state. After k steps, the probability of all monitors to be in the success state should be determined. The probability of all the monitors being in the success state in step k can be calculated as

$$\begin{aligned} p'(s_1, k, N_r) &= \left[\sum_{t=s_1}^{k} \binom{k}{t} p^t (1-p)^{k-t}\right]^{N_r} \\ p'(s_1, k, N_r) &= [1 - F(s_1 - 1, k, p)]^{N_r} \end{aligned} \tag{7.40}$$

where $F(\cdot)$ represents the CDF of the binomial distribution. Then, the probability of finishing the second stage at the k-th transmission can be expressed as

$$p(s_1, k) = p'(s_1, k, N_r) - p'(s_1, k-1, N_r). \tag{7.41}$$

The random variable for the total passing time after k steps and its PDF can mathematically be described as

$$\begin{aligned} T_{S_1,k}(t) &= T_1 + T_2 + \cdots + T_k, \\ f_{T_{S_1},k}(t) &= f_{T_1}(t) * f_{T_1}(t) * \cdots * f_{T_1}(t), \end{aligned} \tag{7.42}$$

respectively. The PDF of the total passing time for the first server is given as

$$f_{T_{S_1;1:s_1}}(t) = \sum_{k=1}^{\infty} p(s_1, k) f_{T_{S_1},k}(t). \tag{7.43}$$

Similarly, the PDF of the total passing time for the server S_i is denoted by $f_{T_{S_i;1:s_i}}(t)$. Then the random variable for the total passing time for the second stage is given as

$$T_{stage_2,\mathrm{NC}} = \max(T_{S_1;1:s_1}(t), \cdots, T_{S_{N_s};1:s_{N_s}}(t)), \tag{7.44}$$

and consequently the CDF of the total passing time for second stage is expressed as

$$F_{T_{\text{stage}_2},\text{NC},s_1,\cdots,s_{N_s}}(t) = \prod_{i=1}^{N_s} F_{T_{S_i;1:s_i}}(t). \tag{7.45}$$

Due to the non-negativeness of the random variable, the expected passing time for the second stage with network coded transmission can be calculated as

$$E[T_{\text{stage}_2,\text{NC};s_1,\cdots,s_{N_s}}] = \int_0^\infty (1 - F_{T_{\text{stage}_2},\text{NC},s_1,\cdots,s_{N_s}}(t))\, dt. \tag{7.46}$$

As a result, the expected total passing time for the second stage is calculated as

$$E[T_{\text{stage}_2,\text{NC}}] = \sum_{\sum s_i = N_t} \frac{N_t!}{\prod s_i!} \frac{1}{{N_s}^{N_s}} E[T_{\text{stage}_2,\text{NC};s_1,\cdots,s_{N_s}}] \tag{7.47}$$

where $i = 1, 2, \cdots, N_s$.

7.4. Numerical Results

Monte Carlo simulations have been conducted to observe the average AoI performance. The expected waiting time for each packet is chosen as $\frac{1}{\lambda} = 1$ s. The success probability for each packet is chosen as $p = \frac{1}{3}$, the length of each packet is defined as $l = 50$, the number of bits for each message is chosen as $m+1 = 8$ bits, the transmission rate is established as $R = 10^6$ bps, and the number of monitors is configured as $N_r = 3$. In Fig. 7.2, the average AoI measurement is provided for uncoded and network coded cases with respect to number of servers (N_s) for $N_t = 25, 50, \ldots 125$. When the number of servers increases, the average number of packets per server decreases, leading to a reduction in the average AoI. For small values of N_s, the impact of network coding is more noticeable. For instance, when $N_s = 10$ and $N_t = 100$, network coding reduces the AoI measurement from 235.40 seconds to 173.34 seconds which is approximately 26%. When $N_s = 40$ and $N_t = 100$, network coding reduces the AoI measurement from 120.11 to 97.26 which is approximately 19%. Moreover, while increasing the number of servers had a serious impact at first, its effect on AoI performance decreased in the later stages.

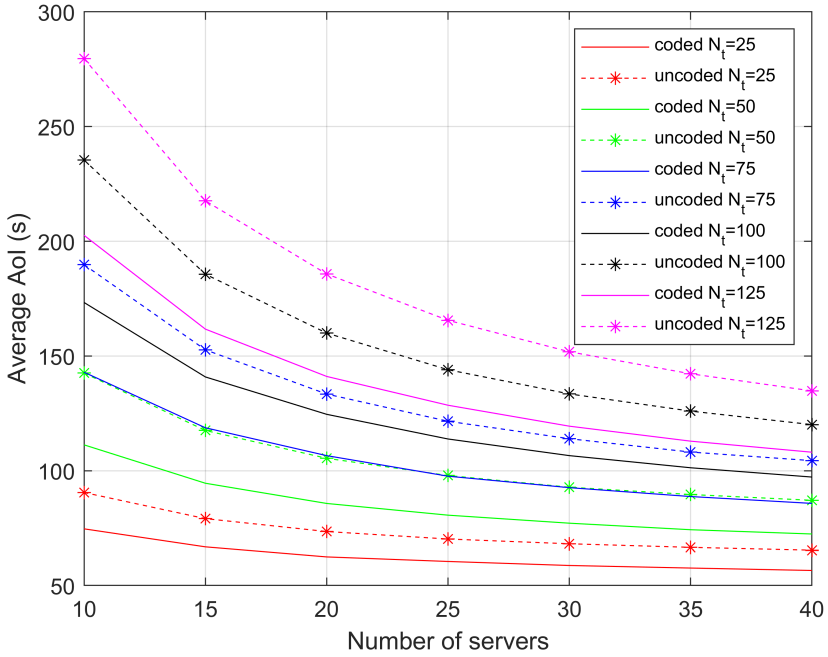


Figure 7.2. Number of servers vs average AoI.

To provide a more insightful interpretation of the results, it is crucial to examine the ratio between the average AoI in network coded transmission and uncoded transmission which is denoted as Δ_{NC}/Δ_{UC}. Fig. 7.3 illustrates how network coded transmission improves AoI performance compared to the uncoded transmission for varying values of the number of sources. Network coded transmission significantly improves performance when there are a larger number of sources. Conversely, when the number of sources is low, the improvement in AoI performance is smaller. The primary reason for this lies in the fact that when there are fewer sources, the number of packets delivered to the servers is less. Then, the probability of being able to encode and transmit packets becomes smaller. Conversely, when there is an abundance of sources, a higher number of packets are delivered to each servers. Therefore, the impact of network coding becomes more pronounced.

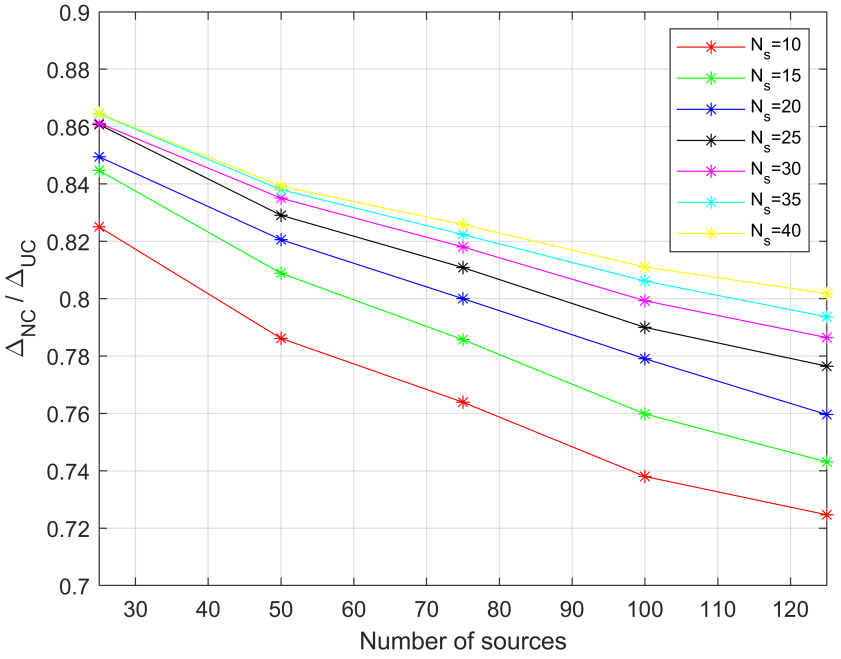


Figure 7.3. Number of sources vs $\frac{\Delta_{NC}}{\Delta_{UC}}$.

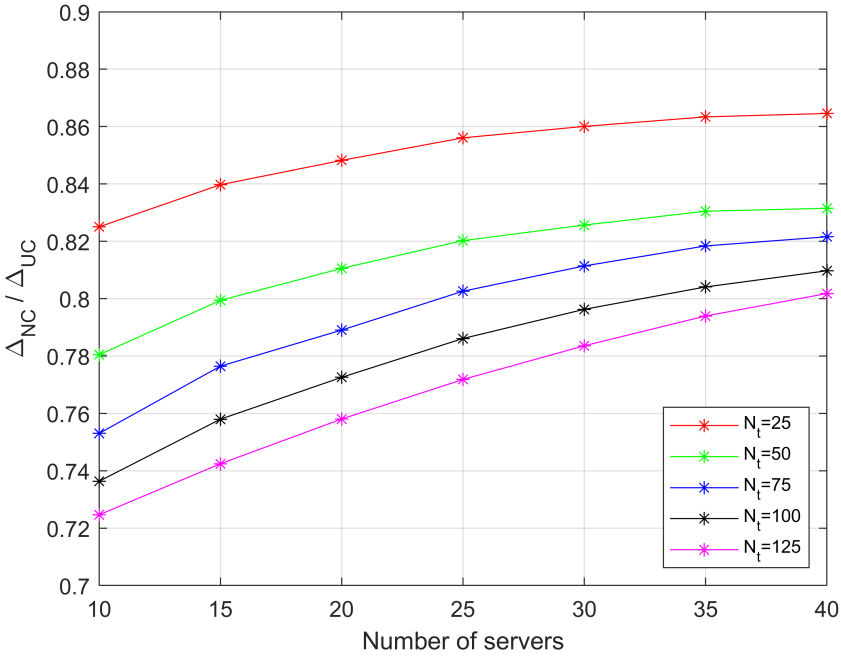


Figure 7.4. Number of servers vs $\frac{\Delta_{NC}}{\Delta_{UC}}$.

Fig. 7.4 shows how the ratio Δ_{NC}/Δ_{UC} varies with changes in the number of servers. As expected, with an increase in the number of servers, the number of packets per server decreases, resulting in a limited impact of network coding. For example, when there are 100 sources and 10 servers in use, network coding has improved the AoI performance by approximately 26%. However, when 40 servers are used, network coding has improved the AoI performance by approximately 19%. As a result, if there is a significantly large number of information sources in the environment but a relatively small number of servers transmitting data to monitors, using network coding becomes more appealing.

7.5. Conclusion

In this chapter, we investigated the impact of network coding on AoI performance in multi-source and multi-hop IoT networks. We derived average AoIs as probabilistic open expressions due to the complexity of the network model for the stages from sources to servers and servers to monitors. Moreover we provide a comparison of AoIs with and without network coding for the stage from servers to monitors. Finally, we demonstrate the improvement of AoI performance for various number of sources, servers and monitors when network coding is employed.

We have assumed that the internal nodes are half-duplex. In scenarios where they are full-duplex, meaning that the transmission does not wait for any stage to finish, determining the optimal number of packets to wait before initiating transmission becomes a significant problem in terms of AoI performance.

