Technische Universität München
TUM School of Computation, Information and Technology

Control-Aware Wireless Resource Management for Networked Control Systems

Onur Ayan, M.Sc.

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:		Prof. Dr.-Ing. Sandra Hirche
Prüfer*innen der Dissertation:	1.	Prof. Dr.-Ing. Wolfgang Kellerer
	2.	Prof. Dr. Anthony Ephremides

Die Dissertation wurde am 11.01.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 15.05.2023 angenommen

Control-Aware Wireless Resource Management for Networked Control Systems

Onur Ayan, M.Sc.

15.05.2023

Abstract

The recent advancements in computing, communications, and sensing have made process control over networks an essential part of today's technology and services. Such applications rely on fast and regular information exchange between hardware and software components in order to accomplish a particular goal in the physical environment. In such a setting, the level of the achieved performance is tightly intertwined with the service offered by the communication network. To date, the main approach taken by the industry and standardization has been centered around either offering higher data rates by introducing new frequency bands or defining specific performance targets to satisfy in the form of a minimum throughput, reliability and/or a maximum end-to-end latency, whereas the meaning and value behind the transmitted bits have been considered irrelevant to the technical problem.

This thesis investigates control-aware network protocol design, specifically focusing on wireless resource management in a multi-user scenario. In particular, it considers a typical networked control scenario comprising multiple feedback loops closed over a wireless communication network. To introduce awareness of the application and context into the network, we assign a value to each piece of information, defined as the expected uncertainty reduction at the destination upon its reception. First, we show that the value of information (VoI) depends on its freshness and control system dynamics, hence is application-specific. Next, we propose novel control-aware resource allocation schemes based on the newly derived VoI metric. In addition, we show that a significantly higher control performance can be achieved than it can be done by conventional techniques. Our evaluation comprises various network topologies ranging from single-hop to multi-hop communication and from constant to time-varying packet loss. Our investigations are not only based on simulation results but also on experimental validations using software-defined radios communicating over a shared wireless channel.

The investigations that are carried out in this thesis should contribute to the network design in future wireless systems, specifically if the users are networked control applications. Our work reveals the need for a paradigm shift from content-agnostic approach to an application- and context-dependent network design if improving the control performance is the primary

goal. This thesis demonstrates an example of control-aware protocol design in wireless communication networks and encourages the adoption of such techniques by the standardization and industry.

Kurzfassung

Die jüngsten Fortschritte in den Bereichen Datenverarbeitung, Kommunikation und Sensorik haben die Prozesssteuerung über Netzwerke zu einem wesentlichen Bestandteil der heutigen Technologie und Dienstleistungen gemacht. Solche Anwendungen sind auf einen schnellen und regelmäßigen Informationsaustausch zwischen Hardware- und Softwarekomponenten angewiesen, um eine bestimmte Aufgabe in der physischen Umgebung zu erfüllen. In einem solchen Umfeld ist das Niveau der erreichten Leistung eng mit dem vom Kommunikationsnetz angebotenen Service verknüpft. Bisher haben sich die Industrie und die Standardisierung hauptsächlich darauf konzentriert, entweder höhere Datenraten durch die Einführung neuer Frequenzbänder anzubieten oder spezifische Leistungsziele in Form eines Mindestdurchsatzes, einer Zuverlässigkeit und/oder einer maximalen End-to-End-Latenz zu definieren. Dadurch wurden die Bedeutung und der Wert hinter den übertragenen Bits als irrelevant für das technische Problem erachtet.

In dieser Arbeit wird das Design von kontrollierten Netzwerkprotokollen untersucht, wobei der Schwerpunkt auf dem drahtlosen Ressourcenmanagement in einem Mehrbenutzerszenario liegt. Insbesondere wird ein typisches vernetztes Regelungsszenario betrachtet, das mehrere Rückkopplungsschleifen umfasst, die über ein drahtloses Kommunikationsnetz geschlossen werden. Um das Bewusstsein für die Anwendung und den Kontext in das Netzwerk einzubringen, weisen wir jeder Information einen Wert zu, der als die erwartete Reduzierung der Unsicherheit am Zielort nach ihrem Empfang definiert ist. Zunächst zeigen wir, dass der Wert der Information (WoI) von ihrer Aktualität und der Dynamik des Regelungssystems abhängt und daher anwendungsspezifisch ist. Anschließend schlagen wir neuartige kontrollorientierte Ressourcenzuweisungssysteme vor, die auf der neu abgeleiteten WoI-Metrik basieren. Darüber hinaus zeigen wir, dass so eine deutlich höhere Kontrollleistung erreicht werden kann als mit herkömmlichen Techniken. Unsere Evaluierung umfasst verschiedene Netzwerktopologien, die von Single-Hop- bis zu Multi-Hop-Kommunikation und von garantierten Übertragungen bis zu zeitvariablen Paketverlusten reichen. Unsere Untersuchungen basieren nicht nur auf Simulationsergebnissen, sondern auch auf experimentellen Validierungen mit

softwaredefinierten Funkgeräten, die über einen gemeinsamen drahtlosen Kanal kommunizieren.

Die in dieser Arbeit durchgeführten Untersuchungen sollten einen Beitrag zum Netzwerkdesign zukünftiger drahtloser Systeme leisten, insbesondere wenn es sich bei den Benutzern um vernetzte Regelungssystemen handelt. Unsere Arbeit zeigt die Notwendigkeit eines Paradigmenwechsels von einem inhaltsagnostischen Ansatz zu einem anwendungs- und kontextabhängigen Netzwerkdesign, wenn die Verbesserung der Kontrollleistung das primäre Ziel ist. Diese Arbeit demonstriert ein Beispiel für kontrollorientiertes Protokolldesign in drahtlosen Kommunikationsnetzen und ermutigt zur Übernahme solcher Techniken durch die Standardisierung und die Industrie.

Acknowledgment

Firstly, I want to express my gratitude to Prof. Wolfgang Kellerer for having me in his team, guiding me through this very special journey, and helping me to overcome all the challenges that I faced over the past years. His management style, great leadership, and open-mindedness made us all perceive LKN as a big family and kept us motivated to learn and explore new horizons. Thanks to his endless support and guidance, my time at the department will remain as a very happy phase of my life.

A special thanks goes to Prof. Anthony Ephremides who hosted me as a visiting researcher in his department at University of Maryland, College Park. Although my time there had to be interrupted due to the Covid-19 pandemic, I was able to learn many things from him during my visit as well as our continuing collaboration afterwards. I am very grateful and honored for having collaborated with such a knowledgeable and special researcher.

I would also like to thank Prof. Sandra Hirche, who has always provided me insightful inputs and answered my endless questions about the control theory counterpart of my research. I learned a lot from her during our Cyber-Physical Networking project.

Next, I want to thank all my former colleagues at LKN for the fruitful discussions and inspiring collaborations. In particular, I would like to thank Mikhail Wilhelm, Markus Klügel, Samuele Zoppi and Halit Murat Gürsu for their valuable inputs and discussions while I was still trying to shape my research direction. I learned a lot from them, especially in the first years of my PhD. In addition, I want to use this opportunity to thank Arled Papa, Polina Kutsevol, Hasan Yağız Özkan, Yash Deshpande and Valentin Haider for our very productive collaborations. A major acknowledgment goes to LKN buddies, who became my best friends in a very short time and transformed our working place into something very special and unique.

I cannot thank my girlfriend, Milena, enough for her support and help since the day we met. Her kind heart, positive energy and beautiful soul has lifted me up and encouraged me in the most challenging days of this process. Moreover, I would also like to thank the entire Aleksić family for their support.

Last but not least, I would like to dedicate this thesis to my parents who supported me in every possible way during my entire life. Their unconditional support and love has accompanied me from the first day of school in Edirne until the last day of my education in Munich. They are the main reason why this thesis and these lines have been written.

Contents

Acronyms

3GPP 3rd Generation Partnership Project 2

ACK Acknowledgment 121, 122

ADRA Age-Dependent Random Access 121, 122, 123, 124, 127, 128, 129

AIRA Age-Independent Random Access 121

AoI Age of Information 2, 3, 4, 7, 8, 11, 12, 13, 14, 21, 23, 24, 25, 26, 29, 30, 34, 37, 38, 40, 41, 44, 47, 48, 49, 50, 51, 53, 55, 57, 59, 67, 73, 74, 75, 78, 79, 80, 82, 83, 84, 85, 87, 90, 92, 93, 99, 100, 101, 110, 111, 113, 115, 116, 117, 119, 121, 122, 123, 124, 125, 126, 128, 129, 130, 132, 134, 135, 136, 139, 140, 141, 157, 158, 159, 161, 163

ATAS Augmentation Type Approximating Sequence 54, 55, 61

BS Base Station 43, 44, 47, 48, 52, 76, 77, 78, 79, 80, 81, 82, 89, 97, 124, 161

CAP Channel Access Probability 121, 122, 163

CRC Cyclic Redundancy Check 106, 127

CSMA Carrier Sense Multiple Access 14

DARE Discrete Algebraic Riccati Equation 18, 73, 81, 115

DL Downlink 2, 6, 8, 44, 45, 46, 47, 48, 76, 77, 78, 82, 87, 88, 96, 97, 159, 161

DP Dynamic Programming 97, 98, 162

FCFS First Come First Serve 7, 13, 21, 32, 40, 113, 121, 135

FCFS-HD First Come First Serve with Head Drop 107, 108, 110, 112, 162

FCFS-TD First Come First Serve with Tail Drop 107, 108, 110, 112, 162

FDD Frequency Division Duplex 107

FH Finite Horizon 63, 68, 70, 71, 72, 73, 76, 85, 89, 93, 96, 97, 98

GE Gilbert-Elliott 64, 77

GW Gateway 113, 114, 121, 123, 124, 125, 163

HW Hardware 1, 6, 7, 9, 124, 135

ICDF Inverse Cumulative Distribution Function 30

IRSA Irregular Repetition Slotted ALOHA 14

KPI Key Performance Indicator 49, 59, 99, 133

LCFS Last Come First Serve 7, 40, 41, 78

LCFS-PD Last Come First Serve with Packet Drop 107, 108, 110, 112, 113, 121, 162, 165

LQG Linear-Quadratic-Gaussian 39, 41, 81, 83, 93, 99, 115, 116, 118, 119, 120, 128, 129, 131, 141

LQR Linear Quadratic Regulator 18, 115, 128

LTI Linear Time-Invariant 15, 18, 31, 45, 52, 63, 105, 106, 113

MAC Medium Access Control 6, 7, 13, 43, 95, 96, 100, 103, 104, 106, 110, 120, 127, 135, 136, 137, 140, 141, 162

MAF Maximum Age First 84, 85, 92

MDP Markov Decision Process 14, 53, 54, 55, 62, 63

MEF Maximum Error First 84, 85, 92, 95, 97, 120, 124, 125, 127, 130, 131, 133, 134

MSE Mean Squared Error 8, 33, 34, 35, 36, 37, 38, 39, 41, 48, 49, 50, 53, 54, 59, 61, 62, 68, 69, 73, 74, 76, 81, 82, 83, 84, 85, 89, 90, 92, 101, 116, 120, 124, 125, 128, 131, 132, 141, 158, 159, 161

MT Maximum Throughput 83, 84, 93, 94, 96

NCS Networked Control Systems 134, 135, 137

NMSE Normalized Mean Squared Error 92, 93, 94, 95, 96, 98, 99, 101, 125, 126, 128, 132, 133, 134, 161, 162

PDF Probability Density Function 117

pMEF Polling-based Maximum Error First 130, 131, 132, 133, 134

PMF Probability Mass Function 7, 21, 29, 34, 35, 37, 41, 158

QoC Quality of Control 1, 4, 6, 8, 9, 18, 20, 39, 43, 81, 93, 103, 115, 116, 120, 131, 137

QoS Quality of Service 5

RAN Radio Access Network 43

RHS Right-hand Side 19, 34, 48, 56, 70, 88, 117, 125, 128

RR Round-Robin 83, 84, 93, 94, 96, 107, 109, 110, 111, 112, 120, 122, 123, 127, 130, 131, 134, 162, 165

RTT Round Trip Time 110, 111, 112

RV Random Variable 117

SA Slotted ALOHA 121, 122, 123, 124, 127, 128, 129

SDR Software-Defined Radio 109, 113, 114, 121, 126, 136, 163

SDRs Software-Defined Radios 7, 9, 104, 106, 109, 120, 121, 123, 126, 135, 136, 162

SoI Semantics of Information 3, 5, 21, 103

SW Software 1, 6

TCP Transmission Control Protocol 135

TDMA Time-Division Multiple Access 135

UAV Unmanned Aerial Vehicle 116

UAVs Unmanned Aerial Vehicles 1, 99, 105

UDP User Datagram Protocol 106, 126, 135

UL Uplink 2, 6, 8, 44, 45, 46, 47, 48, 52, 76, 77, 78, 79, 82, 84, 87, 88, 90, 96, 159, 161

VoI Value of Information 3, 6, 48, 51, 103, 139, 140

WNCS Wireless Networked Control Systems 1, 2, 3, 4, 5, 7, 8, 11, 19, 20, 21, 41, 44, 75, 100, 103, 141

WSN Wireless Sensor Networks 22

ZOH Zero-Order-Hold 66

Chapter 1

Introduction

The recent advancements in computing, communications, sensing and control have made real-time control an essential part of today's technology and services. Examples are abundant: autonomous cars [ZGK13], Unmanned Aerial Vehicles (UAVs) [Cue+19], smart home, smart agriculture, and telerobotics [ZGK13]. Nowadays, a significant portion of such systems relies on timely and regular information exchange between their Hardware (HW) and Software (SW) components over wireless communication networks, mainly because of their reduced installation and maintenance costs and increased system flexibility [Par+18].

From a systems theory perspective, control systems that are closed over a wireless communication network are called *Wireless Networked Control Systems (WNCS)*. A typical WNCS involves a *controller*, which aims to drive the state of a *plant* process to the desired value. The controller actions, also referred to as control input or signal in the literature, are calculated with the help of a *sensor* that operates remotely and sends measurements over a wireless link. As a result, the performance of WNCS becomes strictly intertwined with the behavior of the network. If the network cannot provide the controller with regular and accurate information, e.g., due to high packet loss, or the incoming information does not accurately represent the system's current state, e.g., due to high end-to-end delays, the ability of the system to achieve control-specific goals drops. In this case, we talk about the deterioration of the *control performance* or, equivalently, of the *Quality of Control (QoC)*. Generally, significant degradation of QoC leads to physical damage in the system or the environment. In extreme cases, it may even result in severe injuries or death.

Inserting a wireless network into the feedback loop presents new challenges both for control and communications engineering and research. In addition to requiring multidisciplinary knowledge, it forces researchers to revisit the analytical methods, considerations, and design choices from the conventional control theory to find novel solutions in order to minimize

the adverse effects caused by the network. Broadly speaking, the research on WNCS can be classified into two broad categories [GC10]:

I. *Control over networks* deals with the stability analysis and control strategies for WNCS to model and compensate for network-induced imperfections.

II. *Control of networks* aims to make the communication networks more suitable for WNCS applications through efficient and application-aware protocol design and architectures.

The first category has been heavily studied by the control community over the past decades and is not the focus of this thesis. The second category, into which the scope of this thesis falls, has shown limited adoption by the industry and researchers, although 3rd Generation Partnership Project (3GPP) considers remote control and monitoring as one of the key driving use cases of the fifth generation (5G) cellular networks. Generally speaking, the problem of making the communication networks more suitable for emerging applications has been simplified down to the parameter selection problem of a given network or optimization problems w.r.t. conventional metrics such as throughput or delay [LG04; PAJ11; SSM18; Par+18].

The main approach taken by the networking community has been limited to either offering wider bandwidth to cope with the increasing demand for higher data rates or satisfying selected performance targets specified for the considered use case [CB21]. For example, 3GPP sets up to $1\ Gbit/s$ Downlink (DL), and $500\ Mbit/s$ Uplink (UL) data rate targets for indoor hotspot environments, as well as an end-to-end latency bound up to $10\ ms$ [TS22.261]. In addition, it is also common in the standardization to define performance targets that are specific to the type of the industrial application. To name a few examples, [TS22.104] requires the communication network to achieve a maximum end-to-end latency of 1 ms to 50 ms for mobile robots and 10 ms for process automation and closed loop control for periodic traffic. This way, the network is expected to reliably transmit a given data stream in its entirety as fast as possible, while remaining oblivious to the meaning behind the transmitted bits.

However, due to the rapidly increasing number of connected devices, a bottleneck represented by the scarcity of resources is considered inevitable. Thus, the networking community is forced to explore new metrics and methods beyond one-size-fits-all approaches, as in the case of abstraction through delay or throughput. A step towards this vision was the introduction of Age of Information (AoI), proposed in [Kau+11]. AoI is a metric quantifying *information freshness* and is defined as the time that has elapsed since the generation of the freshest information about a monitored process available at the destination. It is particularly applicable to systems by which fresh information is desired to achieve better performance,

such as in a typical WNCS scenario. The main difference between AoI and conventional metrics, such as throughput or delay, is that AoI is an application layer metric from the perspective of the destination and is defined only if the transmitted information has a generation time. That means it does not apply to general networking use cases, such as file transfer or video-on-demand services, e.g., Netflix, and YouTube, as the transmitted data do not possess any real-time characteristics.

Information freshness has emerged as a novel field in networking research together with increasing popularity of the metric associated with it. AoI has attracted the research community as it gives direct insights into how much a particular packet would contribute to reducing staleness in case of a successful reception. This differs from forwarding a packet as fast as possible or sending as many bits as possible. If the transmitted information is already outdated until its reception time, the benefit of receiving this information is generally expected to be low from the monitor's perspective. In fact, it has been shown that timely updating is not the same as maximizing network utilization or ensuring that the transmitted packets are received with the lowest possible delay [Yat+21; KYG12a].

By definition, the AoI captures the timeliness aspect of information, whereas it is independent of the content. In other words, according to the AoI metric, two pieces of information are of identical importance, if they are equally old. However, if the communication is conveyed for a particular application-specific goal, the real "value" behind a transmission is inseparable from its communication purpose and the context. The value of a piece of information beyond its freshness is addressed by the emerging notion of *Semantics of Information (SoI)*, defined as the significance of data relative to its transmission purpose [Uys+22; Pop+20]. The consideration of SoI opens the way for prioritization of the most valuable information in a congested scenario and/or prevents a wasteful utilization of the limited networked resources by allowing only those transmissions that are essential for the considered task. While the value and freshness of information fall under the umbrella of SoI, it is of utmost importance that future communication systems are designed in a semantics-aware manner if a complete convergence of control and communications is to be achieved.

In the research field of networked control, the application of AoI and Value of Information (VoI) is a novel research question around which this thesis is centered. This thesis aims to design network protocols addressing the resource scarcity problem with a special focus on centralized resource scheduling for WNCS. In particular, it proposes scheduling policies that are AoI- and VoI-aware and investigates the connection between SoI and control performance. The question we would like to answer is how to efficiently distribute resources among multiple feedback control loops closed over a shared wireless communication network if the network

cannot accommodate the generated traffic in its entirety. However, the design, optimization, and realization of such protocols come with a set of challenges that this thesis aims to tackle.

1.1 Research Challenges and Contributions

The following text gives a more detailed overview of the key research challenges this thesis aims to tackle. Moreover, it highlights the main contributions related to each research challenge. In the remainder of this section and the following chapters, we do not strictly follow a chronological order of the underlying publications for presentation purposes.

A) Metric selection, derivation and cross-domain system modeling

One of the first steps towards efficient network design for WNCS is to provide the network the ability to assess the importance of a piece of information[1] for the underlying task. This is only possible through metrics that successfully capture the application-specific performance and the efficiency at fulfilling the communication purpose. However, this is not a straightforward task as today's wireless networks do not cater to the value of a particular piece of information for the targeted communication goal. Instead, multiple variations of throughput, delay and fairness are used to approximate the demand for specific applications and services [Mam+22; TS22.104].

At this point, the following research questions arise: *How to classify the value of a particular piece of information in the context of WNCS*? Specifically, *how is the freshness aspect of information connected to its value and importance for the communication purpose*? Last but not least, *how can the network utilize this additional information to improve its decision-making and the quality of the resulting task accomplishment*? Answering these questions is not trivial as it requires deep knowledge of control and communications.

Contributions:

We first consider a single feedback control loop closed over a multi-hop line network. Firstly, we are interested in the behavior of AoI at the destination when each link is subject to packet loss. To that end, we derive a closed-form expression for the probability distribution of AoI. As our primary objective is to improve the QoC of control applications but not to maximize information freshness, we derive the dependency between the AoI and network-induced estimation error in a feedback control loop. In particular, we show that the problem of QoC maximization is not an equivalent problem to maximizing information freshness. Furthermore, we combine the notion of AoI distribution and estimation error into an optimization

[1] Here, a "piece of information" refers to the smallest amount of bits that are meaningful together and contain the state of a particular process, e.g., a status update packet in process automation.

problem and demonstrate how the distributions of age can be utilized to improve control performance in WNCS.

B) Design of feasible and optimal control-aware networking protocols

Network design for WNCS mandates broad considerations that flow into the design process, e.g., channel conditions for individual users, hardware impediments, energy awareness, and the requirements specific to the underlying services and applications. In current cellular networks, use-case-specific requirements are handled through separation of Quality of Service (QoS) flows into different profiles [TS23.501]. In case of guaranteed bit rate flows, to which most of the industrial applications belong, each traffic flow is characterized by certain QoS parameters such as guaranteed flow bit rate, maximum flow bit rate, and maximum packet loss. Consequently, the network tries to match the QoS requirements of the traffic through cross-layer protocols, e.g., in data link layer through QoS-aware radio resource management. Widely adopted QoS-aware scheduling policies are predominantly based on such requirements, e.g., throughput, delay, and energy consumption [Cap+13]. As a result, the network, and especially how the scheduling is done, is entirely agnostic to the content of the transmitted data and SoI, such as freshness or value. However, the network could benefit from going one step beyond conventional metrics, especially regarding resource management for WNCS.

When it comes to designing networking protocols that are aware of the control applications (i..e, control-aware) and the semantics of the transmitted information, e.g., value, there are several complications and aspects that one needs to consider. First, the scheduler should continuously be able to prioritize users based on channel conditions, the information available at the destination, and the information that is stored in the transmission buffers. However, having practical feasibility in mind, it would be unrealistic to assume that the scheduler has global knowledge about the system, e.g., the actual content of the information that is present on the other side of the network. The challenge here is to find a control-aware solution to the scheduling problem that is feasible from a practical perspective and efficient when prioritizing the most critical and urgent transmissions in the network. Second, the existing literature dealing with centralized scheduling policies that depend on the time dynamics of control systems is very limited. Such policies should be designed while system-specific models from the control theory flow into the decision-making process. This constitutes another challenge as it implies that these control-system-specific models should be propagated down to the data link layer where the scheduling takes place. Moreover, this comes with an additional price of increased complexity and effort while designing such protocols and requires careful consideration for the overall system design.

Contributions:

To tackle the research questions elaborated above, we consider a wireless network comprised of multiple heterogeneous feedback control loops. We set our focus on wireless Medium Access Control (MAC) protocols and study how the limited network resources should be distributed among network resources to achieve a higher QoC. Our cross-layer approach is based on the utilization of the VoI, which captures the amount of uncertainty reduction at the monitor in case of a successful transmission, for scheduling decisions. We show that already with simple heuristics based on the VoI, we are able to improve the control performance when compared to providing information freshness in the network.

In order to extend our heuristic approach, we first consider a shared UL scenario with constant packet loss and formulate a stochastic optimization problem, which we solve optimally through the value iteration technique. If the loss is dynamic, then the scheduling problem is modeled as a finite horizon problem and solved optimally through dynamic programming. In addition, we extend the problem formulation to a joint UL and DL setting. We show through Monte Carlo experiments that the proposed optimal policies outperform the greedy approach as well as other conventional scheduling policies widely used in the existing literature.

C) Practical implementation of control-aware scheduling algorithms

The theory allows us to model the interconnection between control and communication. Doing so, it enables us to design and test novel networking mechanisms hinting at potential performance increase when employed. However, when it comes to real-life deployment, the promised performance gain may diminish on account of the random nature of the wireless channel, hardware imperfections and errors. In consequence, the verification of newly proposed solutions in a practical setup becomes of paramount importance.

To validate novel customized protocols using real-life connections, an idea would be to use widely available products and standards, for instance, a Wi-Fi access point based on the IEEE 802.11 standard. However, altering such devices' communication protocol stack is usually not straightforward, which constitutes a barrier to the implementation of such customized scheduling algorithms in practice. Moreover, there are a lot of aspects flowing into the design process ranging from the selection of the HW and tools to the implementation of both control applications and communication layers in SW. Additionally, the practical implementation introduces new challenges that are mostly assumed to be solved in theoretical papers or simulations, e.g., perfect synchronization among users for time-slotted medium access and global knowledge about the network by decision-making entities. As a result, conducting an experimental study and research on such a complex system is challenging and requires a notable effort and time.

Contributions:

We extend the coverage of this thesis from theoretical research to systems research by implementing an end-to-end system involving multiple control applications running in real-time and communicating over a shared physical wireless channel. Due to the flexible adaptation and implementation of their MAC layer processing, we conduct an experimental study involving multiple Software-Defined Radios (SDRs) and control applications of heterogeneous classes. By implementing a broad range of packet queueing and scheduling policies suggested by this thesis' theoretical contributions, we can validate the performance gain through a control-aware protocol design on real HW and pinpoint the essential design issues for their practical deployment. Our implementation is based on the open-source GNU Radio [Rad] framework that provides a rich set of signal processing blocks to replace the traditional hardware modules of transceivers.

1.2 Outline

Fig. 1.1 presents a high-level structure of this thesis for future reference. The remainder of the content can be broken down into chapters as follows.

Chapter 2

This chapter explains the general structure of a WNCS and presents the components of a feedback control loop. Additionally, it briefly introduces the concept of AoI and its application in communication networks. Thereby, it provides the background knowledge from the control theory necessary to follow the remaining chapters.

Chapter 3

This chapter deals with communication over a multi-hop line network for two distinct packet queueing strategies. First, it studies the probability distribution of AoI for an N-hop line network where each hop employs a Last Come First Serve (LCFS) queue with packet discard, which has been shown to be optimal for maximized information freshness. Here, we perform a probability analysis and derive the Probability Mass Function (PMF) for the discrete-time AoI model. Next, we relax the assumption of the LCFS queueing strategy and consider a multi-user network comprised of multiple First Come First Serve (FCFS) queues, which is the most commonly encountered queueing strategy in today's communication systems. Again having the probability distribution of AoI, we utilize it to formulate an optimization problem, and by solving it, we show that the estimation performance is maximized when the optimal resource allocation is selected. In this chapter, we define the direct relationship between AoI

and the expected mean squared estimation error given an age. The Mean Squared Error (MSE) derived in this section is reused in the following chapters to tackle the centralized wireless resource scheduling problem for WNCS. This chapter is based on the following publications:

- O. Ayan, H. Murat Gürsu, A. Papa and W. Kellerer, "Probability Analysis of Age of Information in Multi-Hop Networks", IEEE Networking Letters (LNET), 2020.
- O. Ayan, A. Ephremides and W. Kellerer, "Age of Information: An Indirect Way To Improve Control System Performance", IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2021.

Chapter 4

In this chapter, we study the centralized scheduling for multi-loop scenarios in which the network resources are scarce and thus have to be efficiently distributed among users. We first study the resulting QoC if the network targets a maximized information freshness through AoI-aware scheduling. In the same section, we propose a greedy scheduling algorithm that is based on the age-dependent MSE derived in Chapter 3. Next, we formulate the scheduling problem as a discounted cost problem and find the optimal solution as an improvement over the greedy scheduling algorithm. Here, we apply the *value iteration* technique to find the optimal scheduling decision given a network state as described in [Ber95]. In contrast to the previous algorithm, which guarantees optimality only if the channel conditions are static over time, we convert the problem to a finite horizon problem and solve it for a single-hop wireless link.