8. IMPACT OF NETWORK CODING ON AGE OF INFORMATION IN ENERGY HARVESTING ENVIRONMENT

8.1. Introduction

Numerous studies have examined the freshness of information in IoT networks in energy harvesting environments. Network coding fundamentally involves aggregating incoming message packets at intermediate nodes to generate outputs for transmission. This method offers enhanced efficiency and network resilience, as evidenced by works such as [34,35]. Unlike traditional networks where resources process and transmit data packets independently, network coding allows intermediate nodes to combine packets and transmit new ones. This approach boosts data rates by leveraging aggregated packet transmission and enhances network redundancy for improved reliability. However, these benefits come with increased computational complexity for network nodes.

In a previous work [227], we demonstrated that in various IoT deployments, the use of network coding for multicast transmissions can increase AoI performance, especially when sensors have high enough computational capabilities. In this chapter, we investigate the impact of network coding on AoI within an energy harvesting enabled environment. To achieve this goal, an energy harvesting model with two parameters, the probability of successful harvesting in a time slot (ω) and the amount of harvested energy (h), is employed. The expressions for peak and average expected AoIs for both uncoded and network coded transmissions, which are derived in [227], have been updated according to energy harvesting model. In short, examining the relationship between energy harvesting and network coding in terms of the AoI metric is the main objective of this chapter. The specific contributions can be itemized as follows:

- The modifications of peak and average expected AoIs are explained for general multicast transmissions with and without network coding with energy harvesting enabled employing a random walk procedure named here as energy walk.

- The interplay between network coding and energy harvesting has been illustrated for diverse IoT networks to clear the picture.

8.2. Energy Harvesting Model

We employ the same system model as in 6.3. In addition to this, the following energy harvesting model is adopted in this section.

In the energy harvesting process, a discrete model similar to the one in [53] is employed. The sender is assumed to rely on energy harvested from the surrounding environment. The sender stores the harvested energy in a battery of finite size Ψ (units of energy). Formally, let $\psi_B(t)$ denote the battery level of the sender B at the beginning of slot t. Thus, $\psi_B(t) \in \{0, ..., \Psi\}$.

We adopt a modified version of the assumption, as seen in [53, 228–232] where transmitting a status update from a sensor to receivers consumes one unit of energy per uncoded multicast packet and the consumption increases proportional to packet length in case of coded packets.

The sender B sends a packet if it has enough energy in its battery. We model the energy arrival at the sender as a Bernoulli process with intensity ω. This characterizes the discrete nature of energy arrivals in a slotted-time system, i.e., at each time slot, a sensor either harvests h unit of energy or not being $h \geq 1$. Let $\xi_B(t) \in \{0, h\}$, denote the energy arrival process of the sender B. Thus, the probability that B harvests h unit of energy during one time slot is ω, i.e., $\Pr\{\xi_B(t) = h\} = \omega$.

Let the random variable $\alpha_B(t) \in \{0, 1\}$ denote the action of the sender B at slot t: $\alpha_B(t) = 1$ if B multicasts to the receivers and $\alpha_B(t) = 0$ otherwise. Consequently, we get the following equality

$$\alpha_B(t) = \mathbb{I}_{\{\psi_B(t)+\xi_B(t)\geq 1\}}. \tag{8.1}$$

Note that the quantity $\alpha_B(t)$ also characterizes the energy consumption of sensor B at slot t. Finally, the evolution of the battery level of B is expressed as

$$\psi_B(t+1) = \min\{\max\{\psi_B(t) + \xi_B(t) - \alpha_B(t), 0\}, \Psi\}. \tag{8.2}$$

It is assumed that the same set of rules are valid for the case where network coding is enabled since $l \gg n$. Otherwise, since the transmission time is greater in case of coded packets, energy arrival would be also greater in the same proportion and again it would be valid to use the same energy harvesting model for both uncoded and coded transmissions.

8.3. AoI Analysis

In a previous work [227], we have provided the exact derivations of peak and average expected AoIs for the same system model with and without network coding. The peak and average expected AoI expressions for uncoded transmissions are shown in equations (6.34) and (6.35) whereas the coded counterparts may be seen in equations (6.46) and (6.47), respectively.

In order to observe the effect of energy harvesting, it is important to note that the energy harvesting model defined in Section 8.2 only has an impact on whether there will be any transmission in the next time slot or not. The other aspects of the problem remain unchanged. Therefore, the energy harvesting process can only potentially have an influence on the AoI expressions via the variables T and T_{coded}. It is clear that the sender B may not be able to transmit in some of the time slots due to insufficient battery condition and this will increase the variables T and T_{coded}. Owing to the assumptions presented in Section 8.2, the effect of energy harvesting comes out to be a coefficient, σ_f, which represents the expected number of energy failures per transmission trial. Then, it is enough to calculate $T/(1-\sigma_f)$ in order to find the expected number of transmissions to deliver a packet successfully to all receivers in the energy harvesting case. The same applies for the variable T_{coded}. Since the peak and average expected AoI expressions for uncoded and coded transmissions shown in equations (6.34), (6.35), (6.46) and (6.47) do not have a straightforward relationship

with the variables T and T_{coded}, there is not an easy way to show the effect of energy harvesting on the impact of network coding on AoI of the network. In other words, harvesting process definitely increases the values of all AoI expressions, however the relative impact is a question to be answered in this section. Finding σ_f, however, is not analytically straightforward. For this reason, we model the energy level of the sender B's battery as a random walk and then estimate the σ_f via Monte Carlo simulations. We call this random walk as "energy walk".

Energy walk may be illustrated as in Fig. 8.1. At time t, the remaining level of energy is given as $\psi_B(t)$. Using equation (8.2), it is observed that the energy level at time $t+1$, $\psi_B(t+1)$, becomes $\psi_B(t)-1$ with probability $1-\omega$ and $\psi_B(t)+h-1$ with probability ω. The relevant constraint is that the energy level must always be in the interval $[0, \Psi]$, i.e., $0 \leq \psi_B(t) \leq \Psi$, $\forall t$.

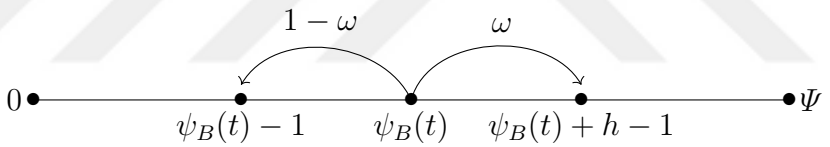


Figure 8.1. Illustration of energy walk.

Note that in the considered network, sender B transmits data packet in every time slot via multicasting. The only case that energy harvesting can affect the system performance is when the energy level of B is zero when it tries to transmit a packet. In order for this to happen, the condition is given as follows:

- $\psi_B(t) = 0$,
- $\xi_B(t) = 0$.

Given the above conditions are met, then we get $\psi_B(t+1) = 0$ using the equation (8.2) and B cannot transmit an update due to energy outage. It is important to note that even if $\psi_B(t) = 0$ and $\psi_B(t+1) = 0$, the packet transmission occurs if $\xi_B(t) = h$ in case $h = 1$. Therefore the two itemized conditions given above must be simultaneously satisfied in order to have an energy outage. The σ_f estimate is found via Algorithm 4.

In order to obtain σ_f values with respect to harvesting parameters $h \in [1, 5]$ and $\omega \in [0.1, 0.5]$, simulations with the following parameters have been conducted: the initial energy level has been set as $\psi_B(1) = 100$ and the total number of time slots to run the experiment has been set as $T = 100000$. The relationship between the parameters h, ω, and σ_f can be observed from Fig. 8.2 in the form of a surface plot.

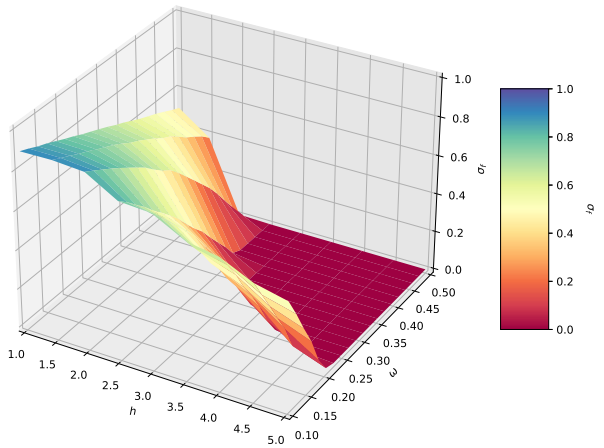


Figure 8.2. σ_f values with respect to harvested amount h and harvesting efficiency ω.

From the values of σ_f with respect to harvested energy amount and harvesting efficiency, the following relationships can be deducted:

- As h increases, holding ω constant, σ_f decreases. This indicates that increasing the harvesting potential h tends to reduce the average number energy outage in the battery as expected
- As ω increases, holding h constant, σ_f decreases. This suggests that increasing the probability of energy harvesting per time slot, ω, tends to reduce the average number energy outage in the battery as expected.
- Additionally, there seems to be an interaction between h and ω. For certain combinations of h and ω, the effect of changing one parameter on σ_f may be amplified or diminished by the other parameter.

Overall, the relationship between h, ω, and σ_f is crucial and a considerable amount of (h, ω) pairs do not result in any energy outage. Therefore, the resulting AoI in the system is very sensitive with respect to changes in parameter values.

Algorithm 4 Energy walk

Parameters:

Ψ : Maximum battery level

ω : Probability that energy is harvested

h : Magnitude of harvested energy

T : Number of time slots

Result: Expected number of energy failures per transmission: σ_f

function ENERGY_WALK(Ψ, ω, h, T)

  Initialize the battery level of the sensor: $\psi_B(1) \leftarrow 100$

  Initialize the number of energy failures: $e_f \leftarrow 0$

  for $t \leftarrow 1$ to T **do**

    if random() $< \omega$ **then**

      Move right: $\psi_B(t+1) \leftarrow \psi_B(t) + h - 1$

    else

      Move left: $\psi_B(t+1) \leftarrow \psi_B(t) - 1$

    end if

    if $\psi_B(t+1) \leq 0$ **then**

      $\psi_B(t+1) \leftarrow 0$

    end if

    if $\psi_B(t+1) \geq \Psi$ **then**

      $\psi_B(t+1) \leftarrow \Psi$

    end if

    if $(\psi_B(t) = 0) \wedge (\xi_B(t) = 0)$ **then**

      Increment e_f

    end if

  end for

  return e_f

Expected number of energy failures per transmission: $\sigma_f \leftarrow \frac{e_f}{T}$

In the next section, simulation results are provided for different type of networks and for different parameter sets to better observe the effect of energy harvesting on the measured AoI values.

8.4. Simulation Results

Monte Carlo simulations are conducted to examine the overall trends in AoI values, with each simulation comprising 50000 trials to sustain a smoother depiction. Network parameters are randomly sampled within predefined intervals for each trial. The specific parameter ranges for each scenario are outlined in TABLE 6.1. Given the diverse nature of IoT devices, their computational capabilities vary widely, spanning from a few megahertz to several gigahertz, depending on their intended application [221]. Moreover, the transmission rates of sensors are determined on a case-by-case basis, detailed in subsequent sections based on the network protocol in use. Also note that the legend of the figures provides information regarding the computational approach employed for the depicted curves.

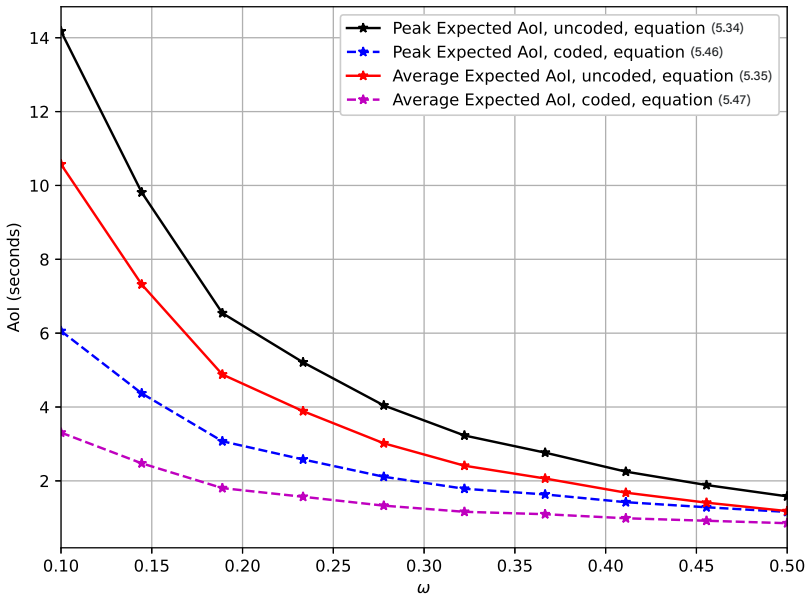


Figure 8.3. AoI measurements for a UAV IoT network ($n = 30$, $k = 5$).

Our first focus is on a high-capacity UAV IoT network utilizing IEEE 802.11n, renowned for its robust data rates, ranging from hundreds of megabits to gigabits per second, determined by specific MIMO configurations detailed in TABLE 6.1. In Fig. 8.3, the peak and average expected AoI measurements are provided for uncoded and network coded cases with respect to ω where $n = 30$, $k = 5$ and $h = 2$. It is observed that network coding is superior to no coding in general. Additionally, when the energy harvesting efficiency decreases, i.e. ω decreases, the gap between uncoded and coded transmissions increases leading to a conclusion that the effect of network coding is augmented in case of rather inefficient harvesting. This is a significant finding as this is not a straightforward implication which can be easily deducted. We conclude that network coding enhances the AoI performance, in general, and this enhancement even increases with decreasing energy harvesting efficiency in networks which consist of sensors with high computational and transmission rate capabilities.

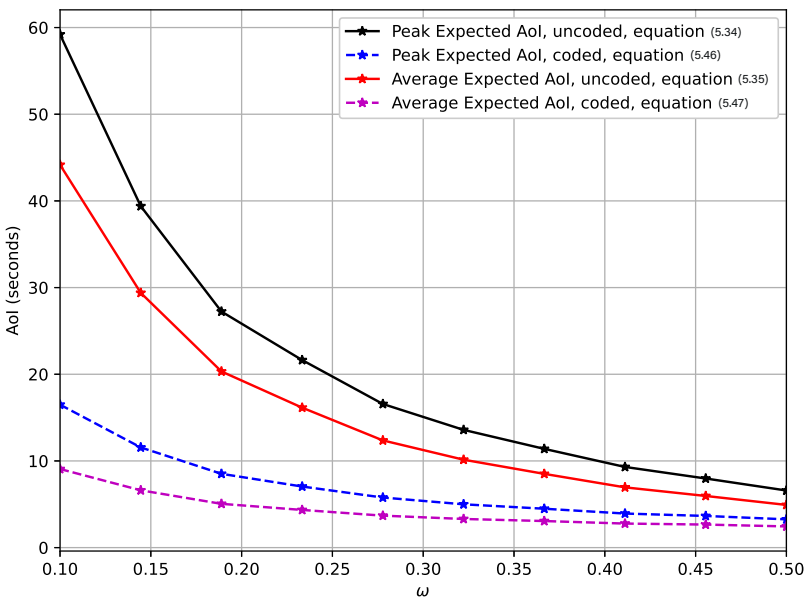


Figure 8.4. AoI measurements for a military IoT network ($n = 30$, $k = 5$).

Secondly, we consider military IoT which has the combination of low data rate protocols and powerful computational capabilities where our focus lies on a network leveraging Bluetooth Low Energy (BLE) protocol [223, 224], known for its modest data

rates, typically operating at around a megabit per second. In Fig. 8.4, the peak and average expected AoI measurements are provided for uncoded and network coded cases with respect to ω where $n = 30$, $k = 5$ and $h = 2$. Again, it is observed that network coding is superior to no coding in general. Additionally, as in the case for UAV IoT network, when the energy harvesting efficiency decreases, the gap between uncoded and coded transmissions increases leading to a conclusion that the effect of network coding is augmented in case of rather inefficient harvesting. The gap between the AoI of coded and uncoded transmissions is even broader in military IoT considered here. In short, network coding enhances the AoI performance, and this enhancement even increases with decreasing energy harvesting efficiency in networks which consist of sensors with high computational and low transmission rate capabilities.

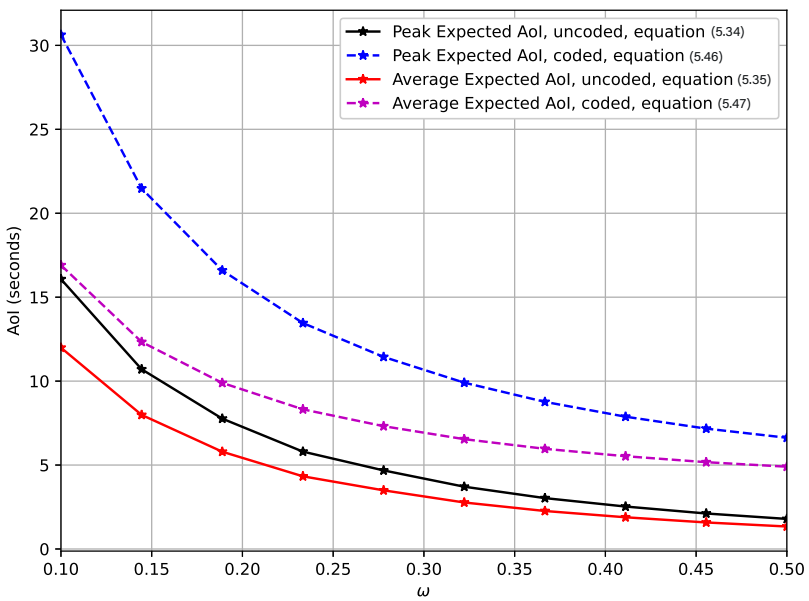


Figure 8.5. AoI measurements for a household IoT network ($n = 30$, $k = 5$).

Finally, a household IoT network using the ZigBee protocol [226], known for its lower data rates, is considered. In Fig. 8.5, the peak and average expected AoI measurements are provided for uncoded and network coded cases with respect to ω where $n = 30$, $k = 5$ and $h = 2$. Differently from other type of IoT networks, it is seen that network coding is unfavorable in terms of AoI for diverse energy harvesting

settings in household IoT. This is due to the insufficiency of the computational powers of nodes since coded transmission require a higher computational power due to the higher computational burden whereas uncoded transmissions need a higher data rate due to the higher number of potential transmissions. However, as in the other type of IoT, it can be stated that when the energy harvesting efficiency decreases, the coded transmission AoI to uncoded transmission AoI ratio gets smaller, i.e., harvesting inefficiences work relatively in favor of network coding. As seen, this is the case for all types of IoT networks considered in this chapter.

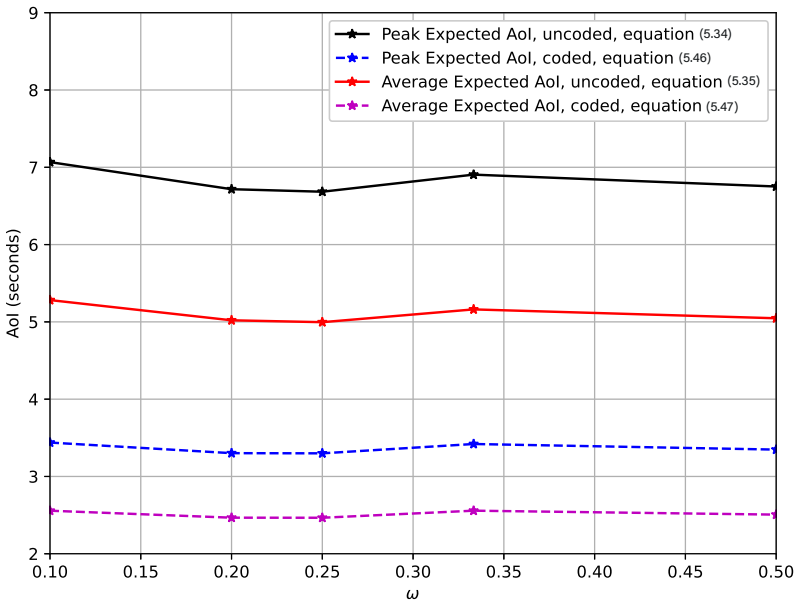


Figure 8.6. AoI measurements ($\omega \times h = 1$, $n = 30$, $k = 5$).

8.4.1. A Brief Note on the Relation of ω and h

ω denotes the probability that the sensor successfully harvests energy in a time slot and h denotes the amount of the harvested energy. Fig. 8.6 provides us with a general understanding on the interplay between ω and h. When $\omega \times h = 1$, ω as variable does not have a meaningful impact on the AoI values. This is an expected result as the efficiency of energy harvesting can be quantified by $\omega \times h$. As a result, we do not further consider the effect of h on the AoI values measured in IoT networks.

8.5. Conclusion

Given the significance of time-sensitive tasks and the adoption of AoI to measure data freshness, minimizing AoI in IoT networks holds paramount importance. Introducing network coding in IoT could potentially decrease AoI and enhance network efficiency by reducing latency and improving information freshness as shown in [227].

In this section, the effect of energy harvesting on network AoI has been considered when network coding is employed and when it is not. Via extensive simulations, the key observation is that energy harvesting magnifies the positive effect of network coding relative to the performance of uncoded transmissions, especially when the harvesting environment is inefficient. Therefore, in environments with energy harvesting sensors, network coding can be employed as a tool to increase data freshness in the network.

9. CONCLUSION

In this thesis, we have thoroughly investigated the concept of AoI in the context of wireless networks, particularly focusing on its application in the IoT. The initial chapter established the groundwork by introducing the fundamental principles of AoI and IoT, highlighting their crucial roles in modern networked environments. Subsequently, we conducted an extensive review of existing literature that utilizes AoI as a significant metric for timeliness across various network scenarios.

A prominent area of exploration involves optimizing dynamic user pairing in downlink NOMA systems. These algorithms aim to minimize data staleness while maximizing network efficiency by dynamically pairing users based on AoI metrics. Additionally, we proposed a novel scheduling strategy tailored for priority-aware IoT networks to enhance AoI optimization. This strategy prioritizes data packets based on their urgency, ensuring timely delivery of critical information. Furthermore, we investigated the impact of block coding on AoI in centralized IoT systems, drawing insights from BEC and BSC. We evaluated the efficacy of block coding techniques in reducing AoI through theoretical analysis and numerical experiments. Moreover, we explored the role of network coding in enhancing information freshness and reliability across diverse IoT networks, including its impact on AoI in multi-source multi-hop scenarios. In summary, this thesis provides a comprehensive examination of AoI optimization in wireless networks, spanning algorithmic design, network architecture, and coding strategies, thereby contributing to the advancement of next-generation wireless communication systems towards achieving timely and reliable information delivery.