In the last section of this chapter, we extend the finite horizon problem to a joint UL and DL scheduling and conduct an extensive study of various scheduling policies from the existing literature. All evaluations in this chapter are based on simulations. The main body of this chapter consists of the following publications:

- O. Ayan, M. Vilgelm, M. Klügel, S. Hirche and W. Kellerer, "Age-of-information vs. value-of-information scheduling for cellular networked control systems", ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), 2019.
- O. Ayan, M. Vilgelm and W. Kellerer, "Optimal Scheduling for Discounted Age Penalty Minimization in Multi-Loop Networked Control", IEEE Consumer Communications & Networking Conference (CCNC), 2020.
- O. Ayan, H. M. Gürsu, S. Hirche and W. Kellerer, "AoI-based Finite Horizon Scheduling for Heterogeneous Networked Control Systems", IEEE Global Communications Conference (GLOBECOM), 2020.

- O. Ayan, A. Ephremides, S. Hirche, W. Kellerer, "Optimal Finite Horizon Scheduling of Wireless Networked Control Systems", IEEE/ACM Transactions on Networking, 2023.

Chapter 5

This chapter implements a centralized scheduling framework on real HW by using SDRs. First, we investigate how the QoC is impacted when different queueing strategies are selected. Next, we proceed with the optimal queueing strategy based on our measurements and tackle the resource allocation problem under different MAC protocols, including contention-based and contention-free techniques. All evaluations in this chapter are conducted through an experimental testbed that involves multiple control and communication processes running in real time. The scope of this chapter is based on the following publications:

- O. Ayan, H. Y. Özkan and W. Kellerer, "An Experimental Framework for Age of Information and Networked Control via Software-Defined Radios", IEEE International Conference on Communications (ICC), 2021.
- O. Ayan, P. Kutsevol, H. Y. Özkan, W. Kellerer, "Semantics- and Task-Oriented Scheduling for Networked Control Systems in Practice", IEEE Access, 2022.

Chapter 6

We conclude this thesis by summarizing the key results. Moreover, we present an outlook for interesting future research directions that can be seen as a natural extension of the main contributions of this work.

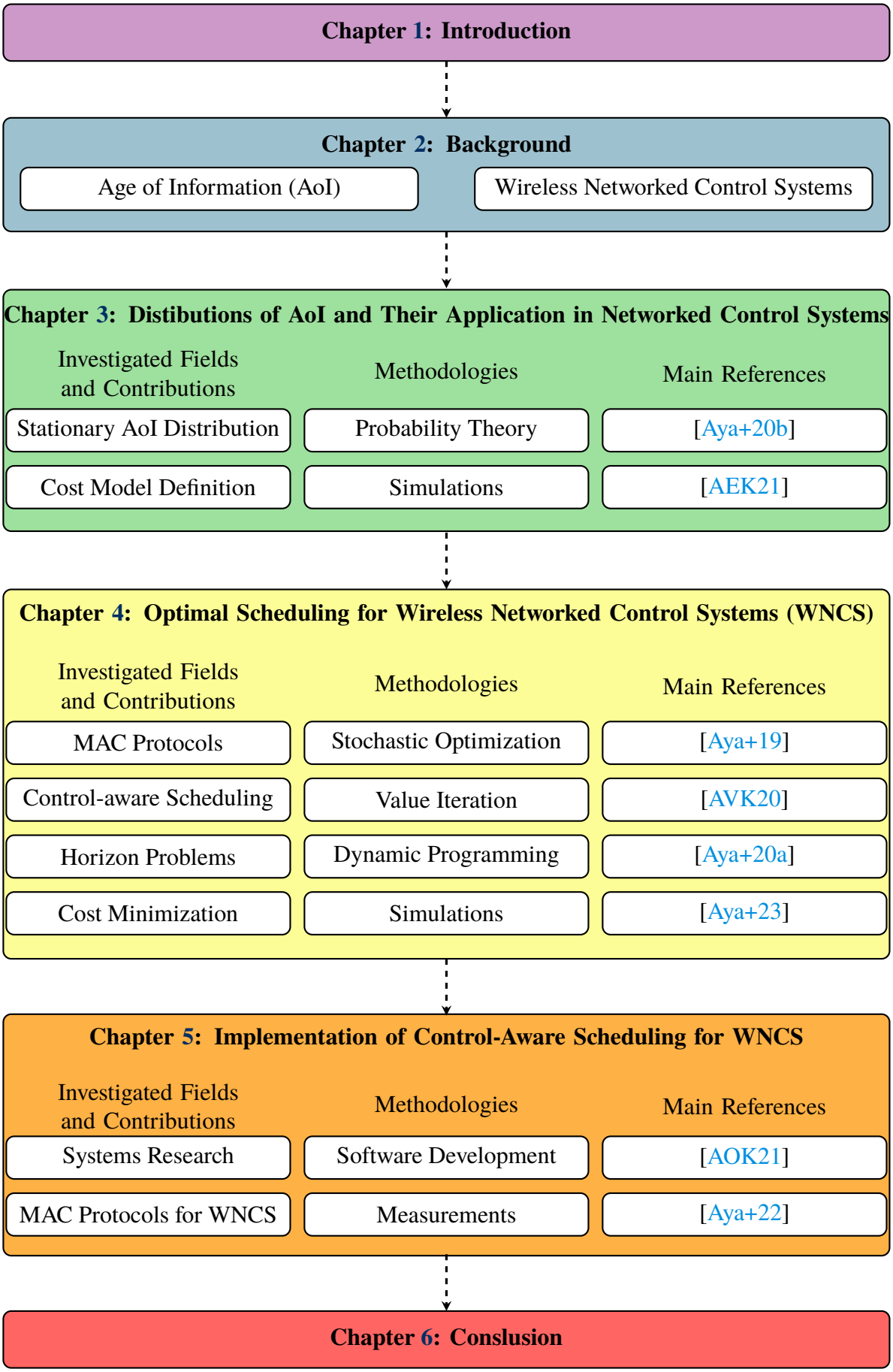


Figure 1.1: Thesis outline highlighting the main contributions, methodologies, and main references of each chapter.

Chapter 2

Background: A Brief Introduction to Information Freshness and Control Systems

This section serves as an introduction to the necessary concepts and topics that are relevant to comprehend the following chapters. It consists of two main blocks. First, we define AoI and give a few examples of how it is used in wireless communication networks. Second, we familiarize the reader with the basics of control theory that appear in the content to come. Please note that the chapter should not serve as a replacement for control theory textbooks. We refer the reader to other sources that are intended to teach control theory, e.g., [rM08].

2.1 Age of Information in Wireless Networks

A notable portion of applications that falls under the category WNCS, e.g., video-based control of an inverted pendulum over a wireless network, a drone in a warehouse that is remotely controlled by a central station, share a common feature: their operation relies on time-stamped state measurements that are transmitted between source-destination pairs through a communication network. Generally, having a more recent, i.e., fresh, observation about the remote system state is beneficial for control performance.

2.1.1 Definition of Age of Information

Suppose a monitor that receives state measurements from a remotely operating sensor in the form of packet transmissions, as depicted in Fig. 2.1. In such a setting, AoI quantifies information freshness from the perspective of the destination. If the freshest information at

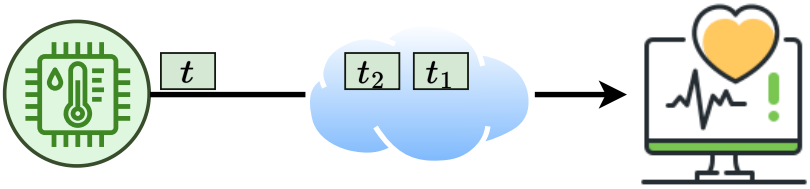


Figure 2.1: A remote sensor, e.g., temperature sensor, humidity sensor, sending system state information to a monitor through a communication network. Measurements are transmitted in the form of status update packets. The sensor has the current state information belonging to time t whereas two past status update packets, i.e., $t > t_2 > t_1$ are on their way to the destination.

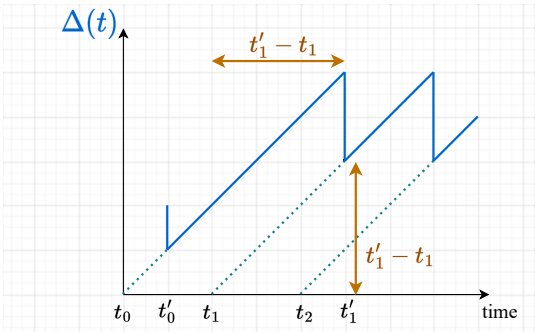


Figure 2.2: An example evolution of AoI in continuous time. Note that the value after a successful reception corresponds to the difference between the generation and the reception time instances.

the monitor at time t has the generation time-stamp $\nu(t)$, then the AoI, $\Delta(t)$, is defined as:

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

In plain words, AoI is the elapsed time since the generation of the most recent information about the system state that the destination has received. This thesis uses AoI and age as synonyms describing the process $\Delta(t)$. Let us have a look at Fig. 2.1 once again and assume that the monitor has successfully received $\nu(t) = t_0$ some time ago. The status update packets containing state measurements from t_1 and t_2 with $t_0 < t_1 < t_2 < t$ are still traversing the communication network. Note that the higher the timestamp, the fresher its information. From Eq. (2.1), it is evident that the value of $\Delta(t)$ grows linearly over time until a fresher packet, e.g., the one generated at t_1, is received. If the packet containing t_1 is delivered at time t_1', the value of $\nu(t_1')$ is updated and the AoI drops to $t_1' - t_1$. In other words, the new value of AoI immediately after the reception corresponds to the time that the packet has spent in the network, i.e., packet delay. An example evolution of AoI following the aforementioned time behavior is depicted in 2.2.

It is important to realize that AoI depends both on delay and inter-delivery time between two successful receptions´ at the destination. One could suppose that a necessary condition to minimize AoI is to inject status update packets into the network as frequently as possible.

However, it has been shown that this may lead to significantly bad AoI performance under certain conditions, e.g. if the packets have to go through FCFS queues [KYG12a]. On the other hand, if the packet generation rate is too low, $\nu(t)$ is updated very rarely, thus leading to high peaks of $\Delta(t)$. This aspect makes the AoI an exciting topic for research and engineering of time-critical applications such as real-time monitoring and control.

We have introduced the concept of AoI and its connection to information freshness. At this point, the following question arises: *How does one measure the performance of a network w.r.t. AoI?* Two metrics that are frequently used in the existing literature stand out: *I)* Time-average AoI, *II)* Average Peak AoI [Yat+21]. The first metric captures the mean AoI during a time period T. If we denote time-average age as $\overline{\Delta}$, it can be formulated as:

$$\overline{\Delta} = \frac{1}{T} \int_0^T \Delta(t) dt. \tag{2.2}$$

Furthermore, the average peak AoI, $\hat{\Delta}$, does not evaluate age during the complete duration of T but considers only those instances at which $\Delta(t)$ is about to drop due to a new reception at the destination. If the set of time instances at which the monitor receives a new status update packet as $\mathcal{D} = \{t'_0, t'_1, \dots\}$, as in Fig. 2.2, the average peak AoI can be obtained from:

$$\hat{\Delta} = \frac{1}{|\mathcal{D}|} \sum_{\{t \mid t \in \mathcal{D}\}} \Delta(t) \tag{2.3}$$

with $|\mathcal{D}|$ being the cardinality of set $\mathcal{D}$. Throughout this thesis, we are going to employ $\overline{\Delta}$ to quantify performance w.r.t. information freshness. Note that a lower $\overline{\Delta}$ indicates a better AoI performance.

2.1.2 Application of Age of Information in Wireless Networks

AoI has been used in various contexts in the existing literature. This sub-section presents a wide selection of research papers that include AoI in their system model, with a special focus on wireless MAC protocols.

Data link layer: When a communication network is comprised of multiple users, they usually need to share network resources in order to exchange information between source-destination pairs. Especially in a wireless network scenario, a simultaneous transmission using the same frequency resources may corrupt the received information, and in return, lead to a packet loss event. As a result, the monitor can not be updated with more recent information about the remote process; hence AoI keeps growing. Therefore, avoiding simultaneous access to the shared medium is crucial to provide information freshness in a multi-user wireless monitoring scenario.

Some of the existing works consider *contention-based* medium access permitting the utilization of the same network resources by multiple users [Che+22; CGL20; KM21; Mun21; MAE20; Yan+22; YK17]. [Che+22] and [Yan+22] focus on the Slotted ALOHA mechanism and study average AoI performance analytically. Moreover, they show that the age performance can be further improved by applying variations of Slotted ALOHA compared to its conventional version. Similarly, [CGL20] proposes a threshold-based slotted random access protocol that is able to outperform Slotted ALOHA. Furthermore, [Mun21] derives a closed-form expression of the average AoI achieved by Irregular Repetition Slotted ALOHA (IRSA). Their work shows that IRSA outperforms slotted ALOHA w.r.t. information freshness in the network.

On the other hand, in [MAE20; KM21], users use Carrier Sense Multiple Access (CSMA) protocol to access the shared medium. [MAE20] provides a mathematical expression for average AoI when CSMA protocol is employed. [KM21] considers status updates arriving according to a stochastic process. They optimize the CSMA protocol to improve information freshness and additionally verify their analytical conclusions through practical measurements. Last but not least, [YK17] shows that Slotted ALOHA leads to a worse age performance by a factor of $2e$ in comparison to a simple contention-free protocol. The selected contention-free protocol is a simple heuristic allowing users to take turns while limiting the maximum number of retransmissions for each packet.

AoI has been used as a metric in *contention-free* medium access protocols. In such a setting, the users follow a particular transmission schedule that aims to prevent collisions and thus increase the chance of a successful reception significantly. One of the most prominent examples of such is [Kad+18], which addresses the problem of AoI minimization in a single-hop wireless network. In their work, the authors study three simplistic scheduling policies and derive performance guarantees for various network configurations. Another work that studies AoI-optimal scheduling policies is [Kad+16]. In [KM21], the authors show that if all users are experiencing identical channel conditions, the greedy AoI scheduler, prioritizing the user with the highest age, is optimal. Furthermore, [HMD20] develops optimal scheduling algorithms based on Markov Decision Process (MDP) and Whittle index policy depending on whether the information arrival statistics are available as prior knowledge. Another work using the Whittle index policy to minimize AoI is [TM19]. In their work, the authors show that the proposed low-complexity algorithm offers close to optimal performance.

Network and Transport Layers: In addition to the data link layer, the problem of improving information freshness has been studied within higher layers. For instance, [Yat21] evaluates AoI performance in a network that utilizes gossip protocol for packet forwarding. [TKM17; TTM22] propose a scheduling policy for multi-hop networks with interference

constraints. In addition, [FKRB19; FKB19a; FKB19b; BSU19] consider a multi-source multi-hop network setting. In these works, authors follow an analytical approach and derive fundamental bounds for average age performance. Another work focusing on multi-hop networks is [SKY19]. However, in contrast to the works mentioned above, the authors of [SKY19] focus on the transport layer and improve the network's information freshness through a congestion control mechanism called *age control protocol*.

2.2 Related Topics From Control Theory

2.2.1 System Modeling

A *model* is a mathematical representation of a physical, biological or information system [rM08]. Models may be derived either from laws of physics or experimental data and allow us to understand and predict the behavior of systems in time. For instance, the evolution of number of cases in a population during pandemic, the motion of an object in three dimensional space, buffer occupancy in a queuing system are examples to such systems that can be explained and predicted through utilization of mathematical models.

A common way of modeling physical systems, whose state change in time according to a certain rule, is using a set of ordinary differential equations such as:

$$\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{dt} = f\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right), \tag{2.4}$$

where the term $d\boldsymbol{x}/dt$ represents the time derivative of the *system state* $\boldsymbol{x}(t) \in \mathbb{R}^n$. Moreover, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is a vector of external control inputs. If the mapping function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ does not explicitly depend on time, then the system is said to be *time-invariant*.

Let us have a look at a simple example. Eq. (2.5), also known as the governing equation, completely describes the mechanics of a spring-mass system with damping. The variable $q(t) \in \mathbb{R}$ is the displacement of the mass m w.r.t. its rest position. The friction element $c(\dot{q})$ is a nonlinear function of the mass velocity $\dot{q}$.

$$m\ddot{q} + c(\dot{q}) + kq = u(t). \tag{2.5}$$

On the other hand, if the function f is *linear* and time-invariant at the same time, then the system is called a *Linear Time-Invariant (LTI)* system. An LTI system can be represented by a set of equations in the following form:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \tag{2.6}$$

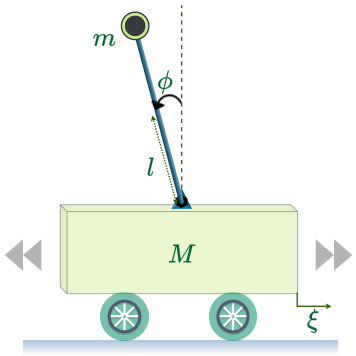


Figure 2.3: An inverted pendulum.

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ are constant matrices. In the literature, the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices are called *system* and *input* matrix, respectively. In many cases, the governing equation of the system contains non-linear elements. However, it is quite common in the control theory literature to linearize them around the equilibrium point that approximates the behavior of the system during operation.

2.2.1.1 An example control application: Inverted Pendulum

A common control application that is widely used in the literature is an *inverted pendulum* depicted in Fig. 2.3. It consists of a pendulum mounted on a motorized cart. The goal of the controller is to keep the pendulum in upright position through horizontal movements of the cart. The continuous-time dynamics of the system are characterized through the following two governing equations:

$$\begin{aligned} (M+m)\ddot{\xi} + b\dot{\xi} - ml\cos(\phi)\ddot{\phi} + ml\sin(\phi)\dot{\phi}^2 &= u \\ -ml\cos(\phi)\ddot{\xi} + (I+ml^2)\ddot{\phi} - mgl\sin(\phi) &= 0 \end{aligned} \tag{2.7}$$

When the system is operating around the equilibrium point, corresponding to a small angular deviation, i.e., $\phi = 0$, we can apply the following approximations:

$$\begin{aligned} \sin(\phi) &\approx \phi \\ \cos(\phi) &\approx 1 \\ \dot{\phi}^2 &\approx 0 \end{aligned} \tag{2.8}$$

After substituting (2.8) into the nonlinear governing equations, (2.7), we obtain the following two linearized equations of motion:

$$\begin{aligned} (M+m)\ddot{\xi} + b\dot{\xi} - ml\ddot{\phi} &= u \\ (I+ml^2)\ddot{\phi} - mgl\phi &= ml\ddot{\xi} \end{aligned} \tag{2.9}$$

Equivalently, we can rearrange the equations into matrix form as follows:

$$\begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)Mml^2} & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u. \tag{2.10}$$

Note the relationship between (2.10) and (2.6) with $\boldsymbol{x} = [\xi \quad \dot{\xi} \quad \phi \quad \dot{\phi}]^T$.

As an alternative to continuous-time representation, one can describe the system state at *discrete time* instances. That is, if we divide the time into equidistant discrete points as $k = 0, 1, 2, \ldots$, we can formulate the system state at $k+1$, i.e., $\boldsymbol{x}[k+1]$, in the form of a difference equation as:

$$\boldsymbol{x}[k+1] = f(\boldsymbol{x}[k], \boldsymbol{u}[k]). \tag{2.11}$$

In other words, (2.11) describes which value the next system state will take as a function of the previous state and the applied control input at time step k. Similar to the continuous case, if the mapping function f is linear in $\boldsymbol{x}$ and $\boldsymbol{u}$, the system is characterized by a matrix equation as:

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k]. \tag{2.12}$$

So far, the considered control system has not contained any stochastic component. That is, given an initial state $\boldsymbol{x}[0]$ and the following control inputs $\boldsymbol{u}[0], \ldots, T$, we can obtain the exact state value at a later time step k according to:

$$\boldsymbol{x}[k] = \boldsymbol{A}^k\boldsymbol{x}[0] + \sum_{j=0}^{k-1} A^{k-j-1}\boldsymbol{B}\boldsymbol{u}[j], \tag{2.13}$$

with $\boldsymbol{A}^p$ representing the p-th power of the matrix $\boldsymbol{A}$. However, in order to bring theory closer to practice, control theory considers random disturbances in the model. This gave rise to *stochastic control theory* that deals with dynamical systems subject to disturbances which are characterized as stochastic processes [Å12]. As a result, we will consider the following difference equation throughout the thesis if not stated otherwise:

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k] + \boldsymbol{w}[k] \tag{2.14}$$

where $\boldsymbol{w} \in \mathbb{R}^n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ are independent random variables following a zero-mean normal distribution with covariance matrix $\boldsymbol{\Sigma}$. Gaussian noise is a very popular choice in the literature to model external disturbances on the system state [Å12; Mam17; Sol+22; Ber95].

2.2.2 Linear Quadratic Regulators

A common and useful way of measuring control performance, i.e., QoC, in control theory, is using a quadratic cost function that depends both on the trajectory of the system state throughout a time horizon and the control effort spent during that period. Consider an LTI system characterized by the difference equation from (2.14) and a quadratic cost function:

$$\mathcal{J} = (\boldsymbol{x}[T])^T \boldsymbol{Q}\boldsymbol{x}[T] + \sum_{k=0}^{T-1} (\boldsymbol{x}[k])^T \boldsymbol{Q}\boldsymbol{x}[k] + (\boldsymbol{u}[k])^T \boldsymbol{R}\boldsymbol{u}[k], \tag{2.15}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{R}^{m\times m}$ are positive semi-definite symmetric matrices of appropriate size. The minimization of $\mathcal{J}$ is referred as Linear Quadratic Regulator (LQR) problem and particularly important if we want to keep the system state close to the origin while taking the control effort into account[1]. The quadratic form is often seen reasonable, as a high penalty is induced if the state deviation is large and a low penalty is induced if the deviation is close to zero [Ber95].

The solution to the LQR problem is given by a linear *stationary* control law:

$$\boldsymbol{u}[k] = -\boldsymbol{L}\boldsymbol{x}[k], \tag{2.16}$$

with $\boldsymbol{L} \in \mathbb{R}^{m\times n}$ mapping a state to the optimal control input. Note that the control law is called stationary as it does not change over time. The optimal feedback gain matrix $\boldsymbol{L}$ minimizing the cost function $\mathcal{J}$ can be calculated as:

$$\boldsymbol{L} = (\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{B} + \boldsymbol{R})^{-1} \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{A}, \tag{2.17}$$

where $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is a positive semi-definite, symmetric matrix satisfying the *Discrete Algebraic Riccati Equation (DARE)*:

$$\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{P}\boldsymbol{A} - (\boldsymbol{A}^T \boldsymbol{P}\boldsymbol{B})(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}\boldsymbol{B})^{-1}(\boldsymbol{B}^T \boldsymbol{P}\boldsymbol{A}) + \boldsymbol{Q}. \tag{2.18}$$

The equation (2.16) is based on the assumption that the controller has the perfect knowledge of the current state $\boldsymbol{x}[k]$ to calculate the control input. However, as we are going to discuss in section 2.2.3 to follow, the linearity of the control law is preserved for the case of imperfect state information.

2.2.3 Wireless Networked Control System(s)

This thesis focuses on control systems that are closed over a wireless communication network. In particular, we assume a scenario in which the system state is transmitted over a wireless

[1] Throughout this thesis, without loss of generality, we assume that the set value lies at the origin, i.e., $\boldsymbol{x} = \boldsymbol{0}$.

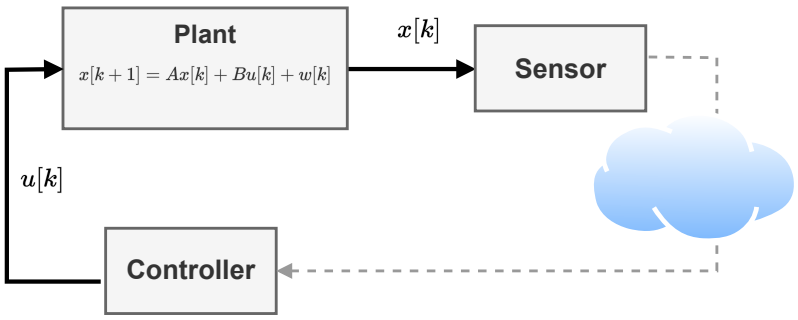


Figure 2.4: A networked control system transmitting sensor measurements over a communication network.

link to the controller. Figure 2.4 illustrates such a feedback loop that includes an imperfect sensor-to-controller link. Considering that a wireless network may introduce packet dropouts and delay, the controller may not necessarily have the perfect knowledge of the current system state $\boldsymbol{x}[k]$ at time k. If this is the case, $\boldsymbol{x}[k]$ on the Right-hand Side (RHS) of (2.16) has to be replaced with, $\hat{\boldsymbol{x}}[k]$, the expected system state from the controller's perspective, i.e,:

$$\boldsymbol{u}[k] = -\boldsymbol{L}\hat{\boldsymbol{x}}[k]. \tag{2.19}$$

A simple way of realizing such a control logic, particularly in practical scenarios, would be to assume the system state to stay constant between two consecutive successful updates at the controller, similar to sample-and-hold mechanism. This approach has been used in various previous works on WNCS, e.g., [HDT13; Tro+21; Bha+21]. However, an alternative technique is running a model-based estimation at the controller that takes advantage of the time-invariant model of the system [LG04; Mam17; Mai+22; MH14].

In fact, for linear systems with quadratic criteria, as in this thesis, it is optimal to consider estimation and control as two separate blocks. That is, an estimation block computing the state estimate in the form of a conditional mean given the observed state. Additionally, a linear feedback that calculates a control input based on the estimated state as if the system state could be measured exactly. This is referred as *separation theorem* in the control literature and such a controller based on it is called *certainty equivalence controller* [Å12; Kl20].

Now, let us consider a scenario as depicted in Fig. 2.4. A sensor periodically samples the plant state and these measurements are transmitted in the form of status update packets over an imperfect wireless link to the controller. Let us further assume that the network introduces packet loss and delay, hence the most recent information about the system state that the controller has is from $\nu(k)$ with $\nu(k) < k$. In other words, it is known to the controller at time step k that the system state at $\nu(k)$ was $\boldsymbol{x}[\nu(k)]$[2]. As a result, the conditional expectation

[2]Here, we implicitly assume that at each sampling event, the sensor records the index of the discrete time step, similar to timestamp, and this information is contained in the status update packets.

of the system state given $\boldsymbol{x}[\nu(k)]$ is defined as:

$$\hat{\boldsymbol{x}}[k] \triangleq \mathbb{E}\left[\boldsymbol{x}[k] \,\middle|\, \boldsymbol{x}[\nu(k)]\right]. \tag{2.20}$$

Due to the stochastic nature of the control system, it is evident that as long as $k \neq \nu(k)$ holds, a mismatch between the actual state $\boldsymbol{x}[k]$ and the estimated state $\hat{\boldsymbol{x}}[k]$ is expected to occur. Consequently, the vector that is defined as the difference between the system state and its conditional expectation can be formulated as:

$$\boldsymbol{e}[k] \triangleq \boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k], \quad \boldsymbol{e} \in \mathbb{R}^n. \tag{2.21}$$

The variable $\boldsymbol{e}[k]$ is also called *network-induced error* in the WNCS literature [YWB00; Mam+17]. The network-induced error plays a major role in the remainder of this thesis. In particular, this thesis is shaped around the following core idea: If the source of performance degradation in QoC is the presence of the imperfect communication network (when compared to the ideal network case), the communication network should try to minimize the network-induced (negative) effects on control applications. As we are going to show in the following sections, we are able to improve the offered control performance by targeting a reduction in the estimation error.

Chapter 3

Distributions of AoI And Their Application in Networked Control Systems

The advancements in sensing, computing, and communications over the past decades have accelerated the adoption of time-sensitive applications in today's networks. Nowadays, a significant portion of data traffic belongs to applications that exchange packets to accomplish application-specific tasks in real-time. In contrast to traditional networking technologies, which were designed to maximize throughput and reduce delay while being oblivious to the information content, the future communication systems are envisioned to incorporate semantics of information such as freshness and value into network and protocol design.

In this chapter, we consider two distinct scenarios that allow us to demonstrate the importance of SoI for network design. In the first scenario, which is mainly based on [Aya+20b], we focus on a single source-destination pair communicating over a multi-hop line network. By modeling the AoI as a discrete-time process, we derive the PMF of AoI analytically, under the assumption that each node is capable of replacing any older packet with a more recent one. In the second scenario, summarizing the main results from [AEK21], we relax the packet replacement assumption but focus on a single-hop queueing system operating according to the FCFS strategy. By leveraging results from the existing literature, which allow us to derive the PMF of AoI in discrete-time queues, we demonstrate how distributions of AoI can be used to improve control performance of WNCS. Some derivations presented in this chapter constitute a basis for the following chapters. An overview of the related work is presented at the end of this chapter.

3.1 The Single-User Multi-Hop Case

3.1.1 System Description

We consider a real-time monitoring scenario in which a physical process sends status updates over an N-hop line network to a monitor. The packets carrying the latest state of the monitored process are generated periodically at the *source node*. We call the generation of an update packet a *sampling event*. The time between two consecutive sampling events is called a *sampling period*.

The network time is divided into equally long slots that can accommodate a single transmission between any consecutive nodes in the network. That is, the transmission of a status update packet starts at the beginning of a slot and completes within the same slot. While a time slot is the smallest time unit in our system model, the sampling period is assumed to be a multiple of a time slot duration. If we denote the sampling period and the time slot duration as t_p and t_s, respectively, it holds that $t_s \cdot m = t_p$ with m being a positive number. For instance, one can imagine a time slot length of $1\ ms$ and a sampling period of $10\ ms$.

Under the assumption that the process' state is Markovian, having received an update, the monitor does not benefit from the reception of older status updates. Thus, older packets are considered obsolete and "non-informative" at any location in the network. To that end, each node discards any older packet in the transmission queue upon the arrival of a fresher update. As a result, our model has no queuing effect as there is always a single packet to be forwarded. A new update at any node is re-transmitted until it is successfully received by its next hop or replaced with more recent information. Please note that this implies the assumption of a feedback mechanism between neighboring nodes, e.g., in the form of an acknowledgment packet.

Generally speaking, in a typical wireless multi-hop scenario, devices are scattered in a wide area. For instance, in an environmental Wireless Sensor Networks (WSN) scenario set up for cattle monitoring, a transmission range of $500\ m$ was used [Cor+10]. In such a setting, depending on the distance between two neighboring nodes, the success probability of a transmission may be less than one, e.g., due to path attenuation. We assume a positive time-invariant packet loss probability between consecutive nodes along the path to model this effect. Such a model corresponds to the Rayleigh block fading model representing the average behavior of the wireless medium. Please note that our model does not allow for spatial or time dependency between two different transmissions.

If we denote the packet loss probability at the n-th hop by p_n, the outcome of any transmission on that link can be abstracted as a Bernoulli trial with a constant failure probability p_n. An example two-hop network that is subject to constant packet loss is depicted in Fig.

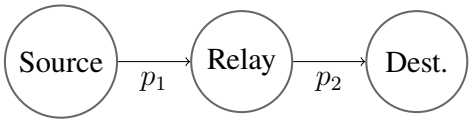


Figure 3.1: An example 2-hop line network with constant packet loss probabilities at each hop. Status update packets are sent from the source node to the relay node, from which they are forwarded to the monitor in a separate transmission.

3.1. The figure illustrates a scenario, in which status update packets are first sent to a relay node with a constant packet loss probability of p_1. The information that is available at the relay node is then forwarded to the destination with a failure probability of p_2.

The nodes follow a fixed transmission schedule and are allowed to transmit only in their respective time slots. In order to simplify the following analysis, we assume that the time slot allocation is in alignment with each node's appearance order along the path. [Alu+09] suggests similar transmission patterns for multi-hop networks that enable the reception of a newly generated status update packet within the same sampling period after N-hops. Fig. 3.2 demonstrates how such a transmission schedule would look like if the source and destination were three hops apart. Note that the status update packet carrying the k-th sample reaches the destination within the same sampling period of seven time slots.

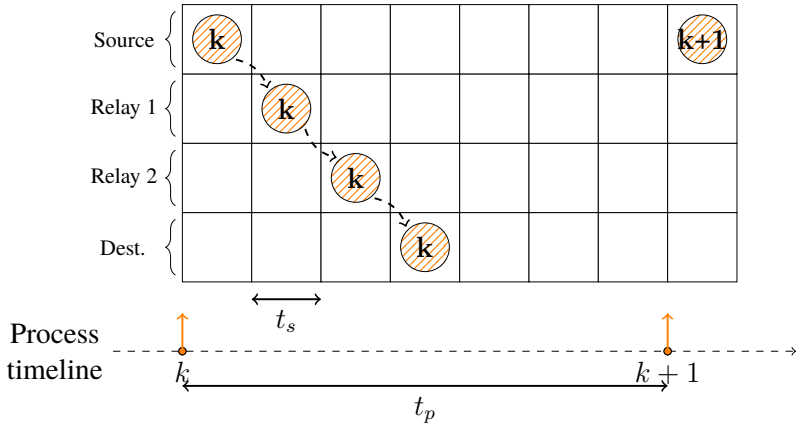


Figure 3.2: Example 3-hop scenario, i.e., $N = 3$ with a sampling period of 7 slots, i.e., $m = 7$. Together with dashed lines, the orange circle illustrates the path of the k-th update. The following status update, $k+1$, is available seven slots after the previous sampling event k. The empty transmission slots indicate that the slots are either idle or allocated to other applications.

Age model: In chapter 2, we introduced the discrete-time model for control systems, which alters its state only in those instances when there is a new sampling event. In other words, the system state remains constant throughout the entire sampling period. Motivated by [Maa+20], which assumes aging only when the system status is changed, we model the time dynamics of AoI as a discrete-time process, being synchronous to the source's sampling

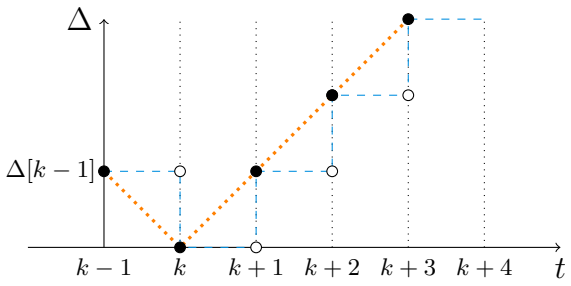


Figure 3.3: An example evolution of the AoI at the receiver. A new status update is received in sampling period k. During the following 3 sampling periods, the monitor fails to update its most recent information.

events. To that end, we evaluate the age at the monitor only once in each sampling period and right before the following sampling event[1]. In doing so, we allow the age to be zero if the most recent system state is available at the destination prior to a sampling event. To illustrate this effect, we present Fig. 3.3 showing an example evolution of the discrete-time AoI process. Despite of the "staircase" shape in network's time resolution, the age increases linearly at every sampling event from the monitor's perspective, i.e., application layer. This differentiation is emphasized in the figure through the blue and orange dashed lines.

For a better understanding of the following analysis, let us give an index to each link over the path starting from one. Given an N-hop line network consisting of N such links and $N+1$ nodes, with node 0 being the source node and node N being the destination, let link n correspond to the first hop between the nodes indexed by 0 and 1. Next, let us introduce an indicator variable $\gamma_n[k] \in \{0,1\}$ for each link $n \in \{1,2,\ldots,N\}$, which indicates the outcome of the transmission in sampling period k over link n . If the transmission fails, γ_n is zero with a constant probability of p_n. Analogously, $\gamma_n[k]$ takes the value of one if the transmission on the n-th link is successful, i.e., $\Pr[\gamma_n[k]=1] = 1-p_n, \forall k$.

Furthermore, given $\Delta_0[k] = 0, \forall[k]$, let $\Delta_n[k] \in \mathbb{N}_0$ represent the AoI at node n after the activation of the n-th link within the k-th sampling period. Hence, it follows that the discrete-time model of the age is characterized by:

$$\Delta_n[k] = \begin{cases} \Delta_{n-1}[k] & , \text{ if } \gamma_n[k]=1, \\ \Delta_n[k-1]+1 & , \text{ otherwise.} \end{cases} \tag{3.1}$$

The reason for Δ_0 to be zero at all times is the source node having access to the latest system state following a sampling event. Let us remind that throughout the network, an older status update packet is replaced if a new piece of information arrives. It follows from the model that $\Delta_N[k]$ is the age at the monitor at the end of the k-th sampling period.

[1]This is similar to the approach in [Kad+16], which allows a reduction of AoI only at the end of a sampling period.

As depicted in Fig. 3.2, our scenario allows each hop to have a single slot between two consecutive sampling events. In addition, the successful forwarding of a new information is represented by a single probability. However, our model can be generalized to multiple slots case under the condition that each slot allocated to a link n occurs before the subsequent link $n+1$ and multiple copies of the same packet is sent using different resources to increase redundancy. In such a setting, the definition of a loss event depends on all transmissions being unsuccessful. Equivalently, if we denote the loss probability of a single transmission by p_n^* and there are C copies of the same packet being sent, then the next node fails to retrieve the new status update only if all of these transmissions fail, i.e., $p_n = (p_n^*)^C$. Hence, the AoI model from (3.1) remains valid for the combined loss probability p_n if the sampling period is long enough to accommodate all transmissions along the path.

3.1.2 Probability Analysis of AoI:

This section presents a mathematical discussion on AoI distribution and provides a recursive algorithm to calculate the occurrence probability of an age value at each network node. The results presented in the following text are based on [Aya+20b]. Let us first start with the single-hop case and build on it in an incremental fashion.

The single-hop case: Consider a source-destination pair connected via a wireless link having a constant packet loss probability $p_1 \in (0,1)$. Following our age model from (3.1), Δ_0 is zero at all times, since the packet at the source is always fresh. This implies that a successful update of the monitor is followed by a reset of AoI to zero. In other words, if $\gamma_1[k] = 1$, then $\Delta_1[k] = \Delta_0[k] = 0$. On the other hand, the possibility of having an age value of δ_1 is possible if and only if δ_1 consecutive failed transmissions follow a successful transmission. Thus, its occurrence probability can be obtained from:

$$\Pr\left[\Delta_1[k] = \delta_1\right] = (1-p_1)\cdot p_1^{\delta_1}, \quad \forall k. \tag{3.2}$$

Consequently, we can write the expected AoI at the monitor as a sum of all possible age values weighted by their occurrence probability as:

$$\mathbb{E}[\Delta_1] = \sum_{\delta_1=0}^{\infty} \Pr\left[\Delta_1[k] = \delta_1\right]\cdot\delta_1 = \frac{p_1}{1-p_1}. \tag{3.3}$$

The two-hop case: Before extending our results to N-hop, let us briefly elaborate on the two-hop scenario consisting of a source, an intermediate (relay), and a destination (monitor) node, as previously depicted in Fig. 3.1. The first link connecting the source and the relay node has a constant loss probability of p_1. Similarly, the second link between the relay-destination pair is subject to packet loss with time-invariant probability of p_2. Note that the occurrence

of a loss event on either links is independent from each other. Therefore, we can analyze each link independently.

The age model at the relay node follows the previously derived equations (3.2) and (3.3). However, the information that is being forwarded over the second link may not be from the current sampling period. To explain this phenomenon in a toy example, suppose that a status update packet from time k is forwarded to the relay node but the second transmission is not successful. Hence, the monitor cannot be updated within the same sampling period, i.e., $\Delta_2[k] = \Delta_2[k-1] + 1$. In the next sampling period, i.e., $k+1$, if the second link gets activated without the first link being active, i.e., $\gamma_1[k+1] = 0$ and $\gamma_2[k+1] = 1$, then the monitor is provided with an information that is one sampling period old. In this case, the AoI at the destination drops to one instead of zero.

Having shown that the newly received information at the monitor may be outdated due to "in-network aging", let us proceed with the closed form equation for a two-hop network. First, let us denote the latest time instance, at which the monitor has been updated, by k_2[2]. At this point, the age at the relay and monitor nodes equalize, in accordance with the success case from (3.1). Hence, we can write $\Delta_1[k_2] = \delta_1$. In consequence, we can treat the AoI at the monitor, i.e., $\Delta_2[k]$, as an independent aging process that depends on the transmission outcome of the second link. Thereby, we can formulate the probability of having an age δ_2 at the monitor as:

$$\Pr\left[\Delta_2[k] = \delta_2 \;\middle|\; \Delta_1[k_2] = \delta_1\right] = \begin{cases} 0 & \text{, if } \delta_2 < \delta_1, \\ (1-p_2)\cdot p_2^{\delta_2-\delta_1} & \text{, if } \delta_2 \geq \delta_1. \end{cases} \tag{3.4}$$

The equation above states that given $\Delta_1[k_2] = \delta_1$, the age at the destination cannot be lower than the age at the relay node, as the correspond packet must have passed through it in the past[3]. Additionally, due to the age increment by one after every sampling period, the model dictates $\delta_2 - \delta_1$ consecutive failures on the second link following the initial successful transmission between the two nodes. In the equation above, δ_1 represents the first link's contribution to the age at the monitor. Additionally, the remaining share $\delta_2 - \delta_1$ of the current age $\Delta_2[k] = \delta_2$ is introduced through the packet loss events on the second link. Alternatively, $\delta_2 - \delta_1$ can be interpreted as the second link's contribution to the current age.

Next, we focus solely on the non-zero addends from (3.4), which allows us to apply the law of total probability. Doing so, we obtain the marginal probability as:

$$\Pr\left[\Delta_2[k] = \delta_2\right] = \sum_{\delta_1=0}^{\delta_2} \Pr\left[\Delta_2[k] = \delta_2 \;\middle|\; \Delta_1[k_2] = \delta_1\right] \cdot \Pr\left[\Delta_1[k_2] = \delta_1\right]. \tag{3.5}$$

[2]Imagine k_2 as a particular sampling period in the past, i.e., $k_2 < k$.

[3]This is results from the line network assumption. It does not hold for general multi-hop networks.

After plugging (3.2) and (3.4) in (3.5) one can derive the following closed-form expression as a function of the known parameters p_1 and p_2:

$$\begin{aligned}\Pr[\Delta_2[k]=\delta_2] =& \sum_{\delta_1=0}^{\delta_2}(1-p_1)\cdot {p_1}^{\delta_1}\cdot(1-p_2)\cdot {p_2}^{\delta_2-\delta_1}\\ =&(1-p_1)(1-p_2){p_2}^{\delta_2}\sum_{\delta_1=0}^{\delta_2}\left(\frac{p_1}{p_2}\right)^{\delta_1}\\ =&(1-p_1)(1-p_2){p_2}^{\delta_2}\cdot\frac{1-\left(\frac{p_1}{p_2}\right)^{\delta_2+1}}{1-\frac{p_1}{p_2}}\\ =&(1-p_1)(1-p_2)\cdot\frac{{p_2}^{\delta_2+1}-{p_1}^{\delta_2+1}}{p_2-p_1}.\end{aligned} \tag{3.6}$$

As the last row of the equation is undefined for the equality case $p_1 = p_2$, due to a zero denominator, one can simply use the first row of (3.6).

The expected AoI at the destination can be derived following the same steps as it has been done for (3.3). Hence, expected age at the monitor is given as:

$$\mathbb{E}\left[\Delta_2\right]=\frac{p_1}{1-p_1}+\frac{p_2}{1-p_2}. \tag{3.7}$$

The n-hop case: It is evident from (3.7) that the expected age at any node n, indexed according to its appearance over the path, can be calculated by:

$$\mathbb{E}\left[\Delta_n\right]=\sum_{i=1}^{n}\frac{p_i}{1-p_i}. \tag{3.8}$$

By applying the law of total probability once again, we provide the following probability for hop n having the age δ_n as:

$$\Pr\left[\Delta_n[k]=\delta_n\right]=\sum_{\delta_{n-1}=0}^{\delta_n}\Pr\left[\Delta_n[k]=\delta_n \mid \Delta_{n-1}[k_n]=\delta_{n-1}\right]\cdot\Pr\left[\Delta_{n-1}[k_n]=\delta_{n-1}\right]. \tag{3.9}$$

For instance, the closed-form expression for the three-hop case can be obtained from (3.9) by plugging in our results from (3.6):

$$
\begin{aligned}
\Pr[\Delta_3[k]=\delta_3] &= \sum_{\delta_2=0}^{\delta_3}(1-p_3){p_3}^{\delta_3-\delta_2}\cdot(1-p_1)(1-p_2)\cdot\frac{{p_2}^{\delta_2+1}-{p_1}^{\delta_2+1}}{p_2-p_1}\\
&= \frac{{p_3}^{\delta_3+1}\prod_{i=1}^{3}(1-p_i)}{p_2-p_1}\cdot\sum_{\delta_2=0}^{\delta_3}\left(\left(\frac{p_2}{p_3}\right)^{\delta_2+1}-\left(\frac{p_1}{p_3}\right)^{\delta_2+1}\right)\\
&= \frac{{p_3}^{\delta_3+1}\prod_{i=1}^{3}(1-p_i)}{p_2-p_1}\cdot\left(\left(\frac{p_2}{p_3}\right)\frac{1-\left(\frac{p_2}{p_3}\right)^{\delta_3+1}}{1-\frac{p_2}{p_3}}-\left(\frac{p_1}{p_3}\right)\frac{1-\left(\frac{p_1}{p_3}\right)^{\delta_3+1}}{1-\frac{p_1}{p_3}}\right)\\
&= \frac{\prod_{i=1}^{3}(1-p_i)}{p_2-p_1}\cdot\left(p_2\cdot\frac{{p_3}^{\delta_3+1}-{p_2}^{\delta_3+1}}{p_3-p_2}-p_1\cdot\frac{{p_3}^{\delta_3+1}-{p_1}^{\delta_3+1}}{p_3-p_1}\right)\\
&= \frac{\prod_{i=1}^{3}(1-p_i)}{p_2-p_1}\cdot\sum_{j=1}^{2}(-1)^j\cdot p_j\cdot\frac{{p_3}^{\delta_3+1}-{p_j}^{\delta_3+1}}{p_3-p_j}
\end{aligned}
\tag{3.10}
$$

Lastly, we present Algorithm 1, a pseudo-code for recursive probability calculation for any n hop. The algorithm has a time complexity of $\mathcal{O}((\delta_n+1)\cdot n)$.

Algorithm 1 Recursive age function: $f(\delta_n, n, \mathbf{p}^n) = o$

Input: δ_n age, n number of hops, $\mathbf{p}^n$ vector of loss probabilities for n hops
Output: o the probability of age δ_n with n hops

```
Initialize: o ← 0
if n = 1 then
    return (1 − p_n) · p_n^{δ_n}
else
    for δ_{n−1} ∈ [0 δ_n] do
        o ← o + ((1 − p_n) · p_n^{δ_n − δ_{n−1}}) · f(δ_{n−1}, n − 1, p^{n−1})
    end for
    return o
end if
```

3.1.3 Numerical Results

In order to verify our analytical results, we conduct a simulative study using a three-hop network. To that end, we simulate a line network consisting of a source, two relay nodes and a destination. Each simulation run consists of $T = 100\ 000$ sampling periods. Each scenario, characterized by a triplet of the packet loss probabilities on each link, is repeated 100 times. The notation for the scenario is in the following form (p_1, p_2, p_3), whereas p_1, p_2, and p_3 are the loss probabilities on the source-to-relay, relay-to-relay, relay-to-destination links, respectively.

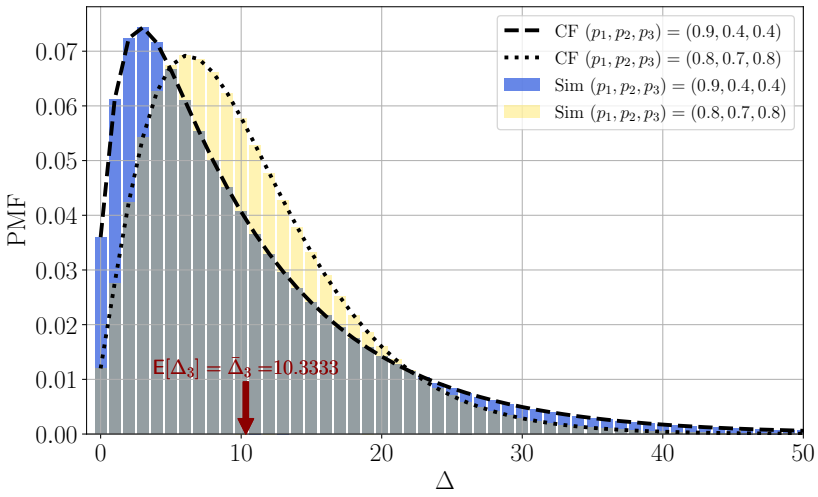


Figure 3.4: AoI probability mass function of two combinations of three hop loss probabilities p_l with average as well as the expected AoI $\mathbb{E}[\Delta_3]$ = 10.3333, rounded to 4 decimal places. Higher loss probability p_1 = 0.9 in the first hop increases the distribution tail compared to moderate loss probabilities on all three links.

In compliance with the age model from the equation (3.1), we evaluate the AoI at the end of each sampling period and the long-term average AoI at the destination is calculated as:

$$\bar{\Delta}_3 \triangleq \frac{1}{T}\sum_{k=0}^{T-1}\Delta_3[k]. \tag{3.11}$$

Note that the equation above is the discrete-time version of the continuous-time mean AoI, given in (2.2). Furthermore, the average peak age, $\hat{\Delta}$, is measured as in (2.3), taking solely the values prior to a successful reception at the monitor into account.

We select the following two scenarios:

- $\mathbf{S_1} = (p_1 = 0.9, p_2 = 0.4, p_3 = 0.4)$,
- $\mathbf{S_2} = (p_1 = 0.8, p_2 = 0.7, p_3 = 0.8)$.

having the equal expected AoI at the monitor rounded to four decimal places, i.e., $\mathbb{E}[\Delta_3] = 10.3333$. $\mathbf{S_1}$ represents a scenario that contains extremely bad channel condition on the first hop but two further links with relatively better conditions. $\mathbf{S_2}$ depicts a scenario in which all three links are subject to moderately high packet loss. Fig. 3.4 shows the AoI PMF for $\mathbf{S_1}$ and $\mathbf{S_2}$. It contains both the analytical and simulation results. While the analytical value is obtained from (3.10), the value corresponding to simulations, labeled as "Sim", is the normalized occurrence frequency of an age value throughout the runs. One can clearly see the analytical and simulation results overlapping. Moreover, we can observe that the tail of the distribution is longer in $\mathbf{S_1}$ than in $\mathbf{S_2}$.

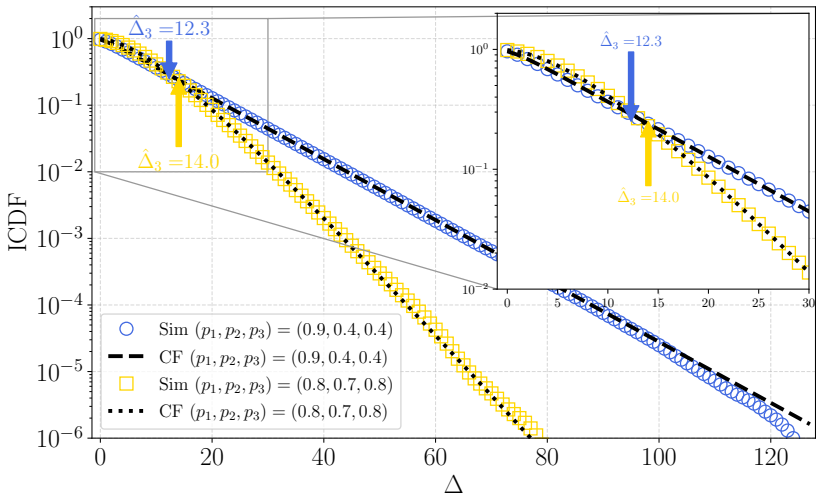


Figure 3.5: Inverse cumulative distribution function for two combinations of three hop loss probabilities p_l with expected age $\mathbb{E}[\Delta_3] = 10.3333$. Higher loss probabilities increase the AoI for higher reliability targets. Moreover, for higher reliability targets, the average peak AoI varies drastically compared to the actual AoI.

Next, we present in Fig. 3.5 the Inverse Cumulative Distribution Function (ICDF), as well as the average peak AoI performance to draw further conclusions regarding the reliability guarantees. In spite of the equality between their expected mean AoI performances, first aspect to notice is the difference between their average peak age performance, marked by arrows in the figure. While $\mathbf{S_1}$ results in $\hat{\Delta}_3 = 12.3$, an average of $\hat{\Delta}_3 = 14.0$ is achieved in the case of $\mathbf{S_2}$. Having that said, we can see from the figure that $\mathbf{S_1}$ and $\mathbf{S_2}$ pose significant differences beyond 10^{-1} when it comes to providing maximum age guarantees. Particularly, if we are dealing with applications that require high reliability in terms of the maximum achieved AoI, e.g., three nines, or equivalently 99.9%, both scenarios differ around a maximum age of 20 levels. However, surprisingly, $\mathbf{S_2}$ is a more desirable setting for limiting the maximum age with a particular probability than $\mathbf{S_1}$, although the peak age performance is worse than $\mathbf{S_1}$. Thereby, we can conclude that neither the average nor the peak AoI is a sufficient standalone indicator if we aim to offer reliability guarantees for real-time applications. Instead, the entire probability distribution should be taken into account.

3.2 The Multi-User Single-Hop Case

In this section, we consider a single-hop wireless network with multiple control applications. As one of the key contributions of this section, one can name the utilization of AoI distribution for control performance maximization. More importantly, this is the first section in this thesis

deriving the relationship between information freshness and estimation performance. This section is based on [AEK21].

3.2.1 System Model

We consider a wireless communication network comprised of N independent LTI control sub-systems. Each sub-system i, the components of which are a plant, a sensor, and a controller, behaves according to the following difference equation in discrete-time:

$$\boldsymbol{x}_i[t+1] = \boldsymbol{A}_i\boldsymbol{x}_i[t] + \boldsymbol{B}_i\boldsymbol{u}_i[t] + \boldsymbol{w}_i[t]. \tag{3.12}$$

The variables are defined as in (2.14) with sub-script i being used for sub-system indexing. Time is assumed to be slotted, while the slot duration is normalized to unity. In contrast to the previous section, throughout the following analysis, each control sub-system is sampled with the slot frequency. That is, when the network's time advances from t to $t+1$, the i-th sub-system's state changes from $\boldsymbol{x}_i[t]$ to $\boldsymbol{x}_i[t+1]$.

The system state measured by each sensor is transmitted over a shared wireless communication link to the controller. Each transmission between the i-th sensor-controller pair starts and ends within the same slot. Moreover, the shared channel is subject to packet losses that varies among sub-systems but is constant over time. In other words, each status update packet is received by the controller with a success probability of μ_i, hence modeled as a packet erasure channel.

The success probability μ_i is strictly related to the resource allocation, which is done centrally prior to deployment. In fact, one can imagine it as follows: A limited set of resources is distributed among multiple users by a central entity. Each system is assigned to a set of resources, e.g., multiple channels, on which they broadcast the copies of a status update packet simultaneously. For this information to be received by the controller, it is enough to receive at least one of those copies, which occurs with the probability μ_i. Hence, μ_i defines the aggregated success probability of the sensor-to-control link. To that end, we approximate the behavior of such a network with scarce resources through the following equation:

$$\begin{aligned} &\sum_{i=1}^{N} \mu_i \leq R, \\ &0 \leq \mu_i \leq 1, \quad \forall i. \end{aligned} \tag{3.13}$$

$R \in \mathbb{R}$ denotes the total network capacity and is a scalar. Simply put, the equation states that once the resource allocation is done, the sum of all sub-systems' success probabilities is upper-bounded by R. Note that we assume μ_i to be linear and continuous in $[0, 1]$. Such a model is

a rough approximation of resource-limited networks, typically abstracted as a maximum rate. However, it captures the key characteristics of a centralized resource allocation problem in networks with resource constraints.

In our considered scenario, we assume that sensors are incapable of replacing an outdated packet with a more recent one, e.g., due to constrained computational capabilities, limited flexibility of the communication stack, etc.. Instead, we assume that they the status update packets must traverse a FCFS queue of infinite capacity. However, the sensors are able to control the packet injection rate by choosing an admission probability $\lambda_i \in [0, 1]$. In consequence, any sampled information $\boldsymbol{x}_i[t]$ is injected into the packet queue with a constant probability of λ_i. Equivalently, any state measurement is discarded with $1 - \lambda_i$ and excluded from further consideration. Lastly, each packet, that is admitted into the packet queue, is retransmitted until at least on of its copies is received[4]. The resulting behavior of the transmission queue can be modeled as a Geo/Geo/1 discrete-time queue with service rate μ_i.