Furthermore, there are intriguing avenues for future research. Analyzing and optimizing AoI in quantum communication networks emerge as a key direction. Considering the potential of quantum communications in future communication systems, a detailed analysis of AoI in quantum networks could pave the way for innovative and impactful contributions within today's technological landscape.

REFERENCES

1. "Ericsson Mobility Report", *https://www.ericsson.com/en/reports-and-papers/mobility-report*, accessed on May 4, 2023.

2. Kaul, S., M. Gruteser, V. Rai and J. Kenney, "Minimizing Age of Information in Vehicular Networks", *IEEE Communications Society Conference on Sensor, Mesh Ad Hoc Communications and Networks*, Salt Lake City, USA, pp. 350–358, 2011.

3. Saito, Y., Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li and K. Higuchi, "Non-Orthogonal Multiple Access for Cellular Future Radio Access", *IEEE Vehicular Technology Conference*, Dresden, Germany, pp. 1–5, 2013.

4. Mostafa, A. E., Y. Zhou and V. W. Wong, "Connectivity Maximization for Narrowband IoT Systems with NOMA", *IEEE International Conference on Communications*, Paris, France, pp. 1–6, 2017.

5. Liu, Y., G. Pan, H. Zhang and M. Song, "On the Capacity Comparison between MIMO-NOMA and MIMO-OMA", *IEEE Access*, Vol. 4, pp. 2123–2129, 2016.

6. Zeng, M., X. Li, G. Li, W. Hao and O. A. Dobre, "Sum Rate Maximization for IRS-Assisted Uplink NOMA", *IEEE Communications Letters*, Vol. 25, No. 1, pp. 234–238, 2021.

7. Köse, A., M. Koca, E. Anarim, M. Médard and H. Gökcesu, "Graph-Theoretical Dynamic User Pairing for Downlink NOMA Systems", *IEEE Communications Letters*, Vol. 25, No. 10, pp. 3234–3238, 2021.

8. Yang, X., H. Wang and Y. Feng, "Sum Rate Maximization for Active RIS-Aided Uplink Multi-Antenna NOMA Systems", *IEEE Wireless Communications Letters*, Vol. 12, No. 7, pp. 1149–1153, 2023.

9. Huang, H., Y. Yang, Z. Ding, H. Wang, H. Sari and F. Adachi, "Deep Learning-Based Sum Data Rate and Energy Efficiency Optimization for MIMO-NOMA Systems", *IEEE Transactions on Wireless Communications*, Vol. 19, No. 8, pp. 5373–5388, 2020.

10. Wang, N., F. Li, D. Chen, L. Liu and Z. Bao, "NOMA-Based Energy-Efficiency Optimization for UAV Enabled Space-Air-Ground Integrated Relay Networks", *IEEE Transactions on Vehicular Technology*, Vol. 71, No. 4, pp. 4129–4141, 2022.

11. Wang, S., Z. Fei, J. Guo, Q. Cui, S. Durrani and H. Yanikomeroglu, "Energy-Efficiency Optimization for Multiple Access in NOMA-Enabled Space–Air–Ground Networks", *IEEE Internet of Things Journal*, Vol. 10, No. 17, pp. 15652–15665, 2023.

12. Raeisidehkordi, E. and H. Bakhshi, "Joint Fairness and Power Optimization Under User Priority in NOMA Networks for IoT Applications", *IEEE Internet of Things Journal*, Vol. 9, No. 14, pp. 12440–12449, 2021.

13. Lin, Y., W. Feng, J. Wang, S. Jin and N. Ge, "User Fairness Optimization for Multi-UAV-Aided NOMA Networks: A Location-Aware Perspective", *IEEE Global Communications Conference*, Madrid, Spain, pp. 1–6, 2021.

14. Kaul, S., R. Yates and M. Gruteser, "Real-Time Status: How Often Should One Update?", *IEEE International Conference on Computer Communications*, Orlando, USA, pp. 2731–2735, 2012.

15. Yates, R. D., Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano and S. Ulukus, "Age of Information: An Introduction and Survey", *IEEE Journal on Selected Areas in Communications*, Vol. 39, No. 5, pp. 1183–1210, 2021.

16. Tang, H., J. Wang, L. Song and J. Song, "Minimizing Age of Information with Power Constraints: Multi-User Opportunistic Scheduling in Multi-State Time-

Varying Channels", *IEEE Journal on Selected Areas in Communications*, Vol. 38, No. 5, pp. 854–868, 2020.

17. Feng, S. and J. Yang, "Age of Information Minimization for an Energy Harvesting Source with Updating Erasures: Without and with Feedback", *IEEE Transactions on Communications*, Vol. 69, No. 8, pp. 5091–5105, 2021.

18. Wang, Q., H. Chen, Y. Gu, Y. Li and B. Vucetic, "Minimizing the Age of Information of Cognitive Radio-Based IoT Systems Under a Collision Constraint", *IEEE Transactions on Wireless Communications*, Vol. 19, No. 12, pp. 8054–8067, 2020.

19. Feng, X., S. Fu, F. Fang and F. R. Yu, "Optimizing Age of Information in RIS-Assisted NOMA Networks: A Deep Reinforcement Learning Approach", *IEEE Wireless Communications Letters*, Vol. 11, No. 10, pp. 2100–2104, 2022.

20. Wu, S., Z. Deng, A. Li, J. Jiao, N. Zhang and Q. Zhang, "Minimizing Age-of-Information in HARQ-CC Aided NOMA Systems", *IEEE Transactions on Wireless Communications*, Vol. 22, No. 2, pp. 1072–1086, 2023.

21. Yaghmaee, M. H., N. F. Bahalgardi and D. Adjeroh, "A Prioritization Based Congestion Control Protocol for Healthcare Monitoring Application in Wireless Sensor Networks", *Wireless Personal Communications*, Vol. 72, pp. 2605 – 2631, 2013.

22. Uysal, E., 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.

23. Say, S., H. Inata, J. Liu and S. Shimamoto, "Priority-Based Data Gathering Framework in UAV-Assisted Wireless Sensor Networks", *IEEE Sensors Journal*,

Vol. 16, No. 14, pp. 5785–5794, 2016.

24. Farag, H., E. Sisinni, M. Gidlund and P. Österberg, "Priority-Aware Wireless Fieldbus Protocol for Mixed-Criticality Industrial Wireless Sensor Networks", *IEEE Sensors Journal*, Vol. 19, No. 7, pp. 2767–2780, 2019.

25. Zhang, T., S. Chen, Z. Chen, Z. Tian, Y. Jia, M. Wang and D. O. Wu, "AoI and PAoI in the IoT-based Multi-Source Status Update System: Violation Probabilities and Optimal Arrival Rate Allocation", *IEEE Internet of Things Journal*, Vol. 10, No. 23, pp. 20617–20632, 2023.

26. Abdel-Aziz, M. K., S. Samarakoon, M. Bennis and W. Saad, "Ultra-Reliable and Low-Latency Vehicular Communication: An Active Learning Approach", *IEEE Communications Letters*, Vol. 24, No. 2, pp. 367–370, 2020.

27. Deng, L., Z. Shi, O. Li and J. Ji, "Joint Coding and Adaptive Image Transmission Scheme Based on DP-LDPC Codes for IoT Scenarios", *IEEE Access*, Vol. 7, pp. 18437–18449, 2019.

28. Tang, X., P. Reviriego, W. Tang, D. G. Mitchell, F. Lombardi and S. Liu, "Joint Learning and Channel Coding for Error-Tolerant IoT Systems Based on Machine Learning", *IEEE Transactions on Artificial Intelligence*, Vol. 5, No. 1, pp. 217–228, 2024.

29. Dommel, J., Z. Utkovski, O. Simeone and S. Stańczak, "Joint Source-Channel Coding for Semantics-Aware Grant-Free Radio Access in IoT Fog Networks", *IEEE Signal Processing Letters*, Vol. 28, pp. 728–732, 2021.

30. Fereydounian, M., H. Hassani, M. V. Jamali and H. Mahdavifar, "Channel Coding at Low Capacity", *IEEE Journal on Selected Areas in Information Theory*, Vol. 4, pp. 351–362, 2023.

31. Li, D., S. Wu, J. Jiao, N. Zhang and Q. Zhang, "Towards Age-Optimal Transmis-

sion in Satellite-Integrated IoT: A Two-Layer Coding Approach", *IEEE Transactions on Vehicular Technology*, Vol. 72, No. 1, pp. 1137–1148, 2022.

32. You, J., S. Wu, Y. Deng, J. Jiao and Q. Zhang, "An Age Optimized Hybrid ARQ Scheme for Polar Codes via Gaussian Approximation", *IEEE Wireless Communications Letters*, Vol. 10, No. 10, pp. 2235–2239, 2021.

33. Ahlswede, R., N. Cai, S.-Y. Li and R. W. Yeung, "Network Information Flow", *IEEE Transactions on Information Theory*, Vol. 46, No. 4, pp. 1204–1216, 2000.

34. Fragouli, C., E. Soljanin *et al.*, "Network Coding Fundamentals", *Foundations and Trends in Networking*, Vol. 2, No. 1, pp. 1–133, 2007.

35. Li, S.-Y., R. W. Yeung and N. Cai, "Linear Network Coding", *IEEE Transactions on Information Theory*, Vol. 49, No. 2, pp. 371–381, 2003.

36. Koetter, R. and M. Médard, "An Algebraic Approach to Network Coding", *IEEE/ACM Transactions on Networking*, Vol. 11, No. 5, pp. 782–795, 2003.

37. Katti, S., H. Rahul, W. Hu, D. Katabi, M. Médard and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding", *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Pisa, Italy, pp. 243–254, 2006.

38. Chen, X. and S. S. Bidokhti, "Benefits of Coding on Age of Information in Broadcast Networks", *IEEE Information Theory Workshop*, Visby, Sweden, pp. 1–5, 2019.

39. Costa, M. and Y. E. Sagduyu, "Age of Information with Network Coding", *Ad Hoc Networks*, Vol. 86, pp. 15–22, 2019.

40. Costa, M., Y. E. Sagduyu, T. Erpek and M. Médard, "Robust Improvement of the Age of Information by Adaptive Packet Coding", *IEEE International Conference*

on Communications, Montreal, Canada, pp. 1–6, 2021.

41. Pan, H., T.-T. Chan, V. C. Leung and J. Li, "Age of Information in Physical-Layer Network Coding Enabled Two-Way Relay Networks", *IEEE Transactions on Mobile Computing*, Vol. 22, No. 8, pp. 4485–4499, 2022.

42. Talak, R., S. Karaman and E. Modiano, "Minimizing Age-of-Information in Multi-Hop Wireless Networks", *Allerton Conference on Communication, Control, and Computing*, Monticello, USA, pp. 486–493, 2017.

43. Bedewy, A. M., Y. Sun and N. B. Shroff, "The Age of Information in Multihop Networks", *IEEE/ACM Transactions on Networking*, Vol. 27, No. 3, pp. 1248–1257, 2019.

44. Li, D., S. Wu, Y. Wang, J. Jiao and Q. Zhang, "Age-Optimal HARQ Design for Freshness-Critical Satellite-IoT Systems", *IEEE Internet of Things Journal*, Vol. 7, No. 3, pp. 2066–2076, 2019.

45. Liu, S., J. Jiao, Z. Ni, S. Wu and Q. Zhang, "Age-Optimal NC-HARQ Protocol for Multi-Hop Satellite-Based Internet of Things", *IEEE Wireless Communications and Networking Conference*, Nanjing, China, pp. 1–6, 2021.

46. Xie, M., Q. Wang, J. Gong and X. Ma, "Age and Energy Analysis for LDPC Coded Status Update with and without ARQ", *IEEE Internet of Things Journal*, Vol. 7, No. 10, pp. 10388–10400, 2020.

47. Pawar, A. B. and S. Ghumbre, "A Survey on IoT Applications, Security Challenges and Counter Measures", *International Conference on Computing, Analytics and Security Trends*, Pune, India, pp. 294–299, 2016.

48. Akar, N. and O. Dogan, "Discrete-Time Queueing Model of Age of Information With Multiple Information Sources", *IEEE Internet of Things Journal*, Vol. 8, No. 19, pp. 14531–14542, 2021.

49. Pan, W., Z. Deng, X. Wang, P. Zhou and W. Wu, "Optimizing the Age of Information for Multi-Source Information Update in Internet of Things", *IEEE Transactions on Network Science and Engineering*, Vol. 9, No. 2, pp. 904–917, 2022.

50. Ko, H., H. Lee, T. Kim and S. Pack, "Information Freshness-Guaranteed and Energy-Efficient Data Generation Control System in Energy Harvesting Internet of Things", *IEEE Access*, Vol. 8, pp. 168711–168720, 2020.

51. Ling, X., J. Gong, R. Li, S. Yu, Q. Ma and X. Chen, "Dynamic Age Minimization with Real-Time Information Preprocessing for Edge-Assisted IoT Devices with Energy Harvesting", *IEEE Transactions on Network Science and Engineering*, Vol. 8, No. 3, pp. 2288–2300, 2021.

52. Dong, Y., P. Fan and K. B. Letaief, "Energy Harvesting Powered Sensing in IoT: Timeliness versus Distortion", *IEEE Internet of Things Journal*, Vol. 7, No. 11, pp. 10897–10911, 2020.

53. Hatami, M., M. Leinonen and M. Codreanu, "AoI Minimization in Status Update Control with Energy Harvesting Sensors", *IEEE Transactions on Communications*, Vol. 69, No. 12, pp. 8335–8351, 2021.

54. Pappas, N., Z. Chen and M. Hatami, "Average AoI of Cached Status Updates for a Process Monitored by an Energy Harvesting Sensor", *Conference on Information Sciences and Systems*, Princeton, USA, pp. 1–5, 2020.

55. Hatami, M., M. Leinonen and M. Codreanu, "Minimizing Average On-Demand AoI in an IoT Network with Energy Harvesting Sensors", *IEEE International Workshop on Signal Processing Advances in Wireless Communications*, Lucca, Italy, pp. 1–5, 2021.

56. Centenaro, M., C. E. Costa, F. Granelli, C. Sacchi and L. Vangelista, "A Sur-

vey on Technologies, Standards and Open Challenges in Satellite IoT", *IEEE Communications Surveys Tutorials*, Vol. 23, No. 3, pp. 1693–1720, 2021.

57. Kharroub, S. K., K. Abualsaud and M. Guizani, "Medical IoT: A Comprehensive Survey of Different Encryption and Security Techniques", *International Wireless Communications and Mobile Computing*, Limassol, Cyprus, pp. 1891–1896, 2020.

58. Vijayalakshmi, S. R. and S. Muruganand, "A Survey of Internet of Things in Fire Detection and Fire Industries", *International Conference on IoT in Social, Mobile, Analytics and Cloud*, Palladam, India, pp. 703–707, 2017.

59. Eshaghi, M. and R. Rashidzadeh, "An Energy Harvesting Solution for IoT Devices in 5G Networks", *IEEE Canadian Conference on Electrical and Computer Engineering*, London, ON, Canada, pp. 1–4, 2020.

60. Yu, B., X. Chen and Y. Cai, "Age of Information for Cellular Internet of Things: Challenges, Key Techniques and Future Trends", *IEEE Communications Magazine*, Vol. 60, No. 12, pp. 20–26, 2022.

61. Abbas, Q., S. A. Hassan, H. K. Qureshi, K. Dev and H. Jung, "A Comprehensive Survey on Age of Information in Massive IoT Networks", *Computer Communications*, Vol. 197, pp. 199–213, 2023.

62. Al-Fuqaha, A., M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys Tutorials*, Vol. 17, No. 4, pp. 2347–2376, 2015.

63. Korala, H., P. P. Jayaraman, A. Yavari and D. Georgakopoulos, "APOLLO: A Platform for Experimental Analysis of Time Sensitive Multimedia IoT Applications", *International Conference on Advances in Mobile Computing and Multimedia*, Fukuoka, Japan, pp. 104–113, 2020.

64. Korala, H., D. Georgakopoulos, P. P. Jayaraman and A. Yavari, "Managing Time-

Sensitive IoT Applications via Dynamic Application Task Distribution and Adaptation", *Remote Sensing*, Vol. 13, No. 20, p. 4148, 2021.

65. Fedullo, T., A. Morato, F. Tramarin, L. Rovati and S. Vitturi, "A Comprehensive Review on Time Sensitive Networks with a Special Focus on Its Applicability to Industrial Smart and Distributed Measurement Systems", *Sensors*, Vol. 22, No. 4, p. 1638, 2022.

66. García-Valls, M. and E. Palomar-Cosín, "An Evaluation Process for IoT Platforms in Time-Sensitive Domains", *Sensors*, Vol. 22, No. 23, p. 9501, 2022.

67. Shumba, A.-T., T. Montanaro, I. Sergi, L. Fachechi, M. De Vittorio and L. Patrono, "Leveraging IoT-Aware Technologies and AI Techniques for Real-Time Critical Healthcare Applications", *Sensors*, Vol. 22, No. 19, p. 7675, 2022.

68. Perez-Ramirez, J., O. Seijo and I. Val, "Time-Critical IoT Applications Enabled by Wi-Fi 6 and Beyond", *IEEE Internet of Things Magazine*, Vol. 5, No. 3, pp. 44–49, 2022.

69. Alriksson, F., L. Boström, J. Sachs, Y.-P. E. Wang and A. Zaidi, "Critical IoT Connectivity Ideal for Time-Critical Communications", *Ericsson Technology Review*, Vol. 2020, No. 6, pp. 2–13, 2020.

70. Huang, K., W. Liu, Y. Li, A. Savkin and B. Vucetic, "Wireless Feedback Control with Variable Packet Length for Industrial IoT", *IEEE Wireless Communications Letters*, Vol. 9, No. 9, pp. 1586–1590, 2020.

71. Al-Gumaei, K., K. Schuba, A. Friesen, S. Heymann, C. Pieper, F. Pethig and S. Schriegel, "A Survey of Internet of Things and Big Data Integrated Solutions for Industrie 4.0", *IEEE International Conference on Emerging Technologies and Factory Automation*, Vol. 1, Turin, Italy, pp. 1417–1424, 2018.

72. Uyoata, U., J. Mwangama and R. Adeogun, "Relaying in the Internet of Things:

A Survey", *IEEE Access*, Vol. 9, pp. 132675–132704, 2021.

73. Zheng, D., Y. Yang, L. Wei and B. Jiao, "Decode-and-Forward Short-Packet Relaying in the Internet of Things: Timely Status Updates", *IEEE Transactions on Wireless Communications*, Vol. 20, No. 12, pp. 8423–8437, 2021.

74. Simiscuka, A. A. and G.-M. Muntean, "Age of Information as a QoS Metric in a Relay-Based IoT Mobility Solution", *International Wireless Communications and Mobile Computing Conference*, Limassol, Cyprus, pp. 868–873, 2018.

75. Zhou, C., H. He, P. Yang, F. Lyu, W. Wu, N. Cheng and X. Shen, "Deep RL-Based Trajectory Planning for AoI Minimization in UAV-Assisted IoT", *International Conference on Wireless Communications and Signal Processing*, Xi'an, China, pp. 1–6, 2019.

76. Shah, K. and Z. Narmavala, "A Survey on Green Internet of Things", *International Conference on Information Processing*, Bangalore, India, pp. 1–4, 2018.

77. Zhou, B. and W. Saad, "Age of Information in Ultra-Dense Computation-Intensive Internet of Things Systems", *International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks*, Philadelphia, USA, pp. 1–8, 2021.

78. López-Pérez, D., M. Ding, H. Claussen and A. H. Jafari, "Towards 1 Gbps/UE in Cellular Systems: Understanding Ultra-Dense Small Cell Deployments", *IEEE Communications Surveys Tutorials*, Vol. 17, No. 4, pp. 2078–2101, 2015.

79. Hejazi, H., H. Rajab, T. Cinkler and L. Lengyel, "Survey of Platforms for Massive IoT", *IEEE International Conference on Future IoT Technologies*, Eger, Hungary, pp. 1–8, 2018.

80. Zhang, H., Y. Kang, Z. Han and H. V. Poor, "AoI Minimization for Grant-Free Massive Access with Short Packets using Mean-Field Games", *IEEE Global*

Communications Conference, Taipei, Taiwan, pp. 1–6, 2020.

81. Khan, A. A., M. H. Rehmani and A. Rachedi, "Cognitive-Radio-Based Internet of Things: Applications, Architectures, Spectrum Related Functionalities, and Future Research Directions", *IEEE Wireless Communications*, Vol. 24, No. 3, pp. 17–25, 2017.

82. Gu, Y., H. Chen, C. Zhai, Y. Li and B. Vucetic, "Minimizing Age of Information in Cognitive Radio-Based IoT Systems: Underlay or Overlay?", *IEEE Internet of Things Journal*, Vol. 6, No. 6, pp. 10273–10288, 2019.

83. Ceran, E. T., D. Gündüz and A. György, "Average Age of Information with Hybrid ARQ Under a Resource Constraint", *IEEE Transactions on Wireless Communications*, Vol. 18, No. 3, pp. 1900–1913, 2019.

84. Krikidis, I., "Average Age of Information in Wireless Powered Sensor Networks", *IEEE Wireless Communications Letters*, Vol. 8, No. 2, pp. 628–631, 2019.

85. Seo, J.-B. and J. Choi, "On the Outage Probability of Peak Age-of-Information for D/G/1 Queuing Systems", *IEEE Communications Letters*, Vol. 23, No. 6, pp. 1021–1024, 2019.

86. Mankar, P. D., Z. Chen, M. A. Abd-Elmagid, N. Pappas and H. S. Dhillon, "A Spatio-Temporal Analysis of Cellular-Based IoT Networks under Heterogeneous Traffic", *IEEE Global Communications Conference*, Madrid, Spain, pp. 1–6, 2021.

87. Champati, J. P., R. R. Avula, T. J. Oechtering and J. Gross, "Minimum Achievable Peak Age of Information Under Service Preemptions and Request Delay", *IEEE Journal on Selected Areas in Communications*, Vol. 39, No. 5, pp. 1365–1379, 2021.

88. Kosta, A., N. Pappas, V. Angelakis *et al.*, "Age of Information: A New Concept, Metric, and Tool", *Foundations and Trends in Networking*, Vol. 12, No. 3, pp.

162–259, 2017.

89. Alliance, N., "5G White Paper", *Next Generation Mobile Networks*, Vol. 1, pp. 1–125, 2015.

90. Shreedhar, T., S. K. Kaul and R. D. Yates, "An Age Control Transport Protocol for Delivering Fresh Updates in the Internet-of-Things", *International Symposium on World of Wireless Mobile and Multimedia Networks*, Washington, USA, pp. 1–7, 2019.

91. Mankar, P. D., Z. Chen, M. A. Abd-Elmagid, N. Pappas and H. S. Dhillon, "Throughput and Age of Information in a Cellular-Based IoT Network", *IEEE Transactions on Wireless Communications*, Vol. 20, No. 12, pp. 8248–8263, 2021.