The controller is a certainty equivalence controller with the control law:

$$\boldsymbol{u}_i[t] = -\boldsymbol{L}_i \hat{\boldsymbol{x}}_i[t]. \tag{3.14}$$

The estimated state $\hat{\boldsymbol{x}}_i[k]$ is defined as in (2.20). It is important to remind that $\hat{\boldsymbol{x}}_i[k]$ is the current state that the controller expects the controlled process to have by looking at the most recent information $\boldsymbol{x}_i[\nu_i[t]]$. Now, let us substitute (3.12) into (2.20) and formulate $\hat{\boldsymbol{x}}_i[k]$ as a function of the known variables:

$$\begin{aligned}
\hat{\boldsymbol{x}}_i[t] &\overset{(2.20)}{=} \mathbb{E}\left[\boldsymbol{A}_i \boldsymbol{x}_i[t-1] + \boldsymbol{B}_i \boldsymbol{u}_i[t-1] + \boldsymbol{w}_i[t-1] \mid \boldsymbol{x}_i[\nu_i(t)]\right] \\
&\overset{(2.20)}{=} \mathbb{E}\Big[\boldsymbol{A}_i \left(\boldsymbol{A}_i \boldsymbol{x}_i[t-2] + \boldsymbol{B}_i \boldsymbol{u}_i[t-2] + \boldsymbol{w}_i[t-2]\right) + \boldsymbol{B}_i \boldsymbol{u}_i[t-1] + \boldsymbol{w}_i[t-1] \mid \boldsymbol{x}_i[\nu_i(t)]\Big] \\
&\overset{(2.20)}{=} \mathbb{E}\left[\boldsymbol{A}_i^{t-\nu_i(t)} \boldsymbol{x}_i[\nu_i(t)] + \sum_{q=1}^{t-\nu_i(t)} \boldsymbol{A}_i^{q-1} \boldsymbol{B}_i \boldsymbol{u}_i[t-q] + \sum_{q=1}^{t-\nu_i(t)} \boldsymbol{A}_i^{q-1} \boldsymbol{w}_i[t-q] \mid \boldsymbol{x}_i[\nu_i(t)]\right] \\
&= \boldsymbol{A}_i^{t-\nu_i(t)} \boldsymbol{x}_i[\nu_i(t)] + \sum_{q=1}^{t-\nu_i(t)} \boldsymbol{A}_i^{q-1} \boldsymbol{B}_i \boldsymbol{u}_i[t-q],
\end{aligned} \tag{3.15}$$

where $\boldsymbol{A}^p$ denotes the p-th power of the matrix $\boldsymbol{A}$. Note that the last line consists only of known vectors and does not contain any random variable. On the contrary, the third line is a function of normal random vectors $\boldsymbol{w}_i$. However, as the normality is preserved by linear transformations, each addend of the form $\boldsymbol{A}_i \boldsymbol{w}_i[t]$ is also normal with zero mean [BT08]. Thus, we were able to get rid of them in the final equation.

How one should interpret the final equation is as follows: When the i-th controller receives a new status update packet, carrying the state information $\boldsymbol{x}_i[t - \nu_i(t)]$ generated at time $\nu_i(t)$,

[4]The feedback about the transmission outcome is provided by an instant ACK/NACK mechanism

it calculates (3.15) and thereby obtains the state estimate remotely. This also implies that the time-invariant system parameters such as $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ as well as the control input history is available at the controller. As the controller is anyways the entity deciding the control inputs, this does not impose additional communication or simplifications in the system model.

3.2.1.1 Estimation Error and Age of Information

Next, we proceed with the network-induced error from (2.21) that is defined as the difference between the actual and estimated system states. We formulate the Mean Squared (estimation) Error MSE as a function of the system parameters:

$$\begin{aligned}
&\mathbb{E}\left[\|\boldsymbol{e}_i[t]\|^2 \mid \boldsymbol{x}_i[\nu_i[t]]\right] \overset{(2.21)}{=} \mathbb{E}\left[\left(\boldsymbol{x}_i[t]-\hat{\boldsymbol{x}}_i[t]\right)^T\left(\boldsymbol{x}_i[t]-\hat{\boldsymbol{x}}_i[t]\right) \mid \boldsymbol{x}_i[\nu_i[t]]\right] \\
&\overset{(3.12),(3.15)}{=} \mathbb{E}\left[\left(\sum_{q=1}^{t-\nu_i(t)} \boldsymbol{A}_i^{q-1}\boldsymbol{w}_i[t-q]\right)^T\left(\sum_{q=1}^{t-\nu_i(t)} \boldsymbol{A}_i^{q-1}\boldsymbol{w}_i[t-q]\right)\right] \\
&= \mathbb{E}\left[\sum_{q=1}^{t-\nu_i(t)} \left(\boldsymbol{w}_i[t-q]\right)^T\left(\boldsymbol{A}_i^{q-1}\right)^T \sum_{q=1}^{t-\nu_i(t)} \boldsymbol{A}_i^{q-1}\boldsymbol{w}_i[t-q]\right] \\
&\overset{\text{A}}{=} \mathbb{E}\left[\sum_{q=1}^{t-\nu_i(t)} \left(\boldsymbol{w}_i[t-q]\right)^T\left(\boldsymbol{A}_i^{q-1}\right)^T \boldsymbol{A}_i^{q-1}\boldsymbol{w}_i[t-q]\right] \\
&\overset{\text{B}}{=} \sum_{q=1}^{t-\nu_i(t)} \mathsf{tr}\left((\boldsymbol{A}_i^{q-1})^T\boldsymbol{A}_i^{q-1}\boldsymbol{\Sigma}_i\right) \qquad (3.16a) \\
&= \sum_{q=0}^{t-\nu_i(t)-1} \mathsf{tr}\left((\boldsymbol{A}_i^{q})^T\boldsymbol{A}_i^{q}\boldsymbol{\Sigma}_i\right). \qquad (3.16b)
\end{aligned}$$

A: Noise vectors are i.i.d. thus uncorrelated.
B: Expectation of a random vector's, i.e., $\boldsymbol{v}$, quadratic norm with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{v}}$ is $\mathbb{E}[\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v}] = (\mathbb{E}[\boldsymbol{v}])^T\boldsymbol{M}\mathbb{E}[\boldsymbol{v}] + \mathsf{tr}(\boldsymbol{M}\boldsymbol{\Sigma}_{\boldsymbol{v}})$. In the equation above and in the remainder of this thesis, tr denotes the trace operator. $\boldsymbol{M}^T$ is the transpose of a matrix $\boldsymbol{M}$.

The equations (3.16a) and (3.16b) are equivalent except for the shifted summation range by one. We provide both versions for convenience. Having that said, these equations describe how one can calculate the MSE as a function of the time-invariant system parameters $\boldsymbol{A}_i$, $\boldsymbol{\Sigma}_{\boldsymbol{i}}$ and the difference between the current time and the generation time of the most recent information available. Note that the latter, i.e., $t-\nu_i(t)$, defines the number of addends. It is important to mention that (3.16b) holds for the equality case as well, since if $t=\nu_i(t)$ it returns zero. This would correspond to a situation, in which the i-th controller has the perfect knowledge of the actual state.

Conforming with (2.1), we can replace the term $t - \nu_i(t)$ with age as:

$$g(\Delta_i[t];t) \triangleq \mathbb{E}\left[\|\boldsymbol{e}_i[t]\|^2\right] = \sum_{q=0}^{\Delta_i[t]-1} \mathsf{tr}\left(\left(\boldsymbol{A}_i^T\right)^q \boldsymbol{A}_i^q \boldsymbol{\Sigma}_i\right). \tag{3.17}$$

$g : \mathbb{N}_0 \mapsto \mathbb{R}$ can be seen as a age-penalty function mapping the AoI of a sub-system i to the MSE. The RHS of the equation (3.17) is strictly increasing in age for any invertible $\boldsymbol{A}_i$ and positive-definite noise covariance matrix $\boldsymbol{\Sigma}_i$[5].

3.2.1.2 Stationary Distribution of AoI

If a discrete-time Geo/Geo/1 queue models the communication between a source-destination pair, the AoI follows a stationary distribution [Kos+20]. In their work, authors derive the PMF of the AoI for given arrival and service rates λ_i and μ_i as[6]:

$$\Pr[\Delta_i = \delta] = \frac{(\mu_i - \lambda_i)\left(\frac{1-\lambda_i}{1-\mu_i}\right)^{-\delta}}{1-\mu_i} - \lambda_i\mu_i\delta(1-\mu_i)^{\delta-1} + \frac{\lambda_i\mu_i(1-\lambda_i)^{\delta}}{\mu_i - \lambda_i} + \frac{\left(\lambda_i^2 - \lambda_i\mu_i(\mu_i+1) + \mu_i^2\right)(1-\mu_i)^{\delta-1}}{\lambda_i - \mu_i}. \tag{3.18}$$

Note that the stability of the queue is a requirement for the equation above, hence $\frac{\lambda_i}{\mu_i} = \rho_i < 1, \forall i$ must hold. Equivalently, the aggregated success probability μ_i of each sub-system i has to be higher than the packet admission probability λ_i.

(3.18) gives the probability of having $\Delta_i = \delta$ at the destination, i.e., controller. Thereby, we can calculate the expected long-term average of the MSE as a function of the network parameters λ_i and μ_i:

$$\begin{aligned} C_i(\lambda_i,\mu_i) &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{e}_i^T[t]\boldsymbol{e}_i[t] \\ &= \sum_{\delta=1}^{\infty} \Pr[\Delta_i = \delta] \cdot g(\delta) \\ &= \sum_{\delta=1}^{\infty} \Pr[\Delta_i = \delta] \cdot \sum_{p=0}^{\delta-1} \mathsf{tr}\left((\boldsymbol{A}_i^T)^p \boldsymbol{A}_i^p \boldsymbol{\Sigma}_i\right) \end{aligned} \tag{3.19}$$

The equation can be interpreted as the weighted average of the age-penalties $g(\delta)$, with weights given by the probabilities $\Pr[\Delta_i = \delta]$. We denote the long-term average MSE as $C_i(\lambda_i,\mu_i)$ throughout the following analysis.

[5] As we consider (non-zero) Gaussian noise in our model $\boldsymbol{\Sigma}_i$ is positive-definite by definition.

[6] Note that (3.18) is a shifted version of the original equation from [Kos+20]. The necessity for adaptation is that our model allows a minimum age of one while in [Kos+20], the minimum achievable AoI is two.

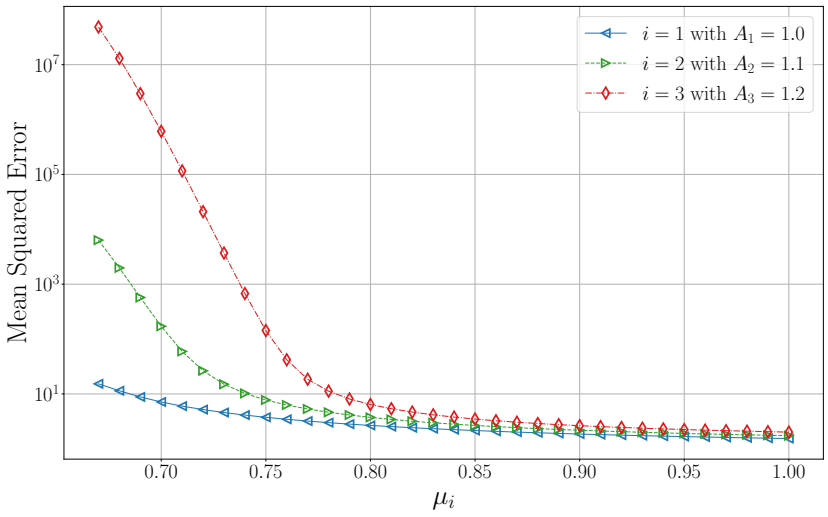


Figure 3.6: The expected long-term average of MSE, i.e., C_i, for varying service rate while the arrival rate is fixed for all sub-systems, i.e., $\lambda_i = 0.65, \forall i$. Each curve corresponds to a different sub-system with the system matrices $A_{\{1,2,3\}} \in \{1.0, 1.1, 1.2\}$. A higher A_i represents a less stable system that is more challenging to control. The noise covariance matrix is one for all sub-systems, i.e., $\mathbf{\Sigma}_i = 1.0, \forall i$.

Let us look at the behavior of C_i for varying service rate, i.e., $\{\mu_i : \lambda_i < \mu_i \leq 1\}$ and fixed $\lambda_i = 0.65$ presented in Fig. 3.6. Each of the three curves, depicted in different colors, belong to a sub-system, with the system matrices being varied between 1.0, 1.1, and 1.2. A higher $\boldsymbol{A}_i$ represents a more challenging control application due to its relatively higher instability[7]. It is evident that the system dynamics play an essential role, especially when the channel conditions are worsened. This is related to the PMF "shifting" more towards larger age values, which in return leads to the amplification of the higher age-penalties. Governed by the aggregated success rate of individual loops, an increase in the service rate predictably leads to an improved estimation performance.

Fig. 3.7, on the other hands plots the MSE when the service rate is fixed. Instead, the arrival rate is varied such that $\{\lambda_i : 0 < \lambda_i < \mu_i\}$. We observe that a very low λ_i leads to worsened estimation performance due to the increased inter-arrival times. In consequence, the monitor cannot be updated regularly due to the under-utilization of the network resources. Contrarily, throttling the admission probability is counterproductive w.r.t. the MSE after a certain λ_i value. As a consequence of the increased arrival rate, the packets start experiencing longer queueing times and thus become outdated when they are eventually received by the controller.

[7]As the eigenvalue of a scalar system matrix $\boldsymbol{A}_i$ equals to the matrix itself. From the fundamentals of control theory, we know that the higher the eigenvalue gets beyond one, it represents a less stable system.

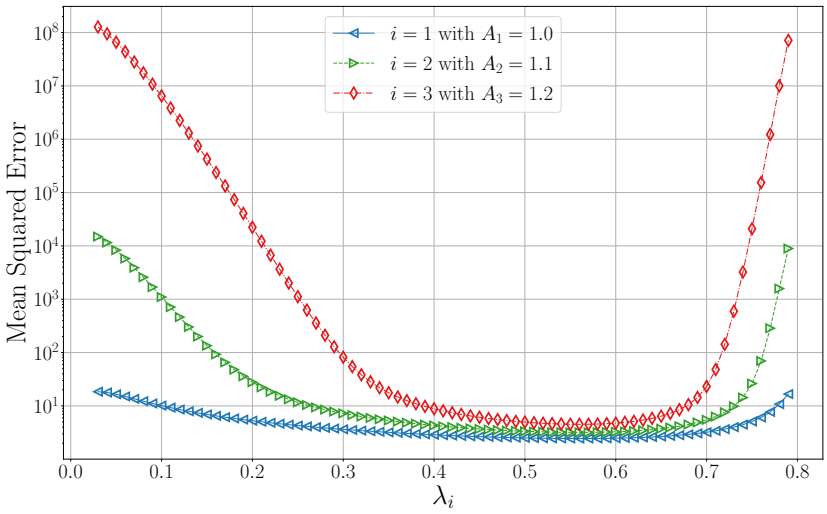


Figure 3.7: The expected long-term average of MSE, i.e., C_i, for a fixed service rate $\mu_i = 0.8$ and noise covariance matrix $\Sigma_i = 1.0, \forall i$. Each curve corresponds to a different sub-system with the system matrices $A_{\{1,2,3\}} \in \{1.0, 1.1, 1.2\}$.

3.2.1.3 Optimization Problem

Now, we consider the total service rate R to be limited. We formulate an MSE minimization problem for given arrival rates in the form of $\boldsymbol{\lambda} = [\lambda_1 \lambda_2 \dots \lambda_N]^T$. The optimization variable is $\boldsymbol{\mu} = [\mu_1 \mu_2 \dots \mu_N]^T$ and the goal is to find a vector $\boldsymbol{\mu}^*$ that leads to the minimum age-penalty in the network:

$$\begin{aligned} \boldsymbol{\mu}^* = \arg\min_{\boldsymbol{\mu}} \quad & \sum_{i=1}^{N} C_i(\lambda_i, \mu_i) \\ \text{s.t.} \quad & \lambda_i - \mu_i < 0, \forall i \\ & \sum_{i=1}^{N} \mu_i \leq R. \\ & 0 \leq \mu_i \leq 1, \forall i \end{aligned} \tag{3.20}$$

In order to make the optimization problem tractable, we assume that the set of feasible arrival and service rates are convex and the domain of the problem is $[0, 1]$.

3.2.1.4 Results and Evaluation

Our goal is solve the optimization problem (3.20) and find the optimal rate allocation in the network that leads to the maximized estimation performance. To that end, we consider $N = 3$ scalar control sub-systems and conduct a simulative analysis consisting of $T = 2000$ time slots, repeated 2000 times. In order to cover a more generalized setting, we assume heterogeneous applications whose system matrices are selected as $\boldsymbol{A}_{\{1,2,3\}} = \{1.0, 1.1, 1.2\}$. Each of these represents a feedback control loop with a heterogeneous time-criticality. Moreover, the

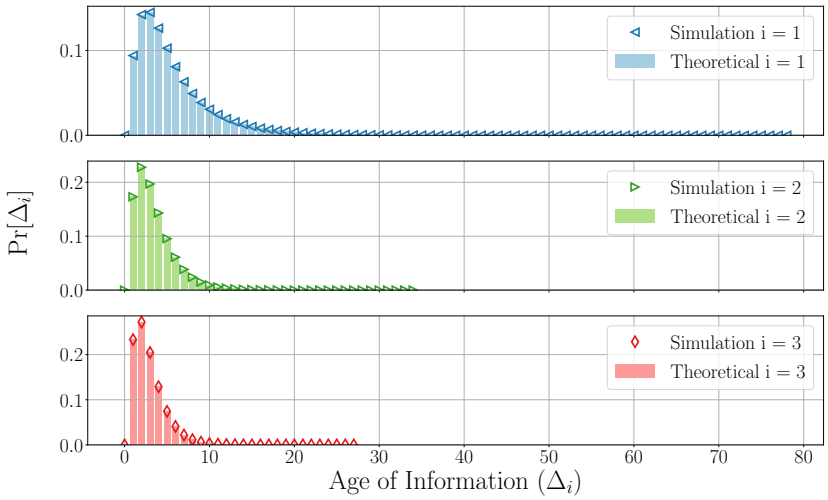


Figure 3.8: The PMF of AoI illustrated together with the normalized occurrence frequency of Δ_i throughout our simulations. Transmission success probabilities are given as $\boldsymbol{\mu} = [0.594, 0.673, 0.733]^T$.

remaining control system parameters are assumed to be identical. That is, $\boldsymbol{\Sigma}_i = 1.0$ and $\boldsymbol{\Sigma}_i = 1.0, \forall i$. For simplicity, we select $\boldsymbol{Q}_i = 1.0$ and $\boldsymbol{R}_i = 0$ for all sub-systems. In other words, we penalize solely the state deviation from zero but neglect the control effort. Hence, the optimal feedback gain matrix from (2.17) simplifies to $\boldsymbol{L}_i = \boldsymbol{A}_i$. Such a control law is also referred as deadbeat control strategy in the literature. Ultimately, we consider equal arrival rate at each sensor, i.e., $\lambda_i = 0.5, \forall i$.

To avoid a trivial solution, we consider a scenario, in which the resources are scarce. Therefore, the total available service rate R has to be less than the number of sub-systems in the network. To that end, we select $R = 2 < N$ and enforce at least one of the sub-systems to have an aggregated success probability less than one, i.e., $\exists i : \mu_i < 1$.

To obtain the solution of (3.20), we use the GEKKO optimization suite based on Python programming language [Bea+18]. As a result, we obtain the following optimal service rates: $\mu_1^* \approx 0.594$, $\mu_2^* \approx 0.673$, and $\mu_3^* \approx 0.733$. Fig. 3.8 shows the resulting stationary age distribution together with the normalized occurrence frequency of each Δ value. While the least critical sub-system, i.e., $i = 1$, reached peak age values of $\Delta_1[t] > 70$ for some t, the least stable loop, i.e., $i = 3$, never exceeded 30 throughout our simulations. The difference is caused by the unbalanced rate allocation between the feedback loops.

Before delving into the main results, let us have a look at the relationship between the MSE, AoI, and the control system dynamics. Fig. 3.9 combines the analytical and numerical results for $g(\Delta_i)$ in a single plot. We can clearly see that as Δ_i increases, the estimation performance of more time-critical applications drop rapidly. Note that the deviation of the

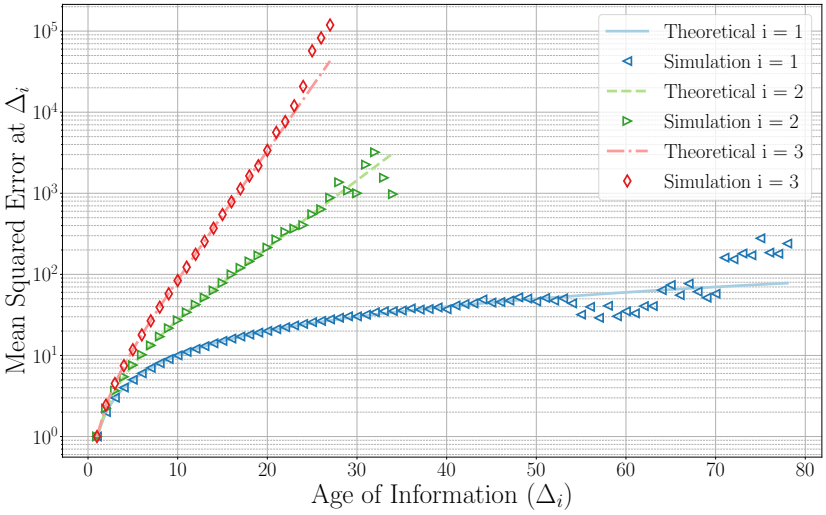


Figure 3.9: The mean squared error plotted against increasing AoI. The figure contains both analytical and numerical results. The results illustrate the growth in MSE for increasing AoI. The oscillations towards higher age values are caused by small sample sizes.

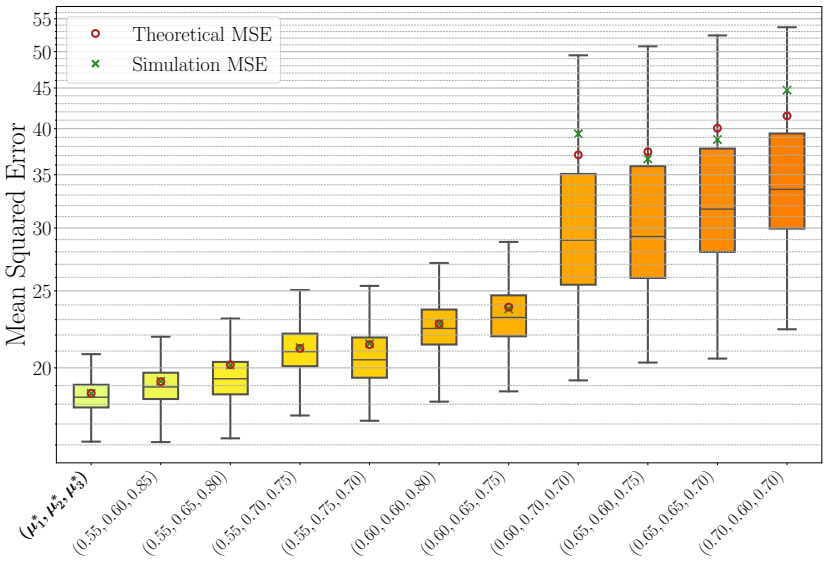


Figure 3.10: The resulting mean squared error illustrated when different allocation vectors $\boldsymbol{\mu} = [\mu_1\ \mu_2\ \mu_3]^T$ along the x-axis are applied. Equal sampling probability of $\lambda_i = 0.5, \forall i$ is selected. Outliers are not displayed to avoid visual clutter. Simulation MSE is calculated by taking the average of all 2000 repetitions. The lower and upper whiskers represent the first and third quartiles, respectively.

numerical (simulation) from analytical (theoretical) results for higher age values is caused by the smallness of their sample size.

Figures 3.10 and 3.11 present the main results of this section. In Fig. 3.10 we compare the estimation performance of $\boldsymbol{\mu}^*$, solving the optimization problem (3.20), to other permutations with three users without violating the constraints. In order to limit the number of permutations, we have selected 0.05 as the resolution of our search space. To that end, we have divided the

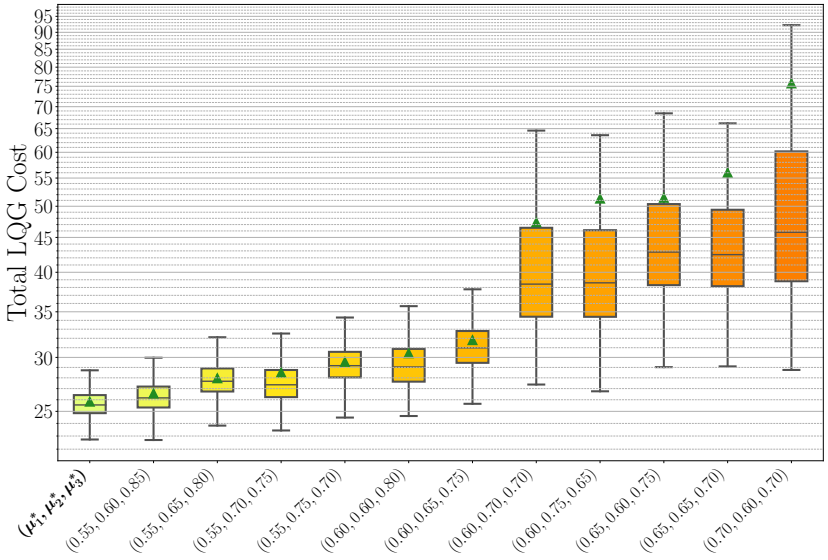


Figure 3.11: The resulting control cost in the network illustrated when different allocation vectors $\boldsymbol{\mu} = [\mu_1\ \mu_2\ \mu_3]T$ along the x-axis are applied. Equal sampling probability of $\lambda_i = 0.5, \forall i$ is selected. Outliers are not displayed to avoid visual clutter. The triangle marker is placed at the overall average cost throughout 2000 repetitions of each scenario. The lower and upper whiskers represent the first and third quartiles, respectively.

linear space between λ_i and one into equally distant sub-spaces with a step-size of 0.05. As we can see from the figure, $\boldsymbol{\mu}*$ outperforms all other rate allocation vectors sorted in ascending order w.r.t. the average MSE[8]. This means, among all explored solutions, the $\boldsymbol{\mu}^*$ is indeed the best-performing one when it comes to maximizing estimation performance.

The MSE allows us to draw conclusions about the estimation performance. On the other hand, the actual QoC is captured by the Linear-Quadratic-Gaussian (LQG) cost given in (2.15). Suppose a human observing the control systems' states. The observer would be able to judge how the control processes are performing only by looking at the state evolution over time but not at the estimation accuracy. Hence, in addition to the MSE, we calculate the time-average LQG cost in the network achieved throughout our simulations, $\mathcal{J}$, as:

$$\mathcal{J} = \sum_{i=1}^{N} \mathcal{J}_i. \tag{3.21}$$

Fig. 3.11 presents the resulting control performance of $\boldsymbol{\mu}^*$ together with other candidates, as in Fig. 3.10. Once again, we can observe that the optimal solution is able to maximize the QoC, although our optimization problem targets the maximization of the estimation accuracy. The indirect improvement of the control performance can be explained through the following intuition: A better decision making at the controller is possible when the destination is

[8]Note that the figure contains only the selected best-performing permutations due to space and presentation considerations.

provided with the information that allows it to estimate the actual state more accurately. This is an intuitive but valuable observation showing how AoI can be utilized as an intermediate tool for the derivation of age-dependent metrics. This close relationship between information freshness, estimation and control reappears in multiple locations in the remainder of this thesis.

3.3 Related Work

Related Work on AoI in Multi-hop Networks

Several studies have addressed the average AoI in multi-hop networks. In particular, [Yat18] and [MAE18] focus on line networks and derive closed-form expressions for average AoI. While [Yat18] assumes LCFS queue, in [MAE18] packets are forwarded in a FCFS fashion to the monitor. Although mostly focusing on the line network case, [TKM17] extends the age problem to a general network setting and show that the general age minimization problem can be solved by converting all flows to single-hop and applying suitable weights on each edge. [FKB19b] considers a multi-source, multi-monitor scenario in a multi-hop setting. The authors derive lower bounds on the instantaneous peak and average AoI by employing fundamental graph-theoretical measures such as connected domination number and average shortest path length. Furthermore, [Chi+21] derives the probability distribution of peak AoI in a tandem system composed by two consecutive FCFS queues. In contrast to the existing works, we extend these by going beyond the mean AoI and deriving the stationary distribution of age in an n-hop line network.

Related Work on AoI in Queueing Systems

The AoI in queueing systems have been the focus of many prior works in the literature. To name a few prominent examples, in [KYG12a], which is one of the first works on AoI, the authors derive the mean AoI for $M/M/1$, $M/D/1$ and $D/M/1$ FCFS queues. [CCE14] shows that discarding outdated packets is significantly more beneficial w.r.t. the average AoI, if the inter-arrival time between two consecutive packets is low. Another work considering $M/M/1$ FCFS queue is [Kos+17]. In their work, authors expand the notion of aging through non-linear functions in time, e.g., exponential and logarithmic functions, to better characterize the importance of a new update for certain applications. [Ino+19] derives the stationary distribution of continuous time AoI under various queueing disciplines such as FCFS, LCFS with or without packet preemption. Last but not least, [Kos+20] derives the stationary distribution of AoI, as well as, peak AoI assuming a discrete-time model. As a main contribution over

the state of the art, we extend the concept of non-linear aging by taking the control system model into account. Additionally, we demonstrate how the stationary distribution of age can be used to maximize the estimation and control performances in a network of multiple control applications of heterogeneous type.

3.4 Summary

In this section, we have examined two distinct scenarios. In the first scenario, we have considered a multi-hop network subject to stationary packet loss on each link. In contrast to the second part of this chapter, we have assumed a LCFS queue of size one at each node, meaning that any older packet in the transmission queue is discarded upon the arrival of a more recent one. In such a setting, we were able to derive the stationary distribution of AoI. In the second part of this chapter, we have gone one step further and utilized the PMF of AoI to tackle optimization problems for WNCS. We were able to do so by deriving non-linear age-penalty functions that depend on the control applications' dynamics. Our analysis has been based on two metrics, namely the MSE and the LQG cost capturing the estimation and control performances, respectively.

The two networking scenarios that have been considered in this chapter enable us to obtain stationary distribution of age. Our results demonstrate how such distributions can be used for cross-layer performance optimization in the context of WNCS. However, in some scenarios, such (stationary) distributions may not exist or can be computationally challenging to calculate. For instance, if the channel conditions are time-varying, and/or if the network is actively managing the network resources during run-time, the distribution is constantly manipulated through the in-network decision-making. To cover such scenarios, instead of using distributions of AoI, the following chapter explores different tools and methods for cross-layer design in WNCS.

Chapter 4

Optimal Scheduling for Wireless Networked Control Systems

In the previous chapter, we have shown the interplay between communications and control in two example networking scenarios. The main results from the previous chapter suggest that timely and regular transmissions play a central role in networked control and one way to improve control performance is to make networks more suitable for control applications. Moreover, the previous results indicate that information freshness, as well as, the QoC can directly or indirectly be controlled through careful configuration of the network. In this chapter, we change our perspective and set our focus on online decision-making mechanisms, specifically within the data-link layer. We tackle the centralized scheduling problem in a wireless communication network that comprises multiple heterogeneous feedback control loops contending for the limited network resources. The key conceptual difference between the following content and the previous chapter is that the network optimization becomes a more dynamic and reactive process. Nevertheless, the network's awareness of semantics of information, such as freshness and value remains important throughout this chapter's content.

Centralized wireless resource scheduling is one of the key concepts and research challenges in today's wireless networks and standards. Despite the complexity the scheduling introduces, when compared to simple contention-based MAC protocols, e.g., ALOHA, the benefit of managing the network resources centrally is significantly larger in most of the cases. Current cellular Radio Access Network (RAN) architectures rely on centralized scheduling, in which the physical radio resources are managed by a central entity called scheduler [DPS20]. A typical RAN architecture locates the scheduler at the Base Station (BS).

Generally, the conventional scheduling algorithms base their decisions on the statistics collected from the users, such as the estimated channel quality, buffer status, and throughput history [Cap+13]. With the introduction of new technology and services, e.g., remote

monitoring, industrial applications, unorthodox metrics such as the AoI have started finding their use in the networking protocol stack, particularly in the central orchestration of network resources in time-critical application scenarios[1]. However, the application of the AoI metric specifically for control over networks is an immature research field. Therefore, in this chapter, we seek answers to the following questions:

1. Is it beneficial to consider the AoI for wireless resource scheduling to achieve better control performance? How does it compare to incorporating value of information into decision-making?
2. Can the semantics of information beyond AoI be considered managing the wireless resources centrally, without assuming a global knowledge at the scheduler?
3. How can the network solve the centralized resource allocation problem optimally if:
   a) the channel conditions are static, represented by constant packet success probabilities between source-destination pairs?
   b) the channel conditions are time-varying, represented by changing packet success probabilities over time?

To answer these questions, we start with a cellular UL and DL scenario that involves multiple WNCS communicating over a BS. We investigate whether AoI sufficiently captures the importance of each transmission or further semantics beyond age are necessary to prioritize the users more efficiently. In the first study, we do not consider any packet loss in the network and propose a greedy UL and DL scheduler. The first study tackles the first and second research questions specified above and reflects the core findings of our publication [Aya+19]. The results reveal that a significantly higher performance can be achieved if the scheduler does not consider only the timeliness aspect of a packet but also its value.

To address the question 3.a), we introduce packet loss and formulate an infinite horizon discounted cost problem targeting the minimization of the network-induced estimation error. In the same study, which is mainly based on [AVK20], we compare the achieved performance to our firstly proposed heuristic approach. Afterwards, we allow the packet loss probability to vary over time. Thereby, we find the optimal action by looking at a finite time horizon into the future, summarizing our work [Aya+20a]. This approach is extended in [Aya+23] through the joint consideration of the UL and DL in the final main section. The studies involving dynamic link qualities, i.e., [Aya+20a; Aya+23] target the question 3.b) specifically.

[1] We name some prominent examples of scheduling algorithms minimizing AoI in chapter 2.

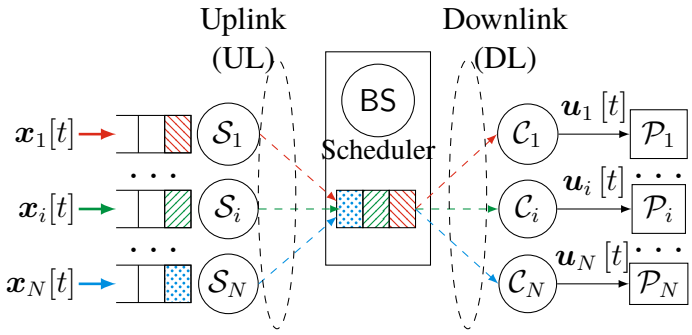


Figure 4.1: The considered scenario consisting of N feedback control loops sharing a star network. The communication between sensor-controller pairs are enabled through a base station (BS). The wireless resources on the UL, as well as DL are allocated by a centralized scheduler located at the BS.

4.1 Greedy Scheduler for WNCS

4.1.1 System Model

We consider a network shared by N independent LTI feedback control loops. Each sub-system is comprised of a sensor $\mathcal{S}_i$, a controller $\mathcal{C}_i$, and a plant $\mathcal{P}_i$. While each plant-controller pair is co-located, the sensors operate remotely; hence, each control sub-system contains a non-ideal sensor-to-controller link. We further assume that each sensor-controller pair cannot communicate directly with each other, but are able to exchange information over a base station (BS). As a result, the periodic observations made by each $\mathcal{S}_i$ are transmitted in the form of status update packets first to the BS, from where they are forwarded to the corresponding controller $\mathcal{C}_i$. Such a topology is called a *star network* in the literature. Fig. 4.1 illustrates the describe scenario. Note that the sensor-to-BS link and BS-to-controller links are called UL and DL, respectively.

The smallest time unit in our system model is a transmission slot of unit length, which is indexed by $t \in \mathbb{N}$. Every transmission starting in a time slot t ends within the same time slot. The behavior of each control sub-system i is governed by the difference equation in discrete-time as:

$$\boldsymbol{x}_i[t+1] = \boldsymbol{A}_i\boldsymbol{x}_i[t] + \boldsymbol{B}_i\boldsymbol{u}_i[t] + \boldsymbol{w}_i[t], \tag{4.1}$$

with the system noise being independent and identically distributed according to a zero-mean Gaussian distribution with the diagonal covariance matrix $\boldsymbol{\Sigma}_i$, i.e., $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_i)$. Note that the sampling period is selected equal to the slot frequency, thus we use t to index sampling periods in (4.1)[2].

[2]One can imagine a slot duration of one millisecond corresponding to a sampling frequency of one kHz, or a slot duration of ten milliseconds corresponding to 100 Hz.

The medium access on both links is orchestrated by a centralized scheduler located at the BS. Let us introduce a decision variable $\pi_i^{UL}[t] \in \{0,1\}$ indicating whether $\mathcal{S}_i$ is scheduled for an UL transmission in time slot t, i.e.:

$$\pi_i^{UL}[t] = \begin{cases} 1, & \text{if } \mathcal{S}_i \text{ is scheduled on the UL in time slot } t \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

Similarly, another variable $\pi_i^{DL}[t] \in \{0,1\}$ takes the value of one if a packet intended for $\mathcal{C}_i$ is transmitted in the t-th slot on the DL. For presentation purposes and to capture the kernel idea of the presented content, we assume that only one user is allowed to transmit on each link[3]:

$$\sum_{i=1}^{N} \pi_i^{UL}[t] \leq 1, \text{ and } \sum_{i=1}^{N} \pi_i^{DL}[t] \leq 1. \tag{4.3}$$

Throughout this section, we assume that every transmission is successful and takes one full slot duration. Moreover, if a new piece of information becomes available, for instance, at the sensor after a sampling event, any older status update packet is discarded and replaced by the more recent information. This ensures that every sensor has always one packet in its UL transmission queue. In addition, having received a new information after a successful transmission on the UL, the BS replaces any older packet from the same sub-system with the fresher update packet. This means that each flow i has at most one packet waiting to be transmitted on the DL. Discarding older packets come from the assumption that the status is Markovian and the sensor observations are perfect. Therefore, having received a more recent update, the controller does not benefit from receiving an older one.

Having said that, we introduce three new variables $\nu_i^{\mathcal{S}_i}[t]$, $\nu_i^{BS}[t]$, and $\nu_i^{\mathcal{C}_i}[t]$ that denote the generation time of the most recent information available at $\mathcal{S}_i$, at the BS, and at $\mathcal{C}_i$, respectively. Due to the considered packet discarding policy and periodic sampling, it holds that $\nu_i^{\mathcal{S}_i}[t] = t, \forall t$. As a result, the behavior of $\nu_i^{BS}[t]$ follows as:

$$\nu_i^{BS}[t+1] = \begin{cases} t, & \text{if } \pi_i^{UL}[t] = 1 \\ \nu_i^{BS}[t], & \text{otherwise.} \end{cases} \tag{4.4}$$

Moreover, $\nu_i^{\mathcal{C}_i}[t]$ can be characterized by the following equation:

$$\nu_i^{\mathcal{C}_i}[t+1] = \begin{cases} \nu_i^{BS}[t], & \text{if } \pi_i^{DL}[t] = 1 \\ \nu_i^{\mathcal{C}_i}[t], & \text{otherwise.} \end{cases} \tag{4.5}$$

Note the transmission delay of one slot in equations (4.4) and (4.5).

[3]Note that further variations of UL and DL resource constraints are possible and have been studied in [Aya+19].

Before the introduction of the greedy scheduling policy, let us define the instantaneous AoI at each network node separately as:

$$\Delta_i^{\mathcal{S}_i}[t] = 0, \tag{4.6}$$
$$\Delta_i^{BS}[t] = t - \nu_i^{BS}[t], \tag{4.7}$$
$$\Delta_i^{\mathcal{C}_i}[t] = t - \nu_i^{\mathcal{C}_i}[t], \tag{4.8}$$

The first row follows from the assumption that each sensor is able to observe the system state $\boldsymbol{x}_i[t]$ without any measurement delay. Thus, the AoI at the sensor is zero at all times. The AoI at the BS, as well as at each controller are obtained by taking the difference between the current time and the freshest information's timestamp, as usual. It is important to mention that the AoI at the BS for a sub-system i has to be less than or equal to the AoI at $\mathcal{C}_i$, because every packet received at $\mathcal{C}_i$ must have passed through the BS. In mathematical terms, $\Delta_i^{BS}[t] \leq \Delta_i^{\mathcal{C}_i}[t], \forall i, t$.

4.1.2 Greedy UL & DL Scheduling Policy

In this section, we introduce two heuristic scheduling policies prioritizing flows on the UL and DL based on the selected metric.

4.1.2.1 Age of Information Scheduler

As the name suggests, the AoI scheduler aims to increase information freshness in the network. As a results, the final goal is given as:

$$\min_{\pi_i^{UL}[t], \pi_i^{DL}[t]} \quad \limsup_{T\to\infty} \frac{1}{T}\frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^{N} \Delta_i^{\mathcal{C}_i}[t]. \tag{4.9}$$

To solve (4.9), we leverage the results from [Kad+18] suggesting that the greedy scheduling policy is in fact age-optimal if all UL transmissions have the same success probability. Although our network is two-hop, we can treat each link separately and first minimize the age at the BS as:

$$\max_{\pi_i^{UL}[t]} \quad \pi_i^{UL}[t] \cdot \Delta_i^{UL}[t], \tag{4.10}$$
$$\text{s.t.} \quad \sum_{i=1}^{N} \pi_i^{UL}[t] \leq 1,$$

and then treat the BS as source, hence define the DL problem as:

$$\max_{\pi_i^{DL}[t]} \quad \pi_i^{DL}[t] \cdot \Delta_i^{DL}[t], \tag{4.11}$$
$$\sum_{i=1}^{N} \pi_i^{DL}[t] \leq 1.$$

Ties are broken arbitrarily. The resulting $\{\pi_i^{UL}[t], \pi_i^{DL}[t]\}$ pair maximizing the RHS of (4.10) and (4.11) is the UL and DL schedule for the current transmission slot t.

4.1.2.2 Value of Information Scheduler

It is evident from equations (4.10) and (4.11) that the AoI scheduler solely takes the instantaneous age of each sub-system, hence aiming to provide Fresher information to the controllers. Now we introduce another heuristic scheduler that goes one step further and considers the expected MSE stemming from the absence of fresh information. We call our proposed scheduler the *value of information (VoI)* scheduler, whereas the value of each packet is matched with the expected MSE reduction at the receiver in case of a successful transmission.

Similar to the AoI scheduler, we select the sub-system with the highest MSE on each link. In contrast to (4.9), the primary goal of the VoI scheduler is not maximizing the information freshness, but rather achieving a higher estimation performance captured by the reduced MSE, i.e.:

$$\min_{\pi_i^{UL}[t], \pi_i^{DL}[t]} \quad \limsup_{T\to\infty} \frac{1}{T}\frac{1}{N} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \mathbb{E}\left[\|\boldsymbol{e}_i[t]\|^2\right]. \tag{4.12}$$

Next, we are interested in finding a heuristic solution to (4.12) and therefore define the following UL and DL scheduling policies:

$$\max_{\pi_i^{UL}[t]} \quad \pi_i^{UL}[t] \cdot g(\Delta_i^{BS}[t]; t), \tag{4.13}$$
$$\text{s.t.} \quad \sum_{i=1}^{N} \pi_i^{UL}[t] \leq 1,$$

and:

$$\max_{\pi_i^{DL}[t]} \quad \pi_i^{DL}[t] \cdot g(\Delta_i^{\mathcal{C}_i}[t]; t), \tag{4.14}$$
$$\text{s.t.} \quad \sum_{i=1}^{N} \pi_i^{DL}[t] \leq 1,$$

with $g_i(\Delta_i[t]; t)$ defined as in (3.17). In simple words, the greedy VoI scheduler selects the user with the highest instantaneous MSE. Our solution is an heuristic solution, thus

provides an upper bound for the scheduling problem in (4.12). Nevertheless, although it falls behind at providing fresh information, as we are going to show in the following discussion, it outperforms the AoI scheduler w.r.t. control performance.

4.1.3 Evaluation

In this section, we present a numerical evaluation comparing the performances of the two heuristic schedulers that we have introduced in section 4.1.2. Our evaluation is based on simulation results, whereas each simulation run consists of $T = 20000$ slots. To simplify the analysis, we consider scalar feedback control loops, i.e., $\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{L}_i, \boldsymbol{\Sigma}_i \in \mathbb{R}, \forall i$. However, the control loops are assumed to be of heterogeneous type, whereas we consider four classes of control loops. A class is characterized by its system matrix as $\boldsymbol{A}_i \in \{1.0, 1.1, 1.2, 1.3\}$. In other words, two sub-systems i and j have different system matrices if they are not from the same class, e.g., $\boldsymbol{A}_i = 1.0$ and $\boldsymbol{A}_j = 1.3$. Analogously, if i and j are from the same class of applications, their system matrices are identical, i.e., $\boldsymbol{A}_i = \boldsymbol{A}_j$. The feedback gain matrix is chosen according to the deadbeat control strategy $\boldsymbol{L}_i = \boldsymbol{A}_i$. Input matrices are equal among loops, i.e., $\boldsymbol{B}_i = 1.0, \forall i$, as well as the characteristics of the system noise, i.e., $\boldsymbol{w}_i \sim \mathcal{N}(0, 1), \forall i$.

As Key Performance Indicator (KPI)s, we use the long-term average AoI per sub-system to quantify the achieved information freshness:

$$\overline{\Delta} = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\Delta_i^{\mathcal{C}_i}[t]. \tag{4.15}$$

Moreover, the average MSE is measured to quantify the adverse effects of the communication network on the remote state estimation. The $\overline{MSE}$ is given as:

$$\overline{MSE} = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\left\|\boldsymbol{e}_i[t]\right\|^2. \tag{4.16}$$

We vary the number of users to capture the relationship between the resource scarcity and information freshness in the network. Specifically, we gradually increase N from four to twenty in discrete steps as $N \in \{4, 8, \ldots, 20\}$. Thereby, we keep the number of sub-systems per class balanced. This means, when $N = 8$, each class is comprised of two-subsystems having equal system matrices, i.e., $\boldsymbol{A}_1 = \boldsymbol{A}_2 = 1.0$, $\boldsymbol{A}_3 = \boldsymbol{A}_4 = 1.1$, etc. Fig. 4.2 illustrates the expected MSE at Δ_i for each control loop class. It is evident that the MSE grows much faster for systems of the fourth class with $\boldsymbol{A}_4 = 1.3$ than it does for those having $\boldsymbol{A}_1 = 1.0$.

Fig. 4.3 shows the average AoI for increasing N when AoI- and VoI scheduling policies are employed. Firstly, we notice that when there are more users in the network, the AoI

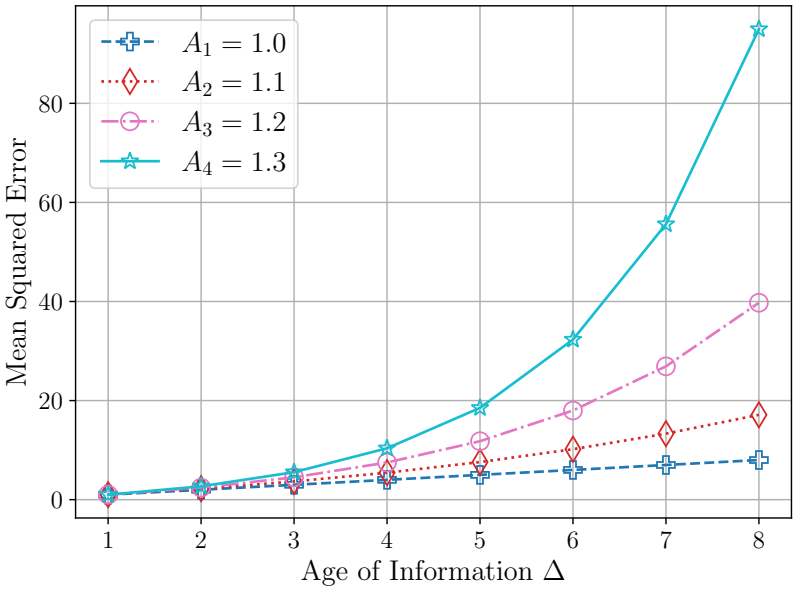


Figure 4.2: The expected MSE at age Δ for the considered scalar control loops. Note that A_i is varied among sub-systems. $W_i = 1.0$ for all sub-systems.

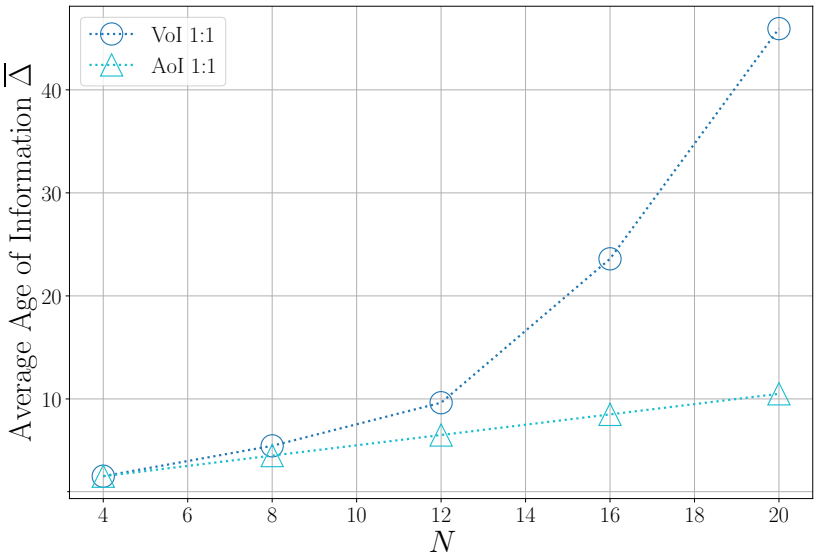


Figure 4.3: Long-term average AoI per sub-system per slot, i.e., $\overline{\Delta}$ for varying number of feedback control loops, i.e., N. A lower $\overline{\Delta}$ indicates a higher information freshness. The sub-systems are of heterogeneous type representing different control task-criticalities.

increases for both schedulers. This is an expected result, since more users lead to a higher resource scarcity, thus, the average transmission frequency of a sub-system decreases. In addition, the AoI scheduler achieves a lower $\overline{\Delta}$ indicating a higher information freshness. As a result, if our primary goal were providing the controllers with regular and timely updates, we can confidently say that the AoI scheduler is a better choice than the VoI scheduler.

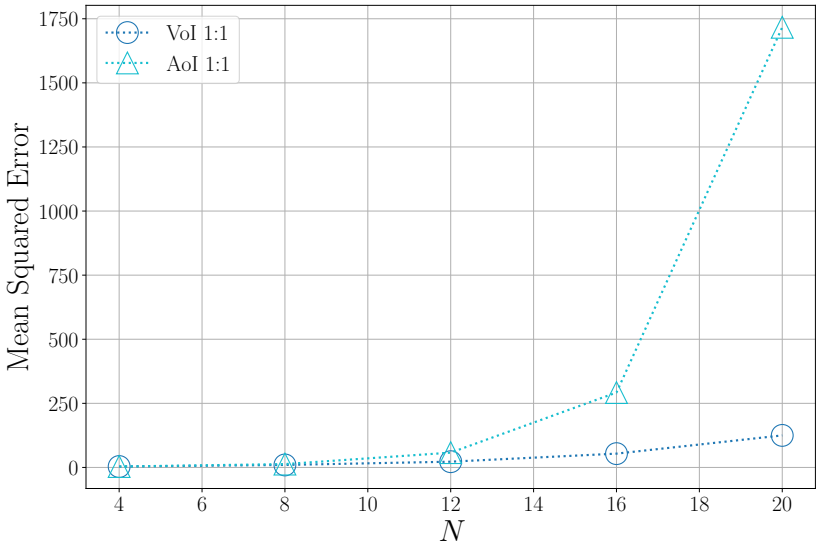


Figure 4.4: Mean squared error (MSE) per sub-system per slot capturing the estimation accuracy. A lower MSE corresponds to a higher performance. The sub-systems are of heterogeneous type representing different control task-criticalities.

Generally speaking, a lower AoI is desirable as it means that the controllers have up-to-date information about the system state. However, optimizing w.r.t. AoI may not necessarily lead to a higher performance, especially if the applications have heterogeneous demands. To validate this statement, let us compare the two schedulers in terms of $\overline{MSE}$. Fig. 4.4 presents the achieved mean squared (estimation) error, for increasing N. One can observe from the figure that for a low number of users, both schedulers deliver relatively close results w.r.t $\overline{MSE}$. On the other hand, as N increases, the AoI scheduler, which is the age-optimal one for our considered scenario, falls behind the heuristic VoI scheduler significantly. This shows that although having fresh information is generally beneficial in real-time monitoring and control scenarios, the network resources should not be distributed solely by looking at the AoI alone. On the contrary, if the primary goal is to perform better in terms of task-specific performance criteria, one needs to explore other metrics than age as it is not capable of capturing application-dependent criticalities except for timeliness.

4.2 Optimal Scheduler for WNCS: The Constant Loss Case

In the previous section, we have shown the superiority of the VoI over AoI as a scheduling metric for networked control systems. However, our study has not involved any packet loss in the network model. To bring our model one step closer to the reality, we introduce packet loss into our model making the outcome of any scheduled transmission a stochastic process.

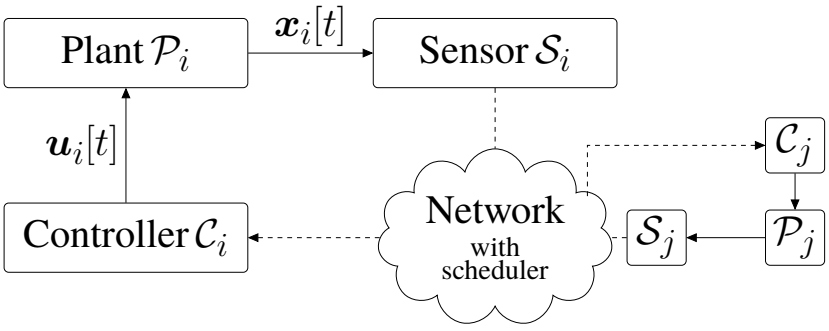


Figure 4.5: The considered scenario consisting of N feedback control loops sharing a wireless channel. The medium access is managed by a centralized scheduler.

4.2.1 System Model

We consider N control sub-systems closed over a wireless network. The time dynamics of the control systems follow the discrete-time LTI model from (4.1). In contrast to section 4.1, the BS-to-controller link is assumed to be ideal. On the other hand, the UL channel between the sensors and the base station is shared, with the network access being managed by a centralized scheduler. Time is divided non-overlapping slots, indexed by t.

Each sub-system i consists of a plant $\mathcal{P}_i$, a sensor $\mathcal{S}_i$, and a controller $\mathcal{C}_i$, as depicted in Fig. 4.5. If a sensor i is granted channel access by the scheduler in time slot t, it transmits the latest state information $\boldsymbol{x}_i[t]$ to the controller $\mathcal{C}_i$[4]. In case of a successful transmission, the controller uses the new information for the remote estimation estimate of the system state, as in (3.15). Each update packet occupies a single transmission slot, which is also equal to the sampling period of sub-systems. We assume a constant delay of one time slot, i.e., any successful transmission scheduled in time slot t can be used by the controller to obtain $\boldsymbol{u}_i[t+1]$.

We assume packet erasure channel, where each transmission by $\mathcal{S}_i$ is successfully received by the controller with a constant probability, i.e., $p_i[t] = p_i, \forall t$ with $p_i \in (0, 1]$. Let us introduce a binary variable $\delta_i[t] \in \{0, 1\}$ denoting the transmission outcome and a second variable $\pi_i[t] \in \{0, 1\}$ indicating the scheduling decision for sub-system i. In other words, if the scheduler allows $\mathcal{S}_i$ to transmit in time slot t, then $\pi_i[t] = 1$. Moreover, if $\pi_i[t] = 1$, then the chance of the transmission to be successful is given as:

$$\Pr[\delta_i[t] = 1 \mid \pi_i[t] = 1] = p_i. \tag{4.17}$$

Similarly, if a sensor $\mathcal{S}_i$ transmits after being granted access, the failure probability is given as:

$$\Pr[\delta_i[t] = 0 \mid \pi_i[t] = 1] = 1 - p_i. \tag{4.18}$$

[4]The sensors replace their packet in their transmission buffer with a more recent measurement if they are not granted channel access.