92. Bae, Y. H. and J. W. Baek, "Age of Information and Throughput in Random Access-Based IoT Systems with Periodic Updating", *IEEE Wireless Communications Letters*, Vol. 11, No. 4, 821–825, 2022.

93. Abbas, Q., S. A. Hassan, H. Pervaiz and Q. Ni, "A Markovian Model for the Analysis of Age of Information in IoT Networks", *IEEE Wireless Communications Letters*, Vol. 10, No. 7, pp. 1596–1600, 2021.

94. Zhou, B. and W. Saad, "Performance Analysis of Age of Information in Ultra-Dense Internet of Things Systems with Noisy Channels", *IEEE Transactions on Wireless Communications*, Vol. 21, No. 5, 3493–3507, 2022.

95. Han, R., J. Wang, L. Bai, J. Liu and J. Choi, "Age of Information and Performance Analysis for UAV-Aided IoT Systems", *IEEE Internet of Things Journal*, Vol. 8, No. 19, pp. 14447–14457, 2021.

96. Hu, L., Z. Chen, Y. Jia, M. Wang and T. Q. Quek, "Asymptotically Optimal Arrival Rate for IoT Networks With AoI and Peak AoI Constraints", *IEEE Communications Letters*, Vol. 25, No. 12, pp. 3853–3857, 2021.

97. Pappalardo, M., E. Mingozzi and A. Virdis, "A Model-Driven Approach to AoI-Based Cache Management in IoT", *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*, Porto, Portugal, pp. 1–6, 2021.

98. Fountoulakis, E., N. Pappas, M. Codreanu and A. Ephremides, "Optimal Sampling Cost in Wireless Networks with Age of Information Constraints", *IEEE International Conference on Computer Communications*, Toronto, Canada, pp. 918–923, 2020.

99. Abbas, Q., S. Zeb, S. A. Hassan, R. Mumtaz and S. A. R. Zaidi, "Joint Optimization of Age of Information and Energy Efficiency in IoT Networks", *IEEE Vehicular Technology Conference*, Antwerp, Belgium, pp. 1–5, 2020.

100. Hsu, Y.-L., C.-F. Liu, S. Samarakoon, H.-Y. Wei and M. Bennis, "Age-Optimal Power Allocation in Industrial IoT: A Risk-Sensitive Federated Learning Approach", *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Helsinki, Finland, pp. 1323–1328, 2021.

101. Hu, L., Z. Chen, Y. Dong, Y. Jia, L. Liang and M. Wang, "Status Update in IoT Networks: Age-of-Information Violation Probability and Optimal Update Rate", *IEEE Internet of Things Journal*, Vol. 8, No. 14, pp. 11329–11344, 2021.

102. Chen, H., Y. Gu and S.-C. Liew, "Age-of-Information Dependent Random Access for Massive IoT Networks", *IEEE International Conference on Computer Communications*, Toronto, Canada, pp. 930–935, 2020.

103. Grybosi, J. F., J. L. Rebelatto and G. L. Moritz, "Age of Information of SIC-Aided Massive IoT Networks with Random Access", *IEEE Internet of Things Journal*, Vol. 9, No. 1, pp. 662–670, 2021.

104. Wang, X., C. Chen, J. He, S. Zhu and X. Guan, "AoI-Aware Control and Commu-

nication Co-Design for Industrial IoT Systems", *IEEE Internet of Things Journal*, Vol. 8, No. 10, pp. 8464–8473, 2020.

105. Fountoulakis, E., M. Codreanu, A. Ephremides and N. Pappas, "Joint Sampling and Transmission Policies for Minimizing Cost under AoI Constraints", arXiv 2103.15450, 2021.

106. Lyu, L., Y. Dai, N. Cheng, S. Zhu, X. Guan, B. Lin and X. Shen, "AoI-Aware Co-Design of Cooperative Transmission and State Estimation for Marine IoT Systems", *IEEE Internet of Things Journal*, Vol. 8, No. 10, pp. 7889–7901, 2020.

107. Zheng, H., K. Xiong, P. Fan, Z. Zhong and K. B. Letaief, "Age of Information-Based Wireless Powered Communication Networks with Selfish Charging Nodes", *IEEE Journal on Selected Areas in Communications*, Vol. 39, No. 5, pp. 1393–1411, 2021.

108. Gu, Y., H. Chen, Y. Zhou, Y. Li and B. Vucetic, "Timely Status Update in Internet of Things Monitoring Systems: An Age-Energy Tradeoff", *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 5324–5335, 2019.

109. Liao, S., J. Jiao, S. Wu, R. Lu and Q. Zhang, "Age-Optimal Power Allocation Scheme for NOMA-Based S-IoT Downlink Network", *IEEE International Conference on Communications*, Montreal, Canada, pp. 1–6, 2021.

110. Chi, K., M. Wu, Y. Chen and C. Xu, "AoI Optimal Dynamic Power Control for IoT Networks: A DRDPG Approach", *IEEE International Conference on Electronic Information and Communication Technology*, Xi'an, China, pp. 48–53, 2021.

111. Chen, M., K. Wu and L. Song, "A Whittle Index Approach to Minimizing Age of Multi-Packet Information in IoT Network", *IEEE Access*, Vol. 9, pp. 31467–31480, 2021.

112. Pan, W., Z. Deng, X. Wang, P. Zhou and W. Wu, "Optimizing the Age of Information for Multi-Source Information Update in Internet of Things", *IEEE Transactions on Network Science and Engineering*, Vol. 9, No. 2, pp. 904–917, 2022.

113. Zhou, B. and W. Saad, "On the Age of Information in Internet of Things Systems with Correlated Devices", *IEEE Global Communications Conference*, Taipei, Taiwan, pp. 1–6, 2020.

114. Hribar, J. and L. DaSilva, "Utilising Correlated Information to Improve the Sustainability of Internet of Things Devices", *IEEE World Forum on Internet of Things*, Limerick, Ireland, pp. 805–808, 2019.

115. Liu, C.-F. and M. Bennis, "Federated Learning with Correlated Data: Taming the Tail for Age-Optimal Industrial IoT", *International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks*, Philadelphia, USA, pp. 1–6, 2021.

116. Hao, X., T. Yang, Y. Hu and B. Hu, "Performance Analysis for Correlated AoI and Energy Efficiency in Heterogeneous CR-IoT System", *IEEE International Conference on Computer Communications*, Vancouver, Canada, pp. 1–8, 2021.

117. Xu, C., H. H. Yang, X. Wang and T. Q. Quek, "Optimizing Information Freshness in Computing-Enabled IoT Networks", *IEEE Internet of Things Journal*, Vol. 7, No. 2, pp. 971–985, 2019.

118. Lou, J., X. Yuan and N.-F. Tzeng, "Instant AoI Optimization in IoT Networks with Packet Combination", *IEEE International Conference on Sensing, Communication, and Networking*, Como, Italy, pp. 1–9, 2020.

119. Wang, X., M. Fang, C. Xu, H. H. Yang, X. Sun, X. Chen and T. Q. Quek, "When to Preprocess? Keeping Information Fresh for Computing Enable Internet of

Things", *IEEE Internet of Things Journal*, Vol. 9, No. 6, pp. 4303–4317, 2022.

120. Fang, M., X. Wang, C. Xu, H. H. Yang and T. Q. Quek, "Computing-Aided Update for Information Freshness in the Internet of Things", *IEEE International Conference on Computer Communications*, Vancouver, Canada, pp. 1–7, 2021.

121. Yazdani, N. and D. E. Lucani, "Online Compression of Multiple IoT Sources Reduces the Age of Information", *IEEE Internet of Things Journal*, Vol. 8, No. 19, pp. 14514–14530, 2021.

122. Xu, C., X. Wang, H. H. Yang, H. Sun and T. Q. Quek, "AoI and Energy Consumption Oriented Dynamic Status Updating in Caching Enabled IoT Networks", *IEEE International Conference on Computer Communications*, Toronto, Canada, pp. 710–715, 2020.

123. Xie, X., H. Wang and M. Weng, "A Reinforcement Learning Approach for Optimizing the Age of Computing Enabled IoT", *IEEE Internet of Things Journal*, Vol. 9, No. 4, pp. 2778–2786, 2022.

124. Akbari, M., M. R. Abedi, R. Joda, M. Pourghasemian, N. Mokari and M. Erol-Kantarci, "Age of Information Aware VNF Scheduling in Industrial IoT Using Deep Reinforcement Learning", *IEEE Journal on Selected Areas in Communications*, Vol. 39, No. 8, pp. 2487–2500, 2021.

125. Wang, S., M. Chen, Z. Yang, C. Yin, W. Saad, S. R. Cui and H. V. Poor, "Distributed Reinforcement Learning for Age of Information Minimization in Real-Time IoT Systems", *IEEE Journal of Selected Topics in Signal Processing*, Vol. 16, No. 3, pp. 501–515, 2022.

126. Hatami, M., M. Jahandideh, M. Leinonen and M. Codreanu, "Age-Aware Status Update Control for Energy Harvesting IoT Sensors via Reinforcement Learning", *IEEE International Symposium on Personal, Indoor and Mobile Radio Commu-*

nications, London, UK, pp. 1–6, 2020.

127. Xu, C., Y. Xie, X. Wang, H. H. Yang, D. Niyato and T. Q. Quek, "Optimal Status Update for Caching Enabled IoT Networks: A Dueling Deep R-Network Approach", *IEEE Transactions on Wireless Communications*, Vol. 20, No. 12, pp. 8438–8454, 2021.

128. Xie, X., H. Wang, L. Yu and M. Weng, "Online Algorithms for Optimizing Age of Information in the IoT Systems with Multi-Slot Status Delivery", *IEEE Wireless Communications Letters*, Vol. 10, No. 5, pp. 971–975, 2021.

129. Stamatakis, G., N. Pappas, A. Fragkiadakis and A. Traganitis, "Autonomous Maintenance in IoT Networks via AoI-Driven Deep Reinforcement Learning", *IEEE International Conference on Computer Communications*, Vancouver, Canada, pp. 1–7, 2021.

130. Yang, Y., X. Wei, R. Xu, L. Peng and L. Liu, "Game-Based Channel Access for AoI-Oriented Data Transmission Under Dynamic Attack", *IEEE Internet of Things Journal*, Vol. 9, No. 11, pp. 8820–8837, 2022.

131. Abd-Elmagid, M. A., A. Ferdowsi, H. S. Dhillon and W. Saad, "Deep Reinforcement Learning for Minimizing Age-of-Information in UAV-Assisted Networks", *IEEE Global Communications Conference*, Waikoloa, USA, pp. 1–6, 2019.

132. Tong, P., J. Liu, X. Wang, B. Bai and H. Dai, "Deep Reinforcement Learning for Efficient Data Collection in UAV-Aided Internet of Things", *IEEE International Conference on Communications*, Dublin, Ireland, pp. 1–6, 2020.

133. Jia, Z., X. Qin, Z. Wang and B. Liu, "Age-Based Path Planning and Data Acquisition in UAV-Assisted IoT Networks", *IEEE International Conference on Communications*, Shanghai, China, pp. 1–6, 2019.

134. Hu, H., K. Xiong, G. Qu, Q. Ni, P. Fan and K. B. Letaief, "AoI-Minimal Tra-

jectory Planning and Data Collection in UAV-Assisted Wireless Powered IoT Networks", *IEEE Internet of Things Journal*, Vol. 8, No. 2, pp. 1211–1223, 2020.