Note that for any user i that has not been scheduled for slot t, the chance of a successful transmission is zero, as sensors transmit only if they are allocated a transmission slot, i.e., $\Pr[\delta_i[t] = 1 \mid \pi_i[t] = 0] = 0$.

Having received a new update, the controller is always provided with a more recent system state information, as there are no out-of-order packets in our model. In other words, the generation time of the most recent information takes a greater value upon a successful reception. Let $\nu_i[t] \in \mathbb{N}$ denote the generation time of the most recent system state that the controller $\mathcal{C}_i$ has observed until t, i.e., $\nu_i[t] = \sup\{\tau \in \mathbb{N} : \tau < t, \pi_i[\tau] \cdot \delta_i[\tau] = 1\}$. If $\mathcal{S}_i$ is scheduled for transmission in time slot t and the transmission is successful, then $\nu_i[t]$ is set to t in the subsequent slot. Otherwise, it stays constant as $\mathcal{C}_i$ has not received a new update. As a result, the AoI can be characterized by the following equation:

$$\Delta_i[t+1] = \begin{cases} 1, & \text{if } \pi_i[t] \cdot \delta_i[t] = 1 \\ \Delta_i[t] + 1, & \text{if } \pi_i[t] \cdot \delta_i[t] = 0 \end{cases} \tag{4.19}$$

The first row follows from the constant delay assumption of one sampling period. Thus, a successfully received packet sets the AoI to one[5]. In all other cases, covered by the second row, the AoI increases by one every time slot, hence linearly in discrete-time. The equation (4.19) characterizes the relationship between the scheduling decision and the AoI. Namely, if the scheduler allows a sub-system i to transmit in slot t, then with a constant probability of p_i, the age is reset to one in the following time slot. On the other hand, for all other sub-systems the AoI increment by one occurs with a probability of one.

Let us introduce a vector $\boldsymbol{s}[t] \in \boldsymbol{\mathcal{S}}$, the i-th element of which is $\Delta_i[t]$, i.e., $\boldsymbol{s}[t] \triangleq (\Delta_1[t], \Delta_2[t], \ldots, \Delta_N[t])$. The state space $\boldsymbol{\mathcal{S}}$ is the N-dimensional set of positive numbers, i.e., $\mathbb{N}_+^N$. Similarly, let $a[t] \in \boldsymbol{\mathcal{A}}$ denote the scheduling action with $a[t] = i$ indicating $\pi_i[t] = 1$ and $\pi_j[t] = 0, \forall j \neq i$. The action space $\boldsymbol{\mathcal{A}}$ is given as $\{\varnothing, 1, \ldots, N\}$. Here, $a[t] = \varnothing$ allows a slot to be idle, i.e., a slot, in which none of the sub-systems is granted channel access.

4.2.2 Problem Statement

We are interested in stationary scheduling policies $\boldsymbol{\Pi}$ that map the state space to admissible action space as $\boldsymbol{\Pi} : \boldsymbol{\mathcal{S}} \mapsto \boldsymbol{\mathcal{A}}$. In contrast to section 4.1, in which the slots were allocated to the user with the highest MSE, we formulate an infinite horizon optimization problem that aims to achieve a minimum MSE in the network by following the framework in [Ber95]. To that end, we model the entire system as an MDP with states $\boldsymbol{s}$, actions a, and the immediate

[5]Hence, the AoI takes a positive integer value, i.e., $\Delta_i[t] \in \mathbb{N}_+, \forall i, t$.

state cost $g(\boldsymbol{s})$ as:

$$g(\boldsymbol{s}[t]) \triangleq \sum_{i=1}^{N} \underbrace{\sum_{q=0}^{\Delta_i[t]-1} \mathsf{tr}\left(\left(\boldsymbol{A}_i^T\right)^q \boldsymbol{A}_i^q \boldsymbol{\Sigma}_i\right)}_{\text{as in (3.17)}}. \tag{4.20}$$

Simply put, (4.20) gives the total MSE in the network given a network state $\boldsymbol{s}[t]$.

Our objective is to find a stationary optimal policy $\boldsymbol{\pi}^* \in \boldsymbol{\Pi}$ that minimizes the total MSE over an infinite horizon, given an initial state $\boldsymbol{s}[0]$, i.e.:

$$\mathcal{J}_{\boldsymbol{\pi}^*}(\boldsymbol{s}[0]) = \min_{\boldsymbol{\pi}\in\boldsymbol{\Pi}} \mathcal{J}_{\boldsymbol{\pi}}(\boldsymbol{s}[0]) \tag{4.21}$$

with:

$$\mathcal{J}_{\boldsymbol{\pi}}(\boldsymbol{s}[0]) = \lim_{T\to\infty} \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{t=0}^{T-1} \gamma^t g(\boldsymbol{s}[t])\right]. \tag{4.22}$$

$\mathbb{E}_{\boldsymbol{\pi}}$ indicates that the expectation is taken w.r.t. the transition probabilities between the states when the scheduling policy $\boldsymbol{\pi}$ is selected. Moreover, $\gamma \in (0,1)$ is a positive scalar value called *discount factor*.

Definition 1. *A scheduling policy is called γ-optimal if it minimizes the discounted cost over an infinite horizon, thus minimizing the right hand side of* (4.22) *[Ber95].*

The role of the discount factor γ is adjusting the effect of the expected future costs on immediate decision. Specifically, any cost that is incurred h slots from now is worth only γ^h times what it would cost if it were received immediately. Hence, as γ approaches one, the scheduler weights future penalties more and thereby becomes more far-sighted.

4.2.3 Scheduler Design

4.2.3.1 Approximating Sequence of the MDP

Theoretically, the MDP consists of infinite states as the age is not upper-bounded, thus can grow infinitely. In fact, the state space corresponds to $\mathbb{N}_+^N$, thus by definition is a countable N-dimensional infinite set. As it is not possible to iterate through infinite states in practice, we apply a technique called *Augmentation Type Approximating Sequence (ATAS)* of the MDP [Sen09].

Let us define a non-empty finite state space $\boldsymbol{\mathcal{S}_M} \subset \boldsymbol{\mathcal{S}}$ that includes only those states with all elements being less than or equal to M, i.e., $\Delta_i \leq M, \forall i$, hence dividing $\boldsymbol{\mathcal{S}}$ into two disjoint sets. On the other hand, the states that contain at least one element exceeding M belong to the infinite set $\boldsymbol{\mathcal{S}} \setminus \boldsymbol{\mathcal{S}_M}$. Put differently, as long as the age values of all sub-systems are less than $M+1$, the state is considered to be in $\boldsymbol{\mathcal{S}_M}$.

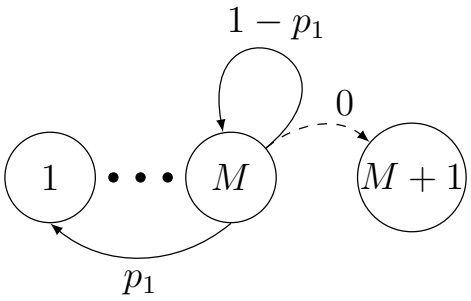


Figure 4.6: Augmentation type approximating sequence for the exemplary MDP with one sub-system i and the maximum augmented age M. Given the current state $\boldsymbol{s}[t] = M$ and action $a[t] = i$, we redistribute the excess probability of a transition from M to $M+1$ back to the state M, i.e., $\Pr[\boldsymbol{s}[t+1] = M | \boldsymbol{s}[t] = M, a[t] = i] = 1 - p_1$.

Now, we apply the ATAS method that approximates the original MDP by limiting the state space. In particular, suppose the following two scenarios: 1) in state $\boldsymbol{s} \in \boldsymbol{\mathcal{S}_M}$ an action a is taken. With a certain probability $p^a_{\boldsymbol{ss'}}$ it leads to a next state $\boldsymbol{s'} \in \boldsymbol{\mathcal{S}_M}$ within the same finite state space $\boldsymbol{\mathcal{S}_M}$. 2) in state $\boldsymbol{s} \in \boldsymbol{\mathcal{S}_M}$ an action a is taken and a transition to another state $\hat{\boldsymbol{s}} \notin \boldsymbol{\mathcal{S}_M}$ is possible with the probability $p^a_{\boldsymbol{s\hat{s}}} \geq 0$. The probability $p^a_{\boldsymbol{s\hat{s}}}$ is called *excess probability* and is defined as the transition probability from a state $\boldsymbol{s} \in \boldsymbol{\mathcal{S}_M}$ to another state $\hat{\boldsymbol{s}} \notin \boldsymbol{\mathcal{S}_M}$ outside the finite space. When the ATAS is applied, the transition probabilities conforming with the first scenario remain unchanged. However, for the second case, all excess probabilities are redistributed to states $\boldsymbol{s'}$ in $\boldsymbol{\mathcal{S}_M}$ according to a probability distribution $f_{\boldsymbol{s'}}(\boldsymbol{s}, a, \hat{\boldsymbol{s}}, M)$. If we denote the resulting MDP by MDP_M, the definition of the ATAS follows from [Sen09].

Definition 2. *A sequence MDP_M is an augmentation type approximating sequence of an MDP if for each $\hat{\boldsymbol{s}} \notin \boldsymbol{\mathcal{S}_M}$, given $\boldsymbol{s}, \boldsymbol{s'} \in \boldsymbol{\mathcal{S}_\mathcal{M}}$, and $a \in \boldsymbol{\mathcal{A}}$, there exists a probability distribution $f_{\boldsymbol{s'}}(\boldsymbol{s}, \boldsymbol{a}, \hat{\boldsymbol{s}}, M)$ such that*

$$p^a_{\boldsymbol{ss'}}(M) = p^a_{\boldsymbol{ss'}} + \sum_{\hat{\boldsymbol{s}} \in \boldsymbol{S} \setminus \boldsymbol{S_M}} p^a_{\boldsymbol{s\hat{s}}}\, f_{\boldsymbol{s'}}(\boldsymbol{s}, a, \hat{\boldsymbol{s}}, M). \tag{4.23}$$

The application of the ATAS in our use case can be explained as follows: Let $\tilde{\Delta}_i^{(M)}[t]$ be the augmented AoI for the i-th sub-system with $\tilde{\Delta}_i^{(M)}[t] = \min(\Delta_i[t], M)$. That is, we truncate AoI to M if its value exceeds M. As we do not allow AoI values greater than M, the newly created state space is finite with $\boldsymbol{S_M} = \{1, \ldots, M\}^N$. Fig. 4.6 illustrates the resulting MDP_M for a single sub-system i given that the scheduler decides to schedule the user at age M, i.e., $a[t] = i$. The figure shows that although the actual age of the sub-system grows beyond M, our model approximates prevents the augmented age to grow beyond M by redistributing the excess probability $1 - p_1$ back to the maximum allowed age M. Hence, the

resulting MDP$_M$ is an approximation of the actual behavior of the system. To avoid visual clutter, we proceed with the approximating sequence to simplify the notation and consider the resulting state space $\boldsymbol{S_M}$.

As we have the states, actions and transition probabilities of the finite MDP$_M$, let us define the immediate cost associated with each state-action pair as:

$$C(\boldsymbol{s}[t], a[t]) = \sum_{i=1}^{N} g(\tilde{\Delta}_i^{(M)}[t]), \tag{4.24}$$

with $\boldsymbol{s} \in \boldsymbol{S_M}$. Note that $C(\boldsymbol{s}[t], \boldsymbol{a}[t])$ considers only the MSE and is independent of the taken action. Nevertheless, one can easily introduce an additive communication cost term to the equation, to capture, for instance, the energy cost of a transmission or possibly heterogeneous resource demands by sub-systems to transmit a single packet.

4.2.3.2 γ-optimal Discounted Error Scheduler (DES)

We propose a γ-optimal stationary scheduling policy minimizing the discounted infinite horizon problem given in (4.22). The state costs are modeled by the approximations in (4.24) for a given augmentation level M. To that end, we employ the standard value iteration approach with dynamic programming from [Ber95]. The proposed algorithm solves the approximated problem optimally for a given γ and an augmentation level M.

Value iteration: Each state $\boldsymbol{s}$ is associated with an initial value function $\mathcal{J}_0(\boldsymbol{s})$. We iterate through all states, while updating the cost of each state as:

$$\begin{aligned}
\mathcal{J}_{k+1}(\boldsymbol{s}) &= \min_{a \in \boldsymbol{\mathcal{A}}} \{\mathbb{E}\left[C(\boldsymbol{s}, \alpha) + \gamma \mathcal{J}_k(\boldsymbol{s'})\right]\} \\
&= \min_{a \in \boldsymbol{\mathcal{A}}} \{C(\boldsymbol{s}, \alpha) + \gamma \mathbb{E}[\mathcal{J}_k(\boldsymbol{s'})]\} \\
&= \min_{a \in \boldsymbol{\mathcal{A}}} \left\{ C(\boldsymbol{s}, \alpha) + \gamma \sum_{\boldsymbol{s'} \in \boldsymbol{S_M}} p^{\alpha}_{\boldsymbol{ss'}}\, \mathcal{J}_k(\boldsymbol{s'}) \right\}.
\end{aligned} \tag{4.25}$$

The cost of each state is dictated by the discounted expected cost after taking the optimal action, i.e., the action that minimizes the RHS of the Bellman equation (4.25). As before, $p^{a}_{\boldsymbol{ss'}}$ denotes the success probability from a state $\boldsymbol{s}$ to a successor state $\boldsymbol{s'}$ after taking the action α, i.e., $\Pr[\boldsymbol{s}[t+1] = \boldsymbol{s'} \mid \boldsymbol{s}[t] = \boldsymbol{s}, a[t] = \alpha]$. In [Ber95], it has been shown that as k goes to infinity, the value functions will converge to an optimal cost, i.e., $\lim_{k\to\infty} \mathcal{J}_k = \mathcal{J}^*$ for any given initial value function $\mathcal{J}_0$. After the convergence, the optimal actions minimizing the RHS of (4.25) constitute the stationary optimal policy $\boldsymbol{\pi}^*$. Algorithm 2 illustrates the practical implementation of the value iteration algorithm for zero initial cost, i.e., $\mathcal{J}_0(\boldsymbol{s}) = 0, \forall \boldsymbol{s}$. A positive threshold value θ is used to decide whether the algorithm has converged. The lower θ is selected, the more precise are the final value functions.

Algorithm 2 Value iteration

Require: θ
  for all $\boldsymbol{s} \in \boldsymbol{S_M}$ **do**
    $\mathcal{J}(\boldsymbol{s}) \leftarrow 0$; {Initialize state costs, e.g., as zero}
  end for
  repeat
    $u \leftarrow 0$; {Helper variable to track the changes in state costs}
    for all $\boldsymbol{s} \in \boldsymbol{S_M}$ **do**
      Update $\mathcal{J}^{+}(\boldsymbol{s})$ as in (4.25);
      $u \leftarrow \max(u, |\mathcal{J}^{+}(\boldsymbol{s}) - \mathcal{J}(\boldsymbol{s})|)$; {Track the maximum change during iteration}
      $\mathcal{J}(\boldsymbol{s}) \leftarrow \mathcal{J}^{+}(\boldsymbol{s})$; {Update the state cost}
    end for
  until $u \leq \theta$
  $\pi^{*}(\boldsymbol{s}) \leftarrow \arg\min_{a \in \boldsymbol{\mathcal{A}}} \mathcal{J}(\boldsymbol{s})$; {Optimal action at $\boldsymbol{s}$ minimizes the RHS of (4.25)}
  return $\boldsymbol{\pi^{*}}$

4.2.4 Evaluation

In this section, we compare the performance of the DES scheduler to the greedy approach from section 4.1, as well as to the γ-optimal AoI scheduler proposed in [HMD17]. Our evaluation is based on numerical results generated through simulations.

4.2.4.1 γ-optimal AoI Scheduler (AoIS)

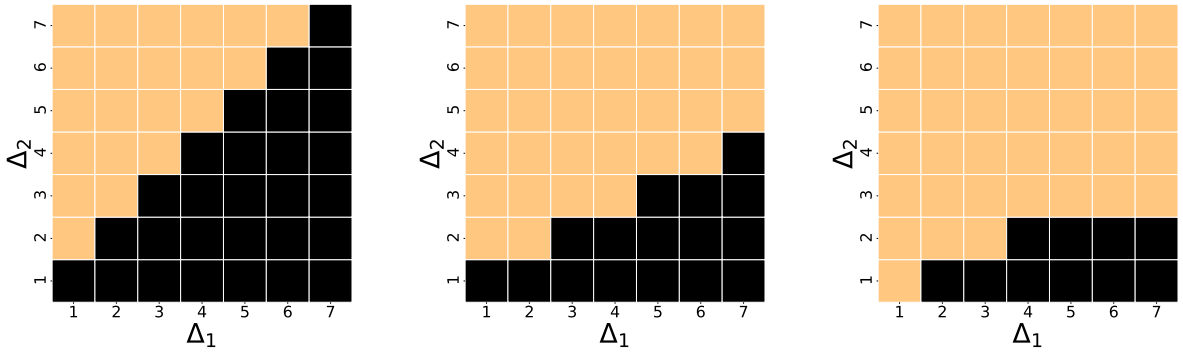
The idea behind the AoIS is very similar to our proposed DES except for the immediate cost of a state. Namely, instead of using the MSE, the each state is penalized by the sum of the instantaneous ages at each controller, i.e.:

$$C_{AoI}(\boldsymbol{s}[t], a[t]) = \sum_{i=1}^{N} \tilde{\Delta}_i^{(M)}[t]. \tag{4.26}$$

In other words, the γ-optimal AoI-based scheduling policy can be obtained by simply replacing the cost function $C(s, a)$ in (4.25) with the one above. As we are going to show in the remainder of this section, the resulting policy outperforms all of the selected schedulers w.r.t. information freshness.

4.2.4.2 Greedy Error Scheduler (GES)

The GES considered in this section is a slightly modified version of the VoI scheduler from section 4.1. Different than the VoI scheduler, which is channel-unaware by definition, the GES weights the age-penalty function $g(\Delta_i)$ by the packet success probability p_i. The inclusion of the channel quality in decision-making is necessary as we have introduced the packet loss



(a) AoIS policy (b) GES policy (c) DES policy

Figure 4.7: Scheduling policies of (a) AoIS, (b) GES, and (C) DES for two sub-systems depicted in a grid structure. The state matrices of the sub-systems are selected as $A_1 = 1.1$ and $A_2 = 1.3$ to capture the heterogeneity among control applications. The loss probabilities are selected to be equal, i.e., $p_1 = p_2 = 0.5$. Moreover, $\gamma = 0.9$, $M = 7$, and $\theta = 0.1$ have been used for the AoIS and DES policies. The dark squares correspond to states $\boldsymbol{s} = (\Delta_1, \Delta_2)$, in which the scheduler decides for the first user, i.e., $a = 1$. In those states colored in yellow, e.g., $\boldsymbol{s} = (1, 7)$, the second user is scheduled, i.e., $a = 2$.

into our model. However, using the success probability as a weighting factor in front of the actual cost function is a heuristic approach. Having said that, we provide the GES policy as follows:

$$\begin{aligned} &\max_{\pi_i \in \{0,1\}} \{\pi_i[t] \cdot p_i \cdot g(\Delta_i[t])\} \\ &\text{s.t.} \quad \sum_{i=1}^{N} \pi_i[t] \leq 1. \end{aligned} \tag{4.27}$$

The sub-system i maximizing the equation is scheduled for transmission, i.e., $a[t] = i$.

Fig. 4.7 depicts the considered policies, namely AoIS, GES, and DES, for a simplified setting. Specifically, the figures shows the stationary policy for two sub-systems differing in system matrices A_i. The policy in Fig. 4.7a is symmetrical, as AoIS is independent of the control system parameters and the packet success probabilities are identical. On the other hand, figures 4.7b and 4.7b show asymmetrical nature caused by the sub-systems' discrepancy in system stability.

4.2.4.3 Simulation Setup

For our numerical evaluation, we consider $N = 5$ scalar control sub-systems that differ in their system matrix as $\boldsymbol{A}_{1,2,3,4,5} = \{1.1, 1.3, 1.5, 1.7, 1.9\}$. The random packet loss probability characterizing the shared wireless channel is identical for each user, i.e., $p_i = 0.9, \forall i$. The feedback gain matrix is given as $\boldsymbol{L}_i = \boldsymbol{A}_i$ corresponding to the deadbeat control strategy. Moreover, the input and noise covariance matrices are selected equal as $\boldsymbol{B}_i = 1$ and $\boldsymbol{\Sigma}_i = 1$

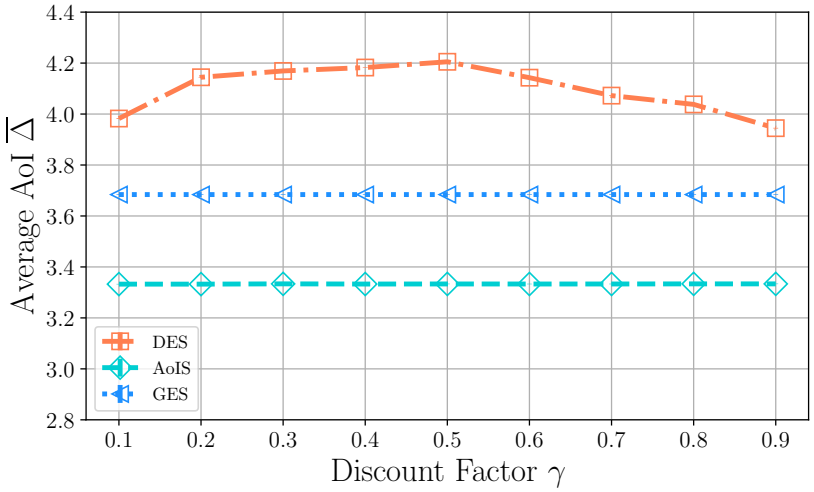


Figure 4.8: Long-term average age of information vs. discount factor γ for discounted error (DES), AoI (AoIS), and greedy error (GES) schedulers. Out of five control loops, only one sub-system is allowed to transmit simultaneously. Transmission success probability is equal for all sub-systems, i.e., $p_i = 0.9\ \forall i$. 95% confidence intervals are too small to be plotted.

for all sub-systems. We choose the augmentation level M as 25, to reduce the state space from infinite to finite length. The stopping condition for the value iteration algorithm presented in Alg. 2 is selected as $\theta = 0.1$.

Our simulative study is based on Monte Carlo experiments. Each simulation run consists of $T = 20\ 000$ time slots and is repeated 100 times. The performance comparison between the selected schedulers are made based on the long-term average MSE and the AoI per time slot calculated as:

$$\overline{MSE} = \frac{1}{T}\frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\|\boldsymbol{e}_i[t]\|^2 \tag{4.28}$$

$$\overline{\Delta} = \frac{1}{T}\frac{1}{N}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\Delta_i[t] \tag{4.29}$$

We employ $\overline{MSE}$ as the main metric quantifying the estimation performance and $\overline{\Delta}$ to capture the average freshness performance at the controllers. The discount factor is varied from 0.1 to 0.9 in discrete steps of 0.1, i.e., $\gamma = \{0.1, 0.2, \ldots, 0.9\}$ to study its effect on the selected KPIs. It is important to remind that as γ increases, the γ-optimal policy becomes more far-sighted and expected future costs play a more important role in current actions.

4.2.4.4 Numerical Results

Fig. 4.8 shows the average AoI in the network when AoIS, GES, and DES policies are utilized for resource allocation. $\overline{\Delta}$ is plotted against increasing values of the discount factor

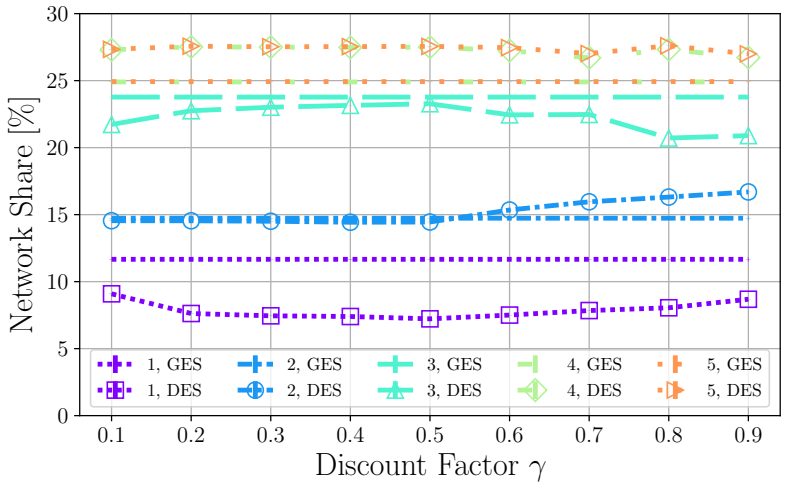


Figure 4.9: Illustration of network resources shares for different sub-system classes over 100 simulation runs. Numbering in the legend indicate the class i with the respective system matrices $A_{1,2,3,4,5} = \{1.1, 1.3, 1.5, 1.7, 1.9\}$. Lines with and without markers belonging to the same class i illustrate the average network resource share granted to i by the discounted (DES) and greedy (GES) error schedulers, respectively. 95% confidence intervals are too small to be displayed.

γ. We can observe that the AoIS outperforms the GES and DES in terms of information freshness at the receiver nodes. An interesting observation is the concave shape of the DES curve for the varying discount factor. In order to elaborate on this effect, we present Fig. 4.9 illustrating the average network shares of individual sub-systems given in percentages. A value of $\alpha_i\%$ indicates that the i-th system has been scheduled in $\alpha_i\%$ of all available transmission slots. The figure presents the network shares of individual classes $i \in \{1, \dots, 5\}$ with system matrices $\boldsymbol{A}_i = \{1.1, 1.3, 1.5, 1.7, 1.9\}$ for varying γ. The results give insights into the resource management and reveals that the discount factor γ effects each application class differently. For instance, an increase in γ from 0.1 to 0.5 does not have any effect on sub-systems two, four, and five. However, doing so prioritizes class three over class one application. As a result, the gap between the first and third feedback loops grows. Such a behavior leads to a reduced fairness w.r.t. the resource distribution and in return an increase in the average age. Further growth of γ beyond 0.5, however, reverses this behavior, and increases the prioritization and network share of the second application with $A_2 = 1.3$. Thereby, the distribution of resources becomes more balanced, hence the AoI decreases. On the other hand, the GES shows a more balanced distribution, which in return outperforms the DES in terms of information freshness, as visible in Fig. 4.8. Additionally, when the AoIS is selected, all sub-systems coincide at 20% regarding the network shares, as they are treated equally irrespective of their system dynamics. However, Fig. 4.8 does not include those curves to improve presentation.

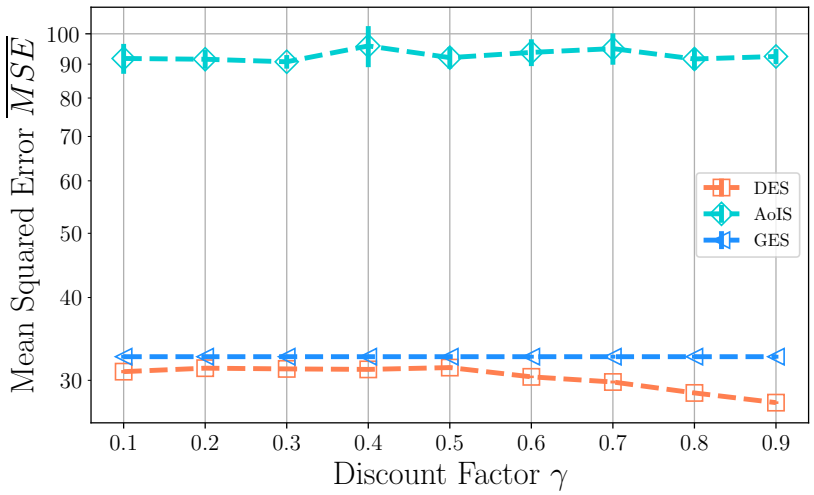


Figure 4.10: Long-term average mean squared error plotted against the discount factor γ. The results are based on discounted error (DES), greedy error (GES), and AoI (AoIS) scheduling policies. Vertical error bars represent 95% confidence intervals for Monte Carlo simulations with 100 repetitions.