135. Sun, M., X. Xu, X. Qin and P. Zhang, "AoI-Energy-Aware UAV-Assisted Data Collection for IoT Networks: A Deep Reinforcement Learning Method", *IEEE Internet of Things Journal*, Vol. 8, No. 24, pp. 17275–17289, 2021.

136. Gong, Z., Q. Cui, C. Chaccour, B. Zhou, M. Chen and W. Saad, "Lifelong Learning for Minimizing Age of Information in Internet of Things Networks", *IEEE International Conference on Communications*, Montreal, Canada, pp. 1–6, 2021.

137. Fang, Z., J. Wang, C. Jiang, Q. Zhang and Y. Ren, "AoI-Inspired Collaborative Information Collection for AUV-Assisted Internet of Underwater Things", *IEEE Internet of Things Journal*, Vol. 8, No. 19, pp. 14559–14571, 2021.

138. Jiang, Z., B. Krishnamachari, S. Zhou and Z. Niu, "Can Decentralized Status Update Achieve Universally Near-Optimal Age-of-Information in Wireless Multiaccess Channels?", *International Teletraffic Congress*, Vol. 1, pp. 144–152, 2018.

139. Park, T., W. Saad and B. Zhou, "Centralized and Distributed Age of Information Minimization with Nonlinear Aging Functions in the Internet of Things", *IEEE Internet of Things Journal*, Vol. 8, No. 10, pp. 8437–8455, 2020.

140. Yu, D., Y. Zou, M. Xu, Y. Xu, Y. Zhang, B. Gong and X. Xing, "Competitive Age of Information in Dynamic IoT Networks", *IEEE Internet of Things Journal*, Vol. 8, No. 20, pp. 15160–15169, 2020.

141. Stamatakis, G., N. Pappas and A. Traganitis, "Optimal Policies for Status Update Generation in an IoT Device with Heterogeneous Traffic", *IEEE Internet of Things Journal*, Vol. 7, No. 6, pp. 5315–5328, 2020.

142. Samir, M., C. Assi, S. Sharafeddine and A. Ghrayeb, "Online Altitude Control and Scheduling Policy for Minimizing AoI in UAV-Assisted IoT Wireless Networks",

IEEE Transactions on Mobile Computing, Vol. 21, No. 7, pp. 2493–2505, 2020.

143. Choudhury, B., V. K. Shah, A. Ferdowsi, J. H. Reed and Y. T. Hou, "AoI-Minimizing Scheduling in UAV-Relayed IoT Networks", *IEEE International Conference on Mobile Ad Hoc and Smart Systems*, Denver, USA, pp. 117–126, 2021.

144. Zhou, B. and W. Saad, "Minimum Age of Information in the Internet of Things with Non-Uniform Status Packet Sizes", *IEEE Transactions on Wireless Communications*, Vol. 19, No. 3, pp. 1933–1947, 2019.

145. Corneo, L., C. Rohner and P. Gunningberg, "Age of Information-Aware Scheduling for Timely and Scalable Internet of Things Applications", *IEEE International Conference on Computer Communications*, Paris, France, pp. 2476–2484, 2019.

146. Li, C., S. Li, Y. Chen, Y. T. Hou and W. Lou, "Minimizing Age of Information Under General Models for IoT Data Collection", *IEEE Transactions on Network Science and Engineering*, Vol. 7, No. 4, pp. 2256–2270, 2019.

147. Liu, L., H. H. Yang, C. Xu and F. Jiang, "On the Peak Age of Information in NOMA IoT Networks with Stochastic Arrivals", *IEEE Wireless Communications Letters*, Vol. 10, No. 12, pp. 2757–2761, 2021.

148. Li, J., Y. Zhou and H. Chen, "Age of Information for Multicast Transmission with Fixed and Random Deadlines in IoT Systems", *IEEE Internet of Things Journal*, Vol. 7, No. 9, pp. 8178–8191, 2020.

149. Wang, J., X. Cao, B. Yin and Y. Cheng, "Sleep-Wake Sensor Scheduling for Minimizing AoI-Penalty in Industrial Internet of Things", *IEEE Internet of Things Journal*, Vol. 9, No. 9, pp. 6404–6417, 2022.

150. Wang, D., H. Qin, Z. Li, P. Qi, N. Zhang, X. Chen and K. Yu, "Minimizing AoI Under Covertness Constraints in Internet of Things Networks", *IEEE International Conference on Frontiers Technology of Information and Computer*,

Greenville, USA, pp. 694–700, 2021.

151. Zhou, B. and W. Saad, "Joint Status Sampling and Updating for Minimizing Age of Information in the Internet of Things", *IEEE Transactions on Communications*, Vol. 67, No. 11, pp. 7468–7482, 2019.

152. Abd-Elmagid, M. A., H. S. Dhillon and N. Pappas, "Online Age-Minimal Sampling Policy for RF-Powered IoT Networks", *IEEE Global Communications Conference*, Waikoloa, USA, pp. 1–6, 2019.

153. Abd-Elmagid, M. A., H. S. Dhillon and N. Pappas, "AoI-Optimal Joint Sampling and Updating for Wireless Powered Communication Systems", *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 11, pp. 14110–14115, 2020.

154. Rovira-Sugranes, A. and A. Razi, "Optimizing the Age of Information for Blockchain Technology with Applications to IoT Sensors", *IEEE Communications Letters*, Vol. 24, No. 1, pp. 183–187, 2019.

155. Abd-Elmagid, M. A., N. Pappas and H. S. Dhillon, "On the Role of Age of Information in the Internet of Things", *IEEE Communications Magazine*, Vol. 57, No. 12, pp. 72–77, 2019.

156. Wang, Q., H. Chen, Y. Li, Z. Pang and B. Vucetic, "Minimizing Age of Information for Real-Time Monitoring in Resource-Constrained Industrial IoT Networks", *IEEE International Conference on Industrial Informatics*, Vol. 1, Toyama, Japan, pp. 1766–1771, 2019.

157. Jiang, Z., B. Krishnamachari, X. Zheng, S. Zhou and Z. Niu, "Timely Status Update in Wireless Uplinks: Analytical Solutions with Asymptotic Optimality", *IEEE Internet of Things Journal*, Vol. 6, No. 2, pp. 3885–3898, 2019.

158. Fizza, K., P. P. Jayaraman, A. Banerjee, D. Georgakopoulos and R. Ranjan, "Age of Data Aware Internet of Things Applications", *IEEE Consumer Communica-*

tions and Networking Conference, Las Vegas, USA, pp. 399–404, 2022.

159. Beytur, H. B., S. Baghaee and E. Uysal, "Towards AoI-Aware Smart IoT Systems", *International Conference on Computing, Networking and Communications*, Las Vegas, USA, pp. 353–357, 2020.

160. Lee, S., M. Kim, J. Lee, R.-H. Hsu and T. Q. Quek, "Is Blockchain Suitable for Data Freshness? An Age-of-Information Perspective", *IEEE Network*, Vol. 35, No. 2, pp. 96–103, 2021.

161. Yazdani, N. and D. E. Lucani, "Age of Information Analysis for Instantly Decompressible IoT Protocols", *IEEE International Conference on Communications*, Dublin, Ireland, pp. 1–6, 2020.

162. Dong, Y., Z. Chen, S. Liu, P. Fan and K. B. Letaief, "Age-upon-Decisions Minimizing Scheduling in Internet of Things: To be Random or to be Deterministic?", *IEEE Internet of Things Journal*, Vol. 7, No. 2, pp. 1081–1097, 2019.

163. Bao, Z., Y. Dong, Z. Chen, P. Fan and K. B. Letaief, "Age-Optimal Service and Decision Processes in Internet of Things", *IEEE Internet of Things Journal*, Vol. 8, No. 4, pp. 2826–2841, 2020.

164. Wakisaka, S., Y.-H. Chiang, H. Lin and Y. Ji, "Timely Information Updates for the Internet of Things with Serverless Computing", *IEEE International Conference on Communications*, Montreal, Canada, pp. 1–6, 2021.

165. Zhou, B., W. Saad, M. Bennis and P. Popovski, "Risk-Aware Optimization of Age of Information in the Internet of Things", *IEEE International Conference on Communications*, Dublin, Ireland, pp. 1–6, 2020.

166. Wang, X., W. Lin, C. Xu, X. Sun and X. Chen, "Age of Changed Information: Content-Aware Status Updating in the Internet of Things", *IEEE Transactions on Communications*, Vol. 70, No. 1, 578–591, 2021.

167. Li, R., Q. Ma, J. Gong, Z. Zhou and X. Chen, "Age of Processing: Age-Driven Status Sampling and Processing Offloading for Edge Computing-Enabled Real-Time IoT Applications", *IEEE Internet of Things Journal*, Vol. 8, No. 19, pp. 14471–14484, 2021.

168. Li, Y., Y. Xu, Q. Zhang and Z. Yang, "Age-Driven Spatially-Temporally Correlative Updating in the Satellite-Integrated Internet of Things via Markov Decision Process", *IEEE Internet of Things Journal*, Vol. 9, No. 15, pp. 13612–13625, 2022.

169. Chiang, Y.-H., H. Lin and Y. Ji, "Information Cofreshness-Aware Grant Assignment and Transmission Scheduling for Internet of Things", *IEEE Internet of Things Journal*, Vol. 8, No. 19, pp. 14435–14446, 2021.

170. Ding, Z., R. Schober and H. V. Poor, "A General MIMO Framework for NOMA Downlink and Uplink Transmission Based on Signal Alignment", *IEEE Transactions on Wireless Communications*, Vol. 15, No. 6, pp. 4438–4454, 2016.

171. Ding, Z., P. Fan and H. V. Poor, "Impact of User Pairing on 5G Nonorthogonal Multiple-Access Downlink Transmissions", *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 8, pp. 6010–6023, 2016.

172. Deraz, S. A., M. R. Rizk, S. Shaaban and K. H. Moussa, "Performance Analysis of MIMO-NOMA Systems Based on Dynamic User Pairing Scheme", Vol. 1447, No. 1, p. 012016, 2020.

173. Faenza, Y., G. Oriolo and G. Stauffer, "Solving the Weighted Stable Set Problem in Claw-Free Graphs via Decomposition", *Journal of the ACM*, Vol. 61, No. 4, pp. 1–41, 2014.

174. Köse, A. and M. Médard, "Scheduling Wireless Ad Hoc Networks in Polynomial Time Using Claw-Free Conflict Graphs", *IEEE International Symposium on Per-*

sonal, Indoor and Mobile Radio Communications, Montreal, Quebec, Canada, pp. 1–7, 2017.

175. Farazi, S., A. G. Klein and D. R. Brown, "Average Age of Information in Update Systems with Active Sources and Packet Delivery Errors", *IEEE Wireless Communications Letters*, Vol. 9, No. 8, pp. 1164–1168, 2020.

176. Atamtürk, A., G. L. Nemhauser and M. W. Savelsbergh, "Conflict Graphs in Solving Integer Programming Problems", *European Journal of Operational Research*, Vol. 121, No. 1, pp. 40–55, 2000.

177. Chudnovsky, M. and P. D. Seymour, "The Structure of Claw-Free Graphs", *Surveys in Combinatorics*, Vol. 327, pp. 153–171, 2005.

178. Köse, A., H. Gökcesu, N. Evirgen, K. Gökcesu and M. Médard, "A Novel Method for Scheduling of Wireless Ad Hoc Networks in Polynomial Time", *IEEE Transactions on Wireless Communications*, Vol. 20, No. 1, pp. 468–480, 2021.

179. Alon, N. and N. Kahale, "Approximating the Independence Number via the ϑ-Function", *Mathematical Programming*, Vol. 80, No. 3, pp. 253–264, 1998.

180. Rappaport, T. S., *Wireless Communications: Principles and Practice*, Prentice Hall, New York, NY, USA, 1998.

181. Tripathi, V. and E. Modiano, "An Online Learning Approach to Optimizing Time-Varying Costs of AoI", *International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, Shanghai, China, pp. 241–250, 2021.

182. Prasad, A., V. Jain and S. Moharir, "Decentralized Age-of-Information Bandits", *IEEE Wireless Communications and Networking Conference*, Nanjing, China, pp. 1–6, 2021.

183. Maatouk, A., 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, 2020.

184. Tripathi, V. and E. Modiano, "A Whittle Index Approach to Minimizing Functions of Age of Information", *Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, pp. 1160–1167, 2019.

185. Weingartner, H. M. and D. N. Ness, "Methods for the Solution of the Multidimensional 0/1 Knapsack Problem", *Operations Research*, Vol. 15, No. 1, pp. 83–103, 1967.

186. Köse, A. and B. Özbek, "Resource Allocation for Underlaying Device-to-Device Communications Using Maximal Independent Sets and Knapsack Algorithm", *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Bologna, Italy, pp. 1–5, 2018.

187. Ferdosian, N., M. Othman, B. M. Ali and K. Y. Lun, "Multi-Targeted Downlink Scheduling for Overload-States in LTE Networks: Proportional Fractional Knapsack Algorithm with Gaussian Weights", *IEEE Access*, Vol. 5, pp. 3016–3027, 2017.

188. Goldsmith, A., *Wireless Communications*, Cambridge University Press, New York, USA, 2005.

189. Sun, Y., E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal and N. B. Shroff, "Update or Wait: How to Keep Your Data Fresh", *IEEE Transactions on Information Theory*, Vol. 63, No. 11, pp. 7492–7508, 2017.

190. Wei, L. and V. Srivastava, "On Distributed Multi-Player Multiarmed Bandit Problems in Abruptly Changing Environment", *IEEE Conference on Decision and Control*, Miami Beach, FL, USA, pp. 5783–5788, 2018.

191. Lai, K. and M. Goemans, "The Knapsack Problem and Fully Polynomial Time Approximation Schemes (FPTAS)", *Retrieved November*, Vol. 3, p. 2012, 2006.

192. Brown, B., "Great Expectations: The Theory of Optimal Stopping", *Journal of the Royal Statistical Society: Series A*, Vol. 135, No. 4, p. 610, 1972.

193. Stigler, G. J., "The Economics of Information", *Journal of Political Economy*, Vol. 69, No. 3, pp. 213–225, 1961.

194. Gittins, J. C., "Bandit Processes and Dynamic Allocation Indices", *Journal of the Royal Statistical Society: Series B*, Vol. 41, No. 2, pp. 148–164, 1979.

195. Liu, Y. and M. Effros, "Finite-Blocklength and Error-Exponent Analyses for LDPC Codes in Point-to-Point and Multiple Access Communication", *IEEE International Symposium on Information Theory*, Los Angeles, CA, USA, pp. 361–366, 2020.

196. Berlekamp, E., R. McEliece and H. Van Tilborg, "On the Inherent Intractability of Certain Coding Problems", *IEEE Transactions on Information Theory*, Vol. 24, No. 3, pp. 384–386, 1978.

197. Arikan, E., "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", *IEEE Transactions on information Theory*, Vol. 55, No. 7, pp. 3051–3073, 2009.

198. Gallager, R. G., *Information Theory and Reliable Communication*, John Wiley and Sons, Inc., New York, USA, 1968.

199. Mori, R. and T. Tanaka, "Performance of Polar Codes with the Construction Using Density Evolution", *IEEE Communications Letters*, Vol. 13, No. 7, pp. 519–521, 2009.

200. Pedarsani, R., S. H. Hassani, I. Tal and E. Telatar, "On the Construction of Polar

Codes", *IEEE International Symposium on Information Theory*, St. Petersburg, Russia, pp. 11–15, 2011.

201. Tal, I. and A. Vardy, "How to Construct Polar Codes", *IEEE Transactions on Information Theory*, Vol. 59, No. 10, pp. 6562–6582, 2013.

202. Arikan, E. and E. Telatar, "On the Rate of Channel Polarization", *IEEE International Symposium on Information Theory*, Seoul, South Korea, pp. 1493–1495, 2009.

203. Shi, Z., H. Wang, Y. Fu, X. Ye, G. Yang and S. Ma, "Outage Performance and AoI Minimization of HARQ-IR-RIS Aided IoT Networks", *IEEE Transactions on Communications*, Vol. 71, No. 3, pp. 1740–1754, 2023.

204. Abd-Elmagid, M. A. and H. S. Dhillon, "Average Peak Age-of-Information Minimization in UAV-Assisted IoT Networks", *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 2, pp. 2003–2008, 2019.

205. Huang, X. and X. Fu, "Fresh Data Collection for UAV-Assisted IoTs Based on Aerial Collaborative Relay", *IEEE Sensors Journal*, Vol. 23, No. 8, pp. 8810–8825, 2023.

206. Kadota, I., 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.

207. Håkansson, V. W., N. K. Venkategowda, S. Werner and P. K. Varshney, "Optimal Scheduling of Multiple Spatiotemporally Dependent Observations for Remote Estimation Using Age of Information", *IEEE Internet of Things Journal*, Vol. 9, No. 20, pp. 20308–20321, 2022.

208. Hatami, M., M. Leinonen, Z. Chen, N. Pappas and M. Codreanu, "On-Demand AoI Minimization in Resource-Constrained Cache-Enabled IoT Networks with

Energy Harvesting Sensors", *IEEE Transactions on Communications*, Vol. 70, No. 11, pp. 7446–7463, 2022.

209. Hsu, Y.-P., "Age of Information: Whittle Index for Scheduling Stochastic Arrivals", *IEEE International Symposium on Information Theory*, Vail, CO, USA, pp. 2634–2638, 2018.

210. Chen, K. and L. Huang, "Age-of-Information in the Presence of Error", *IEEE International Symposium on Information Theory*, Barcelona, Spain, pp. 2579–2583, 2016.

211. Talak, R., S. Karaman and E. Modiano, "Optimizing Information Freshness in Wireless Networks Under General Interference Constraints", *IEEE/ACM Transactions on Networking*, Vol. 28, No. 1, pp. 15–28, 2019.

212. Zhu, Y., X. Yuan, B. Han, Y. Hu and A. Schmeink, "Average Age-of-Information Minimization in EH-Enabled Low-Latency IoT Networks", *IEEE International Conference on Communications*, Montreal, QC, Canada, pp. 1–6, 2021.

213. Koike-Akino, T., P. Larsson, P. Popovski and V. Tarokh, "Non-Linear Network Coding in Two-Way Relaying Discrete Channels", *International Conference on Wireless Communications and Signal Processing*, Nanjing, China, pp. 1–5, 2009.

214. Ho, T., R. Koetter, M. Medard, D. R. Karger and M. Effros, "The Benefits of Coding over Routing in a Randomized Setting", *IEEE International Symposium on Information Theory*, Yokohama, Japan, p. 442, 2003.

215. Wu, Y., P. A. Chou and K. Jain, "A Comparison of Network Coding and Tree Packing", *IEEE International Symposium on Information Theory*, Chicago, Illinois, USA, p. 143, 2004.

216. Sanders, P., S. Egner and L. Tolhuizen, "Polynomial Time Algorithms for Network Information Flow", *ACM Symposium on Parallel Algorithms and Architectures*,

San Diego, California, USA, pp. 286–294, 2003.

217. Liu, Y., Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan and L. Hanzo, "Nonorthogonal Multiple Access for 5G and Beyond", *Proceedings of the IEEE*, Vol. 105, No. 12, pp. 2347–2381, 2017.

218. Ross, S. M., *A First Course in Probability*, Pearson Prentice Hall, New Jersey, United States, 2014.

219. Liu, J., X. Wang, B. Bai and H. Dai, "Age-Optimal Trajectory Planning for UAV-Assisted Data Collection", *IEEE International Conference on Computer Communications*, Honolulu, USA, pp. 553–558, 2018.

220. Avverahalli Ravi, D., *Identifying and Prioritizing Critical Information in Military IoT: Video Game Demonstration*, Ph.D. Thesis, Virginia Tech, 2021.

221. Ojo, M. O., S. Giordano, G. Procissi and I. N. Seitanidis, "A Review of Low-End, Middle-End, and High-End IoT Devices", *IEEE Access*, Vol. 6, pp. 70528–70554, 2018.

222. Perahia, E., "IEEE 802.11n Development: History, Process, and Technology", *IEEE Communications Magazine*, Vol. 46, No. 7, pp. 48–55, 2008.

223. Mackensen, E., M. Lai and T. M. Wendt, "Bluetooth Low Energy (BLE) Based Wireless Sensors", *IEEE Sensors*, Taipei, Taiwan, pp. 1–4, 2012.

224. Narita, I., F. Popescu and C. Despina-Stoian, "Internet of Things Communication Systems Based on MQTT and BLE Protocols", *Journal of Military Technology*, Vol. 4, No. 2, pp. 39–44, 2021.

225. Danbatta, S. J. and A. Varol, "Comparison of Zigbee, Z-Wave, Wi-Fi, and Bluetooth Wireless Technologies Used in Home Automation", *IEEE International Symposium on Digital Forensics and Security*, Barcelos, Portugal, pp. 1–5, 2019.

226. Ramya, C. M., M. Shanmugaraj and R. Prabakaran, "Study on ZigBee Technology", *IEEE International Conference on Electronics Computer Technology*, Kanyakumari, India, pp. 297–301, 2011.

227. Köse, A., M. Koca and E. Anarim, "Impact of Network Coding on Age of Information", *IEEE Internet of Things Journal*, Vol. 11, No. 6, pp. 9725–9739, 2024.

228. Wu, X., J. Yang and J. Wu, "Optimal Status Update for Age of Information Minimization with an Energy Harvesting Source", *IEEE Transactions on Green Communications and Networking*, Vol. 2, No. 1, pp. 193–204, 2017.

229. Arafa, A., K. Banawan, K. G. Seddik and H. V. Poor, "On Timely Channel Coding with Hybrid ARQ", *IEEE Global Communications Conference*, Waikoloa, USA, pp. 1–6, 2019.

230. Arafa, A., J. Yang, S. Ulukus and H. V. Poor, "Age-Minimal Transmission for Energy Harvesting Sensors with Finite Batteries: Online Policies", *IEEE Transactions on Information Theory*, Vol. 66, No. 1, pp. 534–556, 2019.

231. Bacinoglu, B. T., E. T. Ceran and E. Uysal-Biyikoglu, "Age of Information Under Energy Replenishment Constraints", *Information Theory and Applications Workshop*, San Diego, California, USA, pp. 25–31, 2015.

232. Michelusi, N., K. Stamatiou and M. Zorzi, "Transmission Policies for Energy Harvesting Sensors with Time-Correlated Energy Supply", *IEEE Transactions on Communications*, Vol. 61, No. 7, pp. 2988–3001, 2013.

APPENDIX A: COPYRIGHT FOR REUSE OF THE FIGURES

The IEEE does not require individuals working on a thesis to obtain a formal reuse license. Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.

The figures 2.5, 2.6, 2.8, 2.9, and 2.10 have been sourced from IEEE copyrighted papers. The copyright line is included within the captions of these images.