Fig. 4.10 presents the estimation performance captured by the network-wide MSE, i.e., $\overline{MSE}$. The discount factor γ is varied along the x-axis. In alignment with our previous results from section 4.1, the AoIS, which is always optimal for the given γ value, is outperformed by the GES and DES policies. Note that the AoIS neglects the system dynamics and treats all users equally as long as channel conditions are identical. On the other hand, the DES outperforms the greedy scheduler for all γ values. Moreover, a more far-sighted policy achieved by increasing the discount factor helps with reducing the MSE further. Note that the scheduler becomes more far-sighted as γ increases and possible high costs in the future play a greater role in current decisions. Contrarily, when a small γ the GES is selected, the policy rushes for immediate rewards reducing worsening the estimation performance.

4.2.4.5 Selecting the Augmentation Level M

The augmentation level parameter M is one of the key design parameters for the proposed scheduling algorithm. In particular for a given number of users N, the cardinality of the state space $\boldsymbol{\mathcal{S}_M}$ equals to M^N increasing the computational complexity of the value iteration algorithm significantly. On the one hand, the scalability of the proposed approach suffers heavily from a largely selected M value. On the other, if a larger M is utilized, the ATAS approaches to the original problem and may potentially improve the performance. Put differently, the selection of the parameter M is a trade-off between complexity and performance. To that end, we vary the augmentation level M and obtain the stationary optimal policy as described in the algorithm 2. Doing so, we investigate the impact of M on the achieved performance.

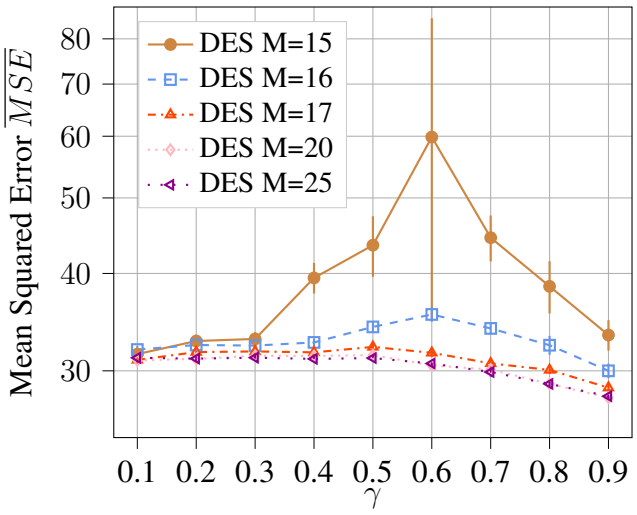


Figure 4.11: Achieved MSE plotted against the discount factor parameter γ when the discounted error scheduler (DES) is employed. Each curve belongs to one of the five selected augmentation levels $M \in \{15, 16, 17, 20, 25\}$. Vertical error bars represent 95% confidence intervals for Monte Carlo simulations with 100 repetitions.

Fig. 4.11 presents the $\overline{MSE}$ plotted against γ for five different values of M, namely, 15, 16, 17, 20, and 25. If we look at the $M = 15$ curve, we can observe that it achieves the worst performance among the considered values. This results from a less accurate cost functions due to a higher approximation level[6]. Consequently, an increase in M leads to a reduction in the MSE. Nevertheless, the performance gain diminishes after a certain M value, which is evident especially from the overlapping $M = 20$ and $M = 25$ curves. This shows that the growth in complexity through a higher M does not necessarily lead to a performance gain in the same magnitude.

In addition, we present the network shares of individual sub-systems i over γ for two of the selected values of M, i.e., $M = 15$ and $M = 25$. As visible in Fig. 4.12, both policies differing only in M parameter follow very similar trajectories. The largest deviation between the two curves can be observed for the first sub-system with $\boldsymbol{A}_i = 1.1$. Despite of the relatively small differences in, how the resources are distributed among sub-systems, the difference in the estimation performance is significant. This shows that although the performance gain diminishes after a certain point, the selection of M parameter should be selected carefully in order to capture the original problem fairly well.

[6]Since we limit the maximum cost to $g(\tilde{\Delta}^{(M)})$ for each sub-system i, the approximating finite sequence does not represent the original MDP, as accurate as, when a higher M is selected.

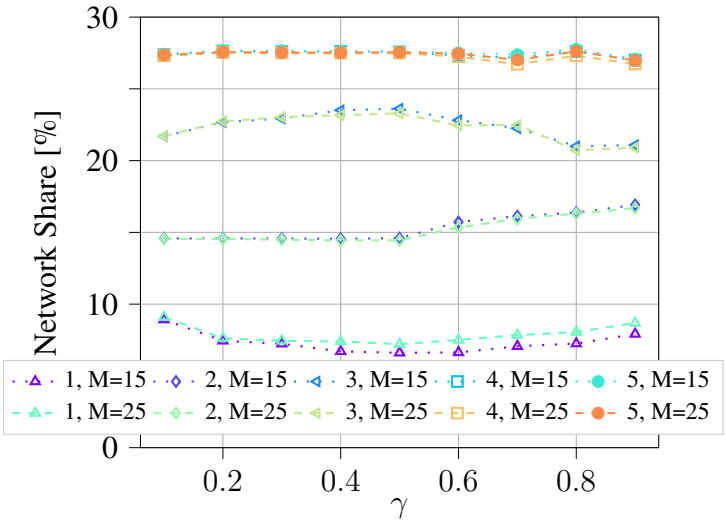


Figure 4.12: Illustration of the network shares for different sub-systems over 100 simulation runs. Numbering in the legend indicate the class i with the respective system matrices $A_{1,2,3,4,5} = \{1.1, 1.3, 1.5, 1.7, 1.9\}$. Identical markers belong to the same class i. Dotted and dashed lines illustrate the average network resource share granted to i by the discounted error scheduler (DES) with augmentation levels $M = 15$ and $M = 25$, respectively. 95% confidence intervals cannot be seen due to their small sizes.

4.3 Optimal Scheduling: The Dynamic Loss Case

In the previous section, we have focused on optimal stationary scheduling policies in a single-hop multi-user scenario. Each sub-system has been subject to a random packet loss, the probability of which was constant over time. In this section, we allow the packet loss to change dynamically while focusing again on a single-hop topology. As the infinite horizon problem formulation would be sub-optimal due to the changing transition probabilities within the MDP, we formulate the centralized scheduling problem as a Finite Horizon (FH) optimization problem. We solve it optimally using the dynamic programming framework from [Ber95].

4.3.1 System Model

We consider a network of N independent LTI feedback control loops sharing a single-hop wireless communication network. Each system consists of a co-located plant-sensor pair and a remotely operating controller. The plant, sensor, and controller of the i-th system are denoted by $\mathcal{P}_i$, $\mathcal{S}_i$, and $\mathcal{C}_i$, respectively. While the plant-to-sensor link is ideal, the sensor observations need to be transmitted over a wireless link to the controller in form of status update packets. We assume that each packet can carry at most a single state information.

Time is normalized to unity and the smallest time unit in our model is a time slot indexed by $t \in \mathbb{N}$. Each slot accommodates one transmission and every transmitted packet starting

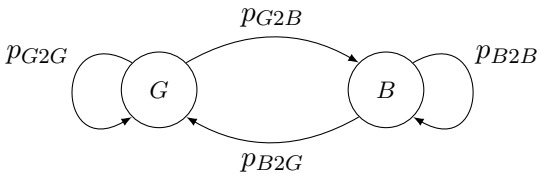


Figure 4.13: Gilbert-Elliott model with good (G) and bad (B) states and their state transition probabilities. In the good state, the packet transmissions along a communication link are more likely to be successful than when in the bad state.

in slot t ends within the same slot. Moreover, the access to the shared wireless medium is managed by a centralized scheduler. At the beginning of each time slot, the scheduler determines which sensor $\mathcal{S}_i$ is allowed to transmit in the same slot, whereas all other sensors except for i do not initiate any transmission. In other words, if we use a binary variable $\pi_i(t) \in \{0, 1\}$ indicating whether $\mathcal{S}_i$ is scheduled in time slot t, then a $\pi_i(t) = 1$ implies a scheduling decision for the i-th sub-system with $\sum_{i=1}^{N} \pi_i(t) = 1$. We assume a constant delay of one slot, i.e., any transmitted information during the time slot t is available at the controller at $t + 1$.

The link quality between each sensor-controller pair is dynamic and modeled in the form of time-varying packet loss probability $p_i(t) \in (0, 1)$. In particular, the channel between each $\mathcal{S}_i$ and $\mathcal{C}_i$ follow the Gilbert-Elliott (GE) model [Gil60], which is based on a two-state Markov chain, as depicted in Fig. 4.13. The state of each sensor-to-controller link, $\sigma_i \in \{G, B\}$, alternates between the *good (G)* and the *bad (B)* states, which define the current packet loss probability between $\mathcal{S}_i$ and $\mathcal{C}_i$. That is, when in G, the loss probability is lower than the one in B, i.e., $p_G < p_B$. As a result, the failure probability of a transmission in time slot t can be characterized as follows:

$$p_i(t) = \begin{cases} p_G & \text{, if } \sigma_i(t) = G, \\ p_B & \text{, if } \sigma_i(t) = B. \end{cases} \tag{4.30}$$

The transition from the good state to the bad state occurs with a stationary probability p_{G2B}. Similar, p_{B2G} denotes the probability for $\sigma_i(t)$ to switch from the G state to the B state, i.e.:

$$p_{G2B} \triangleq \Pr\left[\sigma_i(t+1) = B \mid \sigma(t) = G\right], \forall i, t, \tag{4.31}$$

$$p_{B2G} \triangleq \Pr\left[\sigma_i(t+1) = G \mid \sigma(t) = B\right], \forall i, t. \tag{4.32}$$

It is important to mention that each link behaves independently and the random variables σ_i and σ_j with $i \neq j$ are not correlated. The probabilities of staying in the same state are given as $p_{G2G} \triangleq 1 - p_{G2B}$ and $p_{B2B} \triangleq 1 - p_{B2G}$. The GE model has been widely used in the literature to model packet loss in real-time networks, e.g., [HH08; Bil+15]. Despite its simplicity,

the two-state GE model has been shown to be fairly accurate to represent Rayleigh-fading channels when then channel quality does not vary dramatically over time [SHD98; WC96][7].

Given that $\mathcal{S}_i$ is scheduled, $\delta_i(t) = 1$ and $\delta_i(t) = 0$ indicate a successful and a failed transmission, respectively:

$$\Pr[\delta_i(t) = 0 \mid \pi_i(t) = 1] = p_i(t) \tag{4.33}$$

$$\Pr[\delta_i(t) = 1 \mid \pi_i(t) = 1] = 1 - p_i(t). \tag{4.34}$$

On the other hand, for all users that have not transmitted in slot t, by definition, the chance of a successful reception is zero, i.e.:

$$\Pr[\delta_i(t) = 1 \mid \pi_i(t) = 0] = 0. \tag{4.35}$$

We assume that control loops operate slower than the network. That is, each sensor $\mathcal{S}_i$ observes the system state periodically once in every $D_i \in \mathbb{Z}^+$ slots. We call these periodic instances a *sampling event*, at which a status update packet carrying the latest state information is generated. Additionally, the time between two consecutive sampling events is called a *sampling period*. Let $t_{i,o}$ denote the time of the first sampling event of the i-th sub-system that is selected uniformly in the half-open interval $[0, D_i]$[8]. Hence, the set of time slots, in which a sampling event occurs is defined as:

$$\mathcal{G}_i \triangleq \{t_{i,o}, t_{i,o} + D_i, t_{i,o} + 2D_i, \dots\}. \tag{4.36}$$

The generation of packets is periodic and independent of the communication network. However, due to resource constraints, some of these packets may not get the chance to be transmitted before the next sampling event. In addition, having received an update, the controller does not benefit from receiving an older observation, since older data are considered to be obsolete and non-informative according to our system model. In consequence, the sensors discard any older packet in their transmission upon the generation of a new status information. Hence, each sensor stores only the most recent state of the plant process until the subsequent sampling event. To characterize this mathematically, let us introduce a variable $\nu_i^{\mathcal{S}_i}(t) \in \mathcal{G}_i$ that denotes the generation time of the most recent information at $\mathcal{S}_i$. The following equation states the relationship between the sampling events and $\nu_i^{\mathcal{S}_i}(t)$:

$$\nu_i^{\mathcal{S}_i}(t) = \begin{cases} t & \text{, if } t \in \mathcal{G}_i, \\ \nu_i^{\mathcal{S}_i}(t-1) & \text{, otherwise.} \end{cases} \tag{4.37}$$

[7]For scenarios, in which the channel quality varies dramatically over time, a finite-state Markov channel with more than two states is recommended [WM95].

[8]$t_{i,o} \in \mathbb{N}$ can be interpreted as the time offset initiating the operation of control applications. As we allow $t_{i,o}$ to be different for each sub-system, we do not assume any synchronization between the sampling events of any two different users.

The behavior of $\nu_i^{\mathcal{S}_i}(t)$ is similar to the Zero-Order-Hold (ZOH) concept used to represent discrete-time signals in continuous-time. Note that those slots in the set $\mathcal{G}_i$ are the only ones, in which the information at $\mathcal{S}_i$ is updated.

In the case of a successful transmission, the controller is supplied with the latest information available at the sensor. If we use $\nu_{i,r}^{\mathcal{C}_i}(t)$ to denote the time stamp of the latest information that has been successfully transmitted over the wireless link, it can be characterized by:

$$\nu_{i,r}^{\mathcal{C}_i}(t+1) = \begin{cases} \nu_i^{\mathcal{S}_i}(t) & \text{, if } \delta_i(t) \cdot \pi_i(t) = 1, \\ \nu_{i,r}^{\mathcal{C}_i}(t) & \text{, otherwise.} \end{cases} \tag{4.38}$$

Note the delay of one transmission slot in the equation above.

We assume that any new information received by the controller can be utilized for the subsequent sampling period at the earliest. That is, once the controller determines the control input for the k_i-th sampling period, it cannot be changed further before the next sampling event. Having received a new packet over the wireless link, the controller stores it until the next sampling event at which it decides for the following control input $\boldsymbol{u}_i[k_i+1]$. Therefore, if we denote the generation time of the latest utilized packet by $\mathcal{C}_i$ as $\nu_{i,u}^{\mathcal{C}_i}(t)$, its time evolution is given as:

$$\nu_{i,u}^{\mathcal{C}_i}(t+1) = \begin{cases} \nu_{i,r}^{\mathcal{C}_i}(t+1) & \text{, if } t+1 \in \mathcal{G}_i, \\ \nu_{i,u}^{\mathcal{C}_i}(t) & \text{, otherwise.} \end{cases} \tag{4.39}$$

The distinction between $\nu_{i,r}(t)$ and $\nu_{i,u}(t)$ is necessary, because the controller may receive new packets during a sampling period that affect neither the control input nor the system state before the following sampling period. In Fig. 4.14, we illustrate the relationship between $\nu_i^{\mathcal{S}_i}(t)$, $\nu_{i,r}^{\mathcal{C}_i}(t)$, and $\nu_{i,u}^{\mathcal{C}_i}(t)$ with a brief example. In the figure, the controller receives two status update packets successfully at t_1 and t_2 causing the equalization of $\nu_i^{\mathcal{S}_i}$ and $\nu_{i,r}$. However, as $\nu_{i,u}^{\mathcal{C}_i}$ can only be modified at each sampling event, the $\nu_{i,u}^{\mathcal{C}_i}$ follows $\nu_{i,r}^{\mathcal{C}_i}$ with a certain delay. Note that $\nu_i^{\mathcal{S}_i}$ shows a staircase behavior as the information at the sensor is updated at each sampling event through a new observation of the system state.

We represent the behavior of the i-th control sub-system by the following LTI model:

$$\boldsymbol{x}_i[k_i(t)+1] = \boldsymbol{A}_i\boldsymbol{x}_i[k_i(t)] + \boldsymbol{B}_i\boldsymbol{u}_i[k_i(t)] + \boldsymbol{w}_i[k_i(t)], \tag{4.40}$$

with[9]:

$$k_i(t) = \left\lfloor \frac{t - t_{i,o}}{D_i} \right\rfloor. \tag{4.41}$$

[9]For $t < t_{i,o}$, $k_i(t)$ takes a negative value. We allow this since $t_{i,o}$ defines the initialization of sub-systems and the system behavior before $t_{i,o}$ is not taken into account for the remaining analysis.

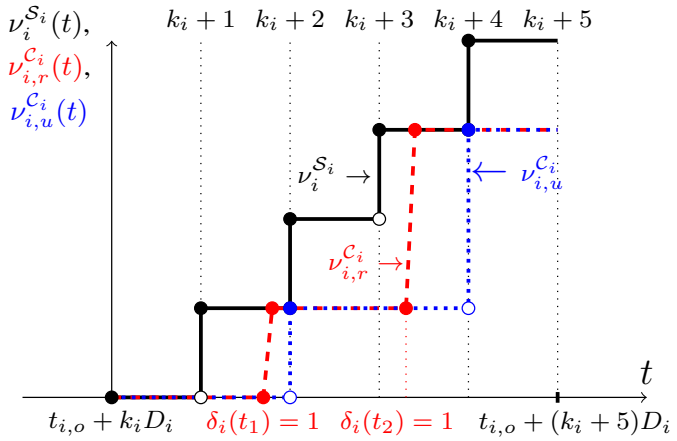


Figure 4.14: Evolution of generation time $\nu_i^{\mathcal{S}_i}(t)$, received time $t_{i,r}(t)$, and the update time $\nu_{i,u}^{\mathcal{C}_i}(t)$ depicted in y-axis versus time in x-axis. $\nu_i^{\mathcal{S}_i}(t)$ and $\nu_{i,u}^{\mathcal{C}_i}(t)$ are updated periodically every D_i slots, while $\nu_{i,r}^{\mathcal{C}_i}(t)$ can be updated asynchronously. On the x-axis with $\delta_i(t_{1,2})=1$, two cases of successful packet transmission for sub-system i are depicted. Note that while $\nu_{i,r}^{\mathcal{C}_i}(t)$ changes subsequent to the reception of any new information, $\nu_{i,u}^{\mathcal{C}_i}(t)$ follows $\nu_{i,r}^{\mathcal{C}_i}(t)$ only at the instances of sampling events.

The role of $k_i(t)$ is providing the transition from slot timing into sampling period, since their granularity differs. In other words, while t is incremented by one every time slot, $k_i(t)$ is incremented every D_i slots. Furthermore, $k_i(t)$ tells us, in which sampling period the sub-system currently is. At the beginning of each sampling period $k_i(t)$, i.e., at $t=t_{i,o}+k_i(t)D_i$, the controller $\mathcal{C}_i$ obtains the control input $\boldsymbol{u}_i[k_i(t)]$ based on the available observation history. We omit the continuous time notation in $k_i(t)$ and use the shorthand notation k_i for brevity in the following analysis.

The discrete-time model enables the progress of the control and communication in different rate. Namely, in control, changes in system state occur at sampling events[10]. Doing so,it can be argued that from the controller's perspective, the information of interest ages in discrete steps and by one every D_i slots. Thus, we define the AoI in units of sampling periods. As a result, the AoI in any slot t is normalized by the sampling period D_i and can be obtained from the following equation:

$$\Delta_i(t)=\left\lfloor \frac{t-\nu_{i,u}^{\mathcal{C}_i}(t)}{D_i} \right\rfloor. \tag{4.42}$$

It is important to point out that as the packets are received at least one slot delayed, they can first be utilized at least D_i slots after their generation. Therefore, the nominator of (4.42) can not be less than D_i according to our network model. As a result, the minimum AoI in our considered scenario is one, i.e., $\Delta_i(t)\geq 1, \forall i,t$.

[10]We assume that the sampling period of each sub-system is selected small enough such that the changes between consecutive sampling events are negligible.

Due to the presence of packet losses and resource scarcity, the controller cannot be updated during every sampling period. Instead, the controller estimates the system state remotely based on the LTI model and the most recent observation it has received. Let $\boldsymbol{x}_i[k_i - \Delta_i[k_i]]$ be the most recent knowledge at $\mathcal{C}_i$ that is $\Delta_i[k_i]$ sampling periods old. Similar to (3.15), we can derive the conditional expectation of the system state as:

$$\hat{\boldsymbol{x}}[k_i] = \boldsymbol{A}_i^{\Delta_i[k_i]} \boldsymbol{x}_i[k_i - \Delta_i[k_i]] + \sum_{q=1}^{\Delta_i[k_i]} \boldsymbol{A}_i^{q-1} \boldsymbol{B}_i \boldsymbol{u}_i[k_i - q]. \tag{4.43}$$

As a result, the expected MSE at $\mathcal{C}_i$ is expressed by:

$$\mathbb{E}\left[\left(\boldsymbol{e}_i[k_i]\right)^T \boldsymbol{e}_i[k_i] \right] = \sum_{r=0}^{\Delta_i[k_i]-1} \mathsf{tr}\left(\left(\boldsymbol{A}_i^T\right)^r \left(\boldsymbol{A}_i\right)^r \boldsymbol{\Sigma}_i \right), \tag{4.44}$$

as before.

4.3.2 Problem Statement

Our goal is propose a scheduler that maximizes the total estimation performance in the network by minimizing the MSE. We are interested in the calls of scheduling policies $\boldsymbol{\pi}$ that consist of a sequence of scheduling decisions $a(\tau)$ for the next $H \in \mathbb{Z}^+$ transmission slots, i.e., $\boldsymbol{\pi} = \{a(t), \ldots, a(t+H-1)\}$. As in the previous section, $a(t) = i$ implies that $\pi_i(t) = 1$ and $\pi_j(t) = 0$ for all $j \neq i$. H is the FH parameter that defines how many future time slots is taken into account, while deciding for the current scheduling action. Therefore, the selection of H controls the far-sightedness of the FH scheduling policy.

Let us define a vector $\boldsymbol{s} \in \mathbb{Z}^{3N}$ representing the network state in terms of $\nu_i^{\mathcal{S}_i}$, $\nu_{i,r}^{\mathcal{C}_i}$, and $\nu_{i,u}^{\mathcal{C}_i}$:

$$\boldsymbol{s}(t) \triangleq \begin{bmatrix} \boldsymbol{\nu}_s(t) \\ \boldsymbol{\nu}_r(t) \\ \boldsymbol{\nu}_u(t) \end{bmatrix} \tag{4.45}$$

with:

$$\boldsymbol{\nu}_s(t) \triangleq \begin{bmatrix} \nu_1^{\mathcal{S}_1}(t) & \ldots & \nu_N^{\mathcal{S}_N}(t) \end{bmatrix}^T \tag{4.46}$$

$$\boldsymbol{\nu}_r(t) \triangleq \begin{bmatrix} \nu_{1,r}^{\mathcal{C}_1}(t) & \ldots & \nu_{N,r}^{\mathcal{C}_N}(t) \end{bmatrix}^T \tag{4.47}$$

$$\boldsymbol{\nu}_u(t) \triangleq \begin{bmatrix} \nu_{1,u}^{\mathcal{C}_1}(t) & \ldots & \nu_{N,u}^{\mathcal{C}_N}(t) \end{bmatrix}^T \tag{4.48}$$

Each of $\boldsymbol{\nu}_s$, $\boldsymbol{r}$, and $\boldsymbol{u}$ are of length N. Next, we define the set of feasible scheduling actions given the network state as:

$$\mathcal{A}(\boldsymbol{s}(t)) = \{\varnothing\} \cup \{i : \nu_i^{\mathcal{S}_i}(t) > \nu_{i,r}^{\mathcal{C}_i}(t)\} \tag{4.49}$$

The equation above implies that the action set is constrained by filtering out those sub-systems that do not have any new data to transmit, i.e., when the most recent observation has been transmitted over the wireless link to the controller. The empty action represents the case, in which none of the sub-systems is eligible for transmission.

Given the network state $\boldsymbol{s}(t)$ and the scheduling decision $a(t)$, the transition probability to a next state $\boldsymbol{s'}(t+1)$ strictly depends on the instantaneous condition of the wireless link, i.e., $p_i(t)$. Let us consider a toy example with $\boldsymbol{\nu}_s(t) = [3 \quad 5]^T$ and $\boldsymbol{\nu}_r(t) = [3 \quad 1]^T$. As the controller of sub-system $i = 1$ has already received the system state that has been generated at $t = 3$, the scheduler does not consider $a(t) = 1$ as a feasible action. On the other hand, the second sensor has a fresher measurement than the one present at $\mathcal{C}_2$. Therefore, the scheduler decides for $a(t) = 2$. Once the action is taken, the transmission can either be successful with a probability of $1 - p_2(t)$ or fail with $p_2(t)$. In the success case, the vector $\boldsymbol{\nu}_r(t+1)$ becomes $\boldsymbol{\nu}_r(t+1) = [3 \quad 5]^T$. In the case of a packet loss, the controller is not updated and it holds that $\boldsymbol{\nu}_r(t+1) = \boldsymbol{\nu}_r(t)$. By removing the action $a(t) = 1$ from $\mathcal{A}(\boldsymbol{s}(t))$, scheduling decision for the first user, we reduce the set of feasible actions without losing optimality. As we are going to show later, this enables us to reduce our search space and improve the computational complexity of the considered problem.

Our goal is to minimize the total MSE in the network by choosing the best possible scheduling actions at each time slot. Let $C(\boldsymbol{s}(t))$ define the total MSE in the network:

$$C(\boldsymbol{s}(t)) \triangleq \mathbb{E}\left[\sum_{i=1}^{N} (\boldsymbol{e}_i[k_i(t)])^T \boldsymbol{e}_i[k_i(t)]\right]. \tag{4.50}$$

One can also interpret $C(\boldsymbol{s}(t))$ as the state cost that is incurred by visiting the state $\boldsymbol{s}(t)$. As a result, given a scheduling policy $\boldsymbol{\pi}$ mapping a network state to a scheduling decision, the expected *finite horizon cost* $\mathcal{J}_{\boldsymbol{\pi}}(\boldsymbol{s}(t))$ can be given as:

$$\mathcal{J}_{\boldsymbol{\pi}}(\boldsymbol{s}(t)) \triangleq \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{\tau=t}^{t+H} C(\boldsymbol{s}(t))\right]. \tag{4.51}$$

The subscript $\boldsymbol{\pi}$ in (4.51) indicates that the expectation is taken when the scheduling policy $\boldsymbol{\pi}$ is employed over the horizon H. Hence, the optimal policy $\boldsymbol{\pi}^*$ is the one minimizing (4.51), i.e.:

$$\mathcal{J}_{\boldsymbol{\pi}^*}(\boldsymbol{s}(t)) = \min_{\boldsymbol{\pi}} \mathcal{J}_{\boldsymbol{\pi}}(\boldsymbol{s}(t)) = \mathcal{J}^*(\boldsymbol{s}(t)). \tag{4.52}$$

Throughout the following analysis, we refer to the optimization problem in (4.52) as the H-stage problem and drop the subscript $\boldsymbol{\pi}$ for brevity.

It is important to emphasize that the H-stage problem considers all possible future states and costs that can be incurred throughout the considered time horizon, i.e., $[t, t+H]$. However,

any state that is unreachable in this time window is irrelevant to the optimization problem. On the contrary, the infinite horizon problem presented in section 4.2 incorporates all states and their respective costs are incorporated into the decision-making through the value iteration technique. We can say that while the H-stage problem is an optimization problem terminating in H time slots from "now", the infinite horizon problem looks at an infinitely long time period.

4.3.3 Finite Horizon Scheduler

We propose a FH scheduler that schedules the "best" user by solving the finite horizon optimization problem. In [Ber95, p. 25] it has been shown that the optimal cost $\mathcal{J}^*(\boldsymbol{s}(t))$ can be obtained by minimizing the RHS of the H-stage problem (4.52) starting at $\boldsymbol{s}(t)$, i.e.:

$$\mathcal{J}_\tau(\boldsymbol{s}(\tau)) = \min_{a(\tau)\in\mathcal{A}(\boldsymbol{s}(\tau))} \mathbb{E}\left[C(\boldsymbol{s}(\tau)) + \mathcal{J}_{\tau+1}(\boldsymbol{s}(\tau+1))\right] \quad , \tau \in \{t, t+1, \ldots, t+H-1\}, \tag{4.53}$$

with the terminal cost given as:

$$\mathcal{J}_{\tau+H}(\boldsymbol{s}(t+H)) = C(\boldsymbol{s}(t+H)). \tag{4.54}$$

The subscript τ of $\mathcal{J}_\tau$ is used to emphasize that the cost is associated with a state $\boldsymbol{s}(\tau)$ that appears at the τ-th stage of the H-stage problem[11].

One can find the optimal cost $\mathcal{J}^*(\boldsymbol{s}(t))$, as well as the optimal policy $\boldsymbol{\pi}^*$ by iterating backwards in time from stage H to zero. As shown in [Ber95], if the optimal action $a^*(\tau)$ minimizing the RHS of (4.53) is taken at each state appearing within the H stages of the FH problem, the policy $\boldsymbol{\pi}^*$ defining the best action at every $\boldsymbol{s}(\tau)$ over the FH is optimal.

The H-stage problem can be modeled as a tree structure, while each node represents a network state $\boldsymbol{s}(\tau)$ occurring at stage $t \leq \tau \leq t+H$. The nodes occurring in the same slot form a level together, whereas the root node constitutes the 0-th level of the tree by itself. Note that the root node corresponds to the current network state $\boldsymbol{s}(t)$. The nodes that appear in the last, i.e., H-th, level are called *leaf nodes*.

The backwards iteration to solve the H-stage problem can be viewed as visiting all levels of the tree starting from the leaf nodes and taking the optimal action at each node that minimizes the expected future costs. It is important to mention that while a level τ is being visited the cost of all possible next states from the next level have already been assigned the value obtained from (4.53). Hence, the operation of the FH scheduler can be summarized by the following steps:

1. Initialize the current state $\boldsymbol{s}(t)$ as the root of the tree structure.

[11] Each stage corresponds to a time slot in our model.

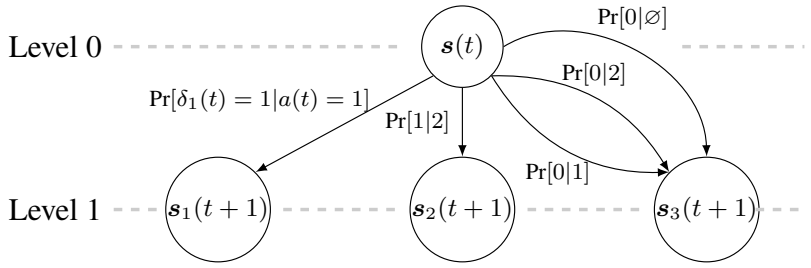


Figure 4.15: An example tree structure with two sub-systems for $H = 1$. Each edge is labeled with the corresponding transition probability, i.e., $\Pr[\delta_i(t) \mid a(t)]$. The states $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ stand for the success cases, in which the scheduled sub-system updates the controller successfully. On the other hand, $\boldsymbol{s}_3$ represents the failure case, in which none of the controllers is provided with a new piece of information.

2. Starting from the root node, determine the feasible actions at each node, $\mathcal{A}(\boldsymbol{s}(\tau))$ for $t \leq \tau < t + H$ and subsequently all possible next states $\boldsymbol{s}(\tau + 1)$ given an action $a(\tau)$.

3. Add all possible next states as child nodes to the next level with the corresponding transition probabilities from the parent node.

4. Repeat steps (2)-(4) until the H-th level of the tree has been constructed.

5. Assign costs to all states starting from the leaf nodes as in (4.53) and (4.54).

Let us consider a toy example with two sub-systems sampled every time slot, i.e., $D_i = 1$ and $\nu_i^{\mathcal{S}_i}(t) = t, \forall i, t$. Having the sampling frequency equal to the slot frequency, is a special case and implies $\nu_{i,r}^{\mathcal{C}_i}(t)$ and $\nu_{i,r}^{\mathcal{C}_i}(t)$ for all i and t. Fig. 4.15 shows a single-level tree constructed by the FH scheduler. The root node is the current network state $\boldsymbol{s}(t)$. $\boldsymbol{s}_1(t+1)$, $\boldsymbol{s}_2(t+1)$, and $\boldsymbol{s}_3(t+1)$ are possible next states to follow $\boldsymbol{s}(t)$. Note that $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, and $\boldsymbol{s}_3$ are also the leaf nodes of the tree structure, as $H = 1$ is assumed. Furthermore, the edges between the states are labeled with the corresponding transition probabilities that are written as the conditional probabilities given a scheduling decision. If we assume the initial state to be $\boldsymbol{s}(t) = [t \quad t \quad a \quad b \quad a \quad b]^T$, the possible next states can be given as:

$$\boldsymbol{s}_1(t+1) = [t+1 \quad t+1 \quad t \quad b \quad t \quad b]^T \tag{4.55}$$

$$\boldsymbol{s}_2(t+1) = [t+1 \quad t+1 \quad a \quad t \quad a \quad t]^T \tag{4.56}$$

$$\boldsymbol{s}_3(t+1) = [t+1 \quad t+1 \quad a \quad b \quad a \quad b]^T \tag{4.57}$$

For the sub-system with a successful transmission, the timestamp of the received information is updated, i.e., $\nu_{i,r}^{\mathcal{C}_i}(t+1) = \nu_{i,u}^{\mathcal{C}_i}(t+1) = \nu_i^{\mathcal{S}_i}(t) = t$. For those sub-systems that are either not scheduled or fail to update the destination despite of being scheduled in t, $\nu_{i,r}^{\mathcal{C}_i}(t+1)$ and $\nu_{i,u}^{\mathcal{C}_i}(t+1)$ remain unchanged relative to the previous slot.

Control Application Parameters					
Class 1	$\boldsymbol{A}^{(1)} = 1.1$	$\boldsymbol{B}^{(1)} = 1.0$	$\boldsymbol{L}^{(1)} = 1.1$	$\boldsymbol{\Sigma}^{(1)} = 1.0$	$D^{(1)} = 3$
Class 2	$\boldsymbol{A}^{(2)} = 1.2$	$\boldsymbol{B}^{(2)} = 1.0$	$\boldsymbol{L}^{(2)} = 1.2$	$\boldsymbol{\Sigma}^{(2)} = 1.0$	$D^{(2)} = 3$
Class 3	$\boldsymbol{A}^{(3)} = 1.3$	$\boldsymbol{B}^{(3)} = 1.0$	$\boldsymbol{L}^{(3)} = 1.3$	$\boldsymbol{\Sigma}^{(3)} = 1.0$	$D^{(3)} = 3$
Class 4	$\boldsymbol{A}^{(4)} = 1.4$	$\boldsymbol{B}^{(4)} = 1.0$	$\boldsymbol{L}^{(4)} = 1.4$	$\boldsymbol{\Sigma}^{(4)} = 1.0$	$D^{(4)} = 3$

Table 4.1: Selected parameters of our four considered control application classes. The superscript indicates the class index and should not be confused with matrix power.

Having the entire tree constructed as described previously, the scheduler assigns state cost to each state starting from the leaf nodes. Once the algorithm completes, it obtains the optimal scheduling policy $\boldsymbol{\pi}^*$ defining the best action $a^*(t)$ at the root node. Consequently, the optimal action is executed by the FH scheduler. It is important to emphasize that, although the scheduler looks into H future slots and obtains the optimal action for each possible future state, the steps (1)-(5) have to be repeated every time slot. This is because the root node, as well as packet the loss probabilities alter.

The complexity of the FH scheduling algorithm is dictated by the number of nodes in the tree structure. In particular, if all control loops are sampled as fast as the slot frequency, as in our toy example, the resulting tree structure consists of $^{((N+1)^{H+1}-1)}/_{N}$ nodes. This represents the worst-case scenario in terms of computational complexity. In the Big-$\mathcal{O}$ notation, the scheduling algorithm's complexity can be expressed by $\mathcal{O}(N^H)$. Increasing the sampling period beyond one reduces the number of nodes, because not having any new information to transmit after updating the destination already with the most recent one, reduces the action space. Put differently, if $\nu_i^{\mathcal{S}_i}(t) = \nu_{i,r}^{\mathcal{C}_i}(t)$, meaning that the destination has already the most recent information about the system state, then the scheduler does not consider that particular user within its action set, i.e., $i \notin \mathcal{A}(\boldsymbol{s}(t))$. Thereby, the number of nodes becomes less than the worst-case.

4.3.4 Numerical Evaluation

4.3.4.1 Simulation Details

In order to evaluate the performance of our proposed scheduler, we conduct a numerical evaluation based on Monte Carlo experiments. To that end, we consider four classes of applications, each having a different system matrix as given in Tab. 4.1. The difference in system matrices lead to different open-loop instabilities, thus represent a heterogeneous task-criticality among sub-systems, with class 4 being the most challenging one. Note that the

feedback gain matrix $\boldsymbol{L}^{(\cdot)}$ is chosen equal to the system matrix, corresponding the deadbeat control strategy[12].

We simulate eight feedback control loops, i.e., $N = 8$ for $T = 20000$ time slots. Each application class is represented by two sub-systems, i.e., $\boldsymbol{A}_1 = \boldsymbol{A}_2 = \boldsymbol{A}^{(1)}$ and $\boldsymbol{A}_3 = \boldsymbol{A}_4 = \boldsymbol{A}^{(2)}$, etc. The control loops are initialized with $\boldsymbol{x}_i[0] = \boldsymbol{w}_i[0]$ with the system noise following $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_i)$. In addition, the arrival time of the first sampling event is selected randomly from a discrete uniform distribution as $t_{i,o} = \mathcal{U}\{0, D_i - 1\}$ such that the control loops operate in an asynchronous fashion. The Gilbert-Elliott channel model is characterized by $p_G = 0.2$, $p_B = 0.6$, $p_{G2B} = 0.1$, and $p_{B2G} = 0.2$. While the considered model applies to all sub-systems in the network, their individual channel states $\sigma_i(t)$ are not synchronized.

The FH parameter H is varied from one to six to investigate the impact of making the FH scheduler more far- or short-sighted on the estimation performance. Each configuration characterized by the selected H is repeated 200 times. Similar to the previous sections, the estimation performance is captured by the long-term average MSE in the network, i.e.:

$$\overline{MSE} = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=0}^{T-1}(\boldsymbol{e}_i(t))^T\boldsymbol{e}(t). \tag{4.58}$$

The average AoI performance is defined analogously and captures the information freshness.

4.3.4.2 Results

Fig. 4.16 shows the long-term average MSE achieved by the FH scheduler for $H \in \{1, 2, \ldots, 5\}$. The figure illustrates $\overline{MSE}$ obtained as in (4.58) together with the mean MSE of individual control application classes. We observe a drastic reduction in $\overline{MSE}$ as we increase the FH parameter H from one to two. However, beyond $H = 3$, despite of increasing far-sightedness of the scheduler, the performance gain diminishes. The mean MSE values for each H are given in the table below.

Horizon H	1	2	3	4	5
$\overline{MSE}$	13.72	6.64	6.08	5.96	5.94
Improvement in %	-	51.58 %	8.40 %	2.06 %	0.27 %

The second row contains the $\overline{MSE}$ improvement in percentages relative to the previous H parameter, i.e., $H = 2$ relative to $H = 1$, $H = 3$ relative to $H = 2$, etc. It is important to mention that for higher values of H, the contribution of individual application classes, tend to meet. This effect follows from equal weighting of MSE among sub-systems while obtaining

[12]This is equivalent to assuming $\boldsymbol{Q}_i = 1.0$ and $\boldsymbol{R}_i = 0$ for all $\mathcal{C}_i$, neglecting the penalty for control effort. By solving the DARE from (2.17), we obtain $\boldsymbol{L}_i = \boldsymbol{A}_i$ for the optimal feedback gain matrix.

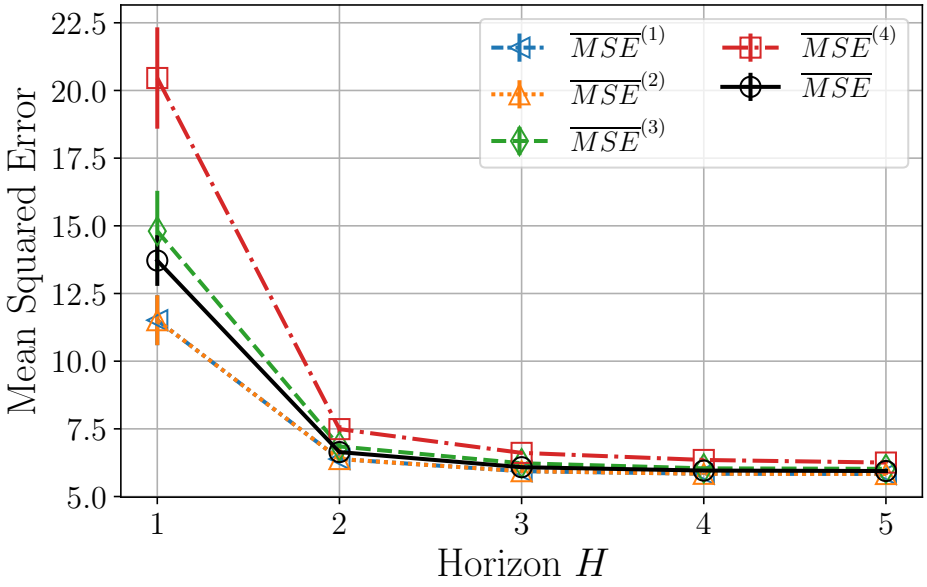


Figure 4.16: Long-term average MSE for increasing H. While the solid line, i.e., $\overline{MSE}$ represents the average performance of the entire network, the dashed lines belong to the considered control application classes. The vertical bars represent 95% confidence interval.

the cost in (4.20) which in turn leads to equal long-term averages as the scheduling actions become more "long-term optimal".

Fig. 4.17 presents the resulting performance w.r.t. information freshness. As one can see from the figure, the long-term average AoI is different for each application class. This is an expected result, as the proposed scheduling considers the MSE as the state cost and not the instantaneous AoI of sub-systems. This leads to an unbalanced distribution of resources due to the heterogeneous task-criticalities of systems. For instance, the sub-systems from the fourth class achieve a lower AoI than those of class one, i.e., $\overline{\Delta}^{(4)} < \overline{\Delta}^{(1)}$. This shows that less stable systems have been scheduled more frequently than more stable ones. In addition, we observe from the figure that as H increases, $\overline{\Delta}^{(3)}$ and $\overline{\Delta}^{(4)}$ decrease. This is caused by the increasing foresight of the scheduler and ability to prevent high future costs that may follow due to multiple consecutive failed transmissions. It is important to mention that although the sub-systems from the fourth class have received more resources than other application classes, their contribution to the network-wide $\overline{MSE}$ is still the highest.

The figure clearly shows that the distribution of network resources among different application classes changes for varying H. Moreover, the estimation performance depicted in Fig. 4.16 is affected by the scheduling decisions. However, the performance gain beyond $H = 3$ is not significant despite the exponentially increasing computational complexity. The following table shows the number of tree nodes capturing the complexity together with the worst-case bound:

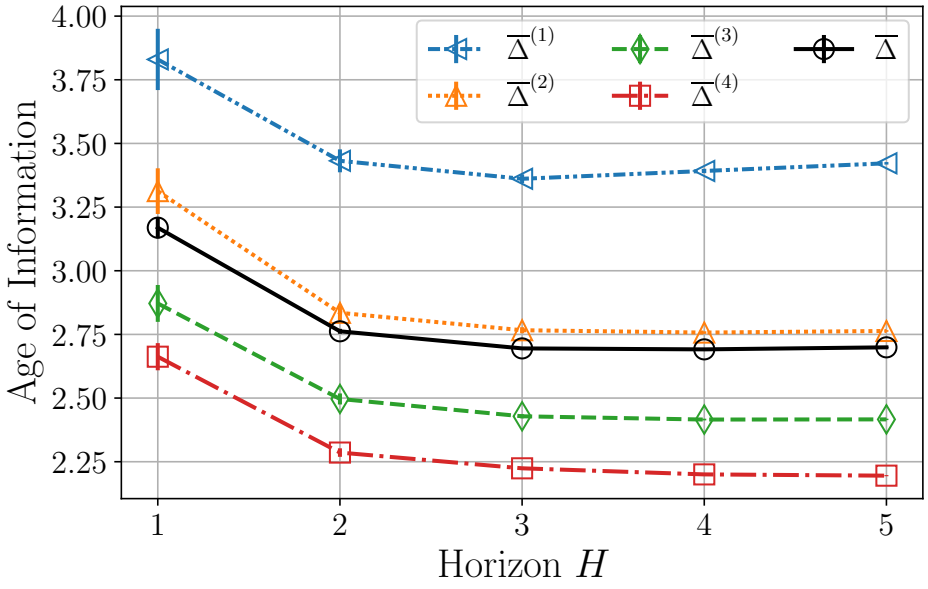


Figure 4.17: Long-term average AoI for increasing H. While the solid line, i.e., $\overline{\Delta}$ represents the average performance of the entire network, the dashed lines belong to the considered control application classes. The vertical bars represent 95% confidence interval.

Horizon H	1	2	3	4	5
FH	9.96	84	678	5451	43840
Worst-case	10	91	820	7381	66430

Table 4.2: Comparison of the number of tree nodes in worst-case scenario to the measured average in simulations.

The difference between the first and second rows is due to the constrained action space, which narrows down the algorithm's search space. This shows that obtaining the optimal policy becomes significantly more costly w.r.t. the complexity. However, the performance does not improve in the same order of magnitude beyond a certain point[13]. Nevertheless, such a trade-off between the complexity and performance can be found by controlling the H parameter.

4.4 Joint Uplink & Downlink Scheduling of Wireless Networked Control Systems

In the previous section, we have introduced a mathematical framework that formulates the centralized resource allocation for WNCS closed over a single-hop network as a finite horizon

[13]Note that this particular point is strictly scenario dependent.

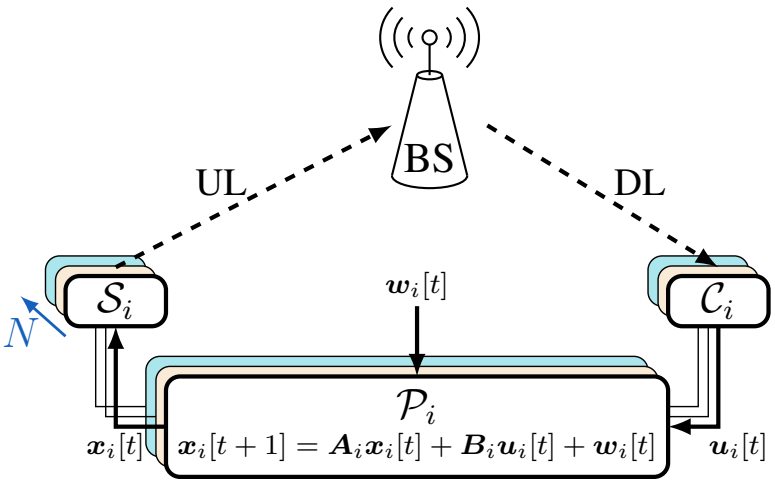


Figure 4.18: N linear time invariant (LTI) control systems are closed over a shared star network. The UL and DL transmissions occur according to a transmission schedule determined by the BS. While the solid connectors represent ideal links, the wireless links prone to packet loss and delay are depicted by dashed connectors.

optimization. We have shown that as the scheduling algorithm considers a longer time horizon into the future, the scheduling algorithm performs better that comes with the cost of computational complexity. In this section, we introduce a second wireless hop, which adds a DL to the previously considered UL scenario, hence making the proposed algorithm very applicable in cellular networks. The proposed scheduling algorithm that is optimal for the considered FH H, jointly considers the information flow on both UL and DL. Moreover, we compare our algorithm, which minimizes the MSE for a given H, to other age-penalty functions used in the existing literature. Numerical results reveal that the consideration of control system specific parameters, as done by our proposed metric, outperforms all other considered approximating functions w.r.t. estimation and control performances. This section can be seen as an extension of sections 4.1 and 4.3.

4.4.1 System Model

We consider N independent feedback control loops that are closed over a shared wireless communication network[14]. Each loop i is comprised of a plant $\mathcal{P}_i$, a controller $\mathcal{C}_i$, and a sensor $\mathcal{S}_i$. The plant state is fully observable by the sensor at all times. Each observation of the plant state is transmitted in the form of status update packets over a wireless link to the BS, from where they are forwarded to $\mathcal{C}_i$. The sensor-to-BS and the BS-to-controller links are called UL and DL, respectively. Such a topology is referred to as a *star network* in the literature [TM21] and resembles the one observed in cellular networks, when each

[14]Here, independent means that the system dynamics of one sub-system do not affect any other sub-system.

sensor-controller pair is located in the same cell of a mobile network. Similarly, the star network can be observed in industrial communications, where status updates are first sent to an access point and forwarded to the destination along a second link. Fig. 4.18 illustrates the considered scenario in the following analysis.

Time is normalized to unity and indexed in the form of slots, i.e., $t \in \mathbb{Z}$. A time slot is the smallest time unit in our model. Each transmission of a status update packet starts and ends within the same slot t. Moreover, any UL packet that has been transmitted in slot t cannot be forwarded earlier than in the subsequent slot $t+1$ over the DL. This implies that any packet generated by $\mathcal{S}_i$ requires at least two transmission slots until it is successfully decoded by the controller $\mathcal{C}_i$[15].

Access to the wireless medium is controlled by a centralized scheduler that is located at the BS. Each sensor transmits only when it is granted access to the UL channel. Moreover, if there are multiple packets waiting to be forwarded in the DL, the scheduler decides which one to send. Let $\pi_i^{UL}(t), \pi_i^{DL}(t) \in \{0,1\}$ indicate whether a user has been scheduled for a UL and DL transmission in slot t, respectively. The scheduler assumes that any two or more simultaneous in the same link would fail due to collision. Therefore, if sensor $\mathcal{S}_i$ is scheduled for a UL, i.e., $\pi_i(t) = 1$, then for any other sensor $\mathcal{S}_j$ with $j \neq i$, $\pi_j(t) = 0$ holds. Analogously, the scheduler transmits only a single DL packet in slot t. As a result, it holds for both links that $\sum_i \pi_i^{UL}(t) \leq 1$ and $\sum_i \pi_i^{DL}(t) \leq 1$ at all times.

Each link follows the GE model, as depicted in Fig. 4.13. If the $\mathcal{S}_i$-to-BS is in a good state, i.e., $\sigma_i^{UL}(t)$, the outcome of a UL transmission by $\mathcal{S}_i$ fails with a probability of $p_i^{UL}(t) = p_G$. Analogously, p_i^{DL} denotes the failure probability of a transmission between the BS and $\mathcal{C}_i$. On the other hand, when in B state, the packet loss probability in the corresponding link becomes p_B, with $p_B > p_G$. The resulting behavior can be formulated as:

$$p_i^{UL}(t) = \begin{cases} p_G & \text{, if } \sigma_i^{UL}(t) = G, \\ p_B & \text{, if } \sigma_i^{UL}(t) = B, \end{cases} \qquad p_i^{DL}(t) = \begin{cases} p_G & \text{, if } \sigma_i^{DL}(t) = G, \\ p_B & \text{, if } \sigma_i^{DL}(t) = B, \end{cases} \tag{4.59}$$

The transitions between the good and bad states occur with a stationary probability p_{G2B} and p_{B2G} for all sub-systems, i.e.:

$$\begin{aligned} p_{G2B} &\triangleq \Pr\left[\sigma_i^{UL}(t) = B \mid \sigma_i^{UL}(t-1) = G\right] = \Pr\left[\sigma_i^{DL}(t) = B \mid \sigma_i^{DL}(t-1) = G\right], \\ p_{B2G} &\triangleq \Pr\left[\sigma_i^{UL}(t) = G \mid \sigma_i^{UL}(t-1) = B\right] = \Pr\left[\sigma_i^{DL}(t) = G \mid \sigma_i^{DL}(t-1) = B\right]. \end{aligned} \tag{4.60}$$

[15]This is different than in sections 4.2 and 4.3 considered models, which assume a constant delay of one slot in case of a success.

It is important to mention that $\sigma_i^{UL}(t)$ and $\sigma_i^{DL}(t)$ are two independent random variables.

Let the binary variable $\delta_i^{UL}(t) \in \{0, 1\}$ indicate a successful reception for the UL by $\delta_i^{UL}(t) = 1$. Moreover, it becomes zero in case of a failed transmission with a probability of $p_i^{UL}(t)$ given that the sensor $\mathcal{S}_i$ has been scheduled, i.e.:

$$\begin{aligned} &\Pr\left[\delta_i^{UL}(t) = 0 \mid \pi_i^{UL}(t) = 1\right] = p_i^{UL}(t), \\ &\Pr\left[\delta_i^{UL}(t) = 1 \mid \pi_i^{UL}(t) = 1\right] = 1 - p_i^{UL}(t). \end{aligned} \tag{4.61}$$

The definition for the DL is analogous:

$$\begin{aligned} &\Pr\left[\delta_i^{DL}(t) = 0 \mid \pi_i^{DL}(t) = 1\right] = p_i^{DL}(t), \\ &\Pr\left[\delta_i^{DL}(t) = 1 \mid \pi_i^{DL}(t) = 1\right] = 1 - p_i^{DL}(t). \end{aligned} \tag{4.62}$$

For the sake of completeness, we provide the probability of a successful reception by the BS under the condition that a sub-system i has not been scheduled in time slot t as zero, i.e., $\Pr\left[\delta_i^{UL/DL}(t) = 1 \mid \pi_i^{UL/DL}(t) = 0\right] = 0$. This follows from the assumption that any sub-system i refrains from transmitting unless it has been granted medium access by the scheduler.

We assume that sensors measure the system state at the beginning of each slot[16]. This means that a new packet is generated by a sensor $\mathcal{S}_i$ every slot and is ready to be transmitted without any measurement delay. We assume a LCFS strategy with packet drop meaning that every new packet replaces the outdated one in the transmission queue. As a result, there is always a single packet in the transmission queue that contains information about the current system state.

4.4.1.1 Age of Information Model

According to the description above, out network comprises $2N + 1$ nodes, i.e., N sensors, N controllers and the BS. We construct a model, in which we define the AoI from the perspective of each node separately. To that end, let us first define the AoI for loop i at the BS, i.e., $\Delta_i^{BS}(t)$ as the elapsed time since the generation of the freshest information. It can be expressed as the difference between now and the generation time of the most recent information as:

$$\Delta_i^{BS}(t) \triangleq t - \nu_i^{BS}(t), \tag{4.63}$$

where the dynamics of $\nu_i^{BS}(t)$ are given as:

$$\nu_i^{BS}(t+1) = \begin{cases} \nu_i^{BS}(t) & \text{, if } \pi_i^{UL}(t) \cdot \delta_i^{UL}(t) = 1, \\ t & \text{, if } \pi_i^{UL}(t) \cdot \delta_i^{UL}(t) = 0. \end{cases} \tag{4.64}$$

[16]This is different than the previous section, in which we assume the sampling period of multiple slots. Although both assumptions are valid from the practical point of view, the model used in this section simplifies the presentation.

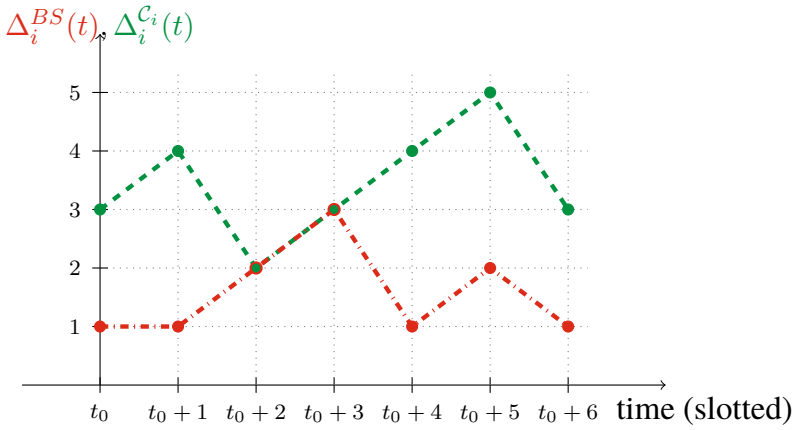


Figure 4.19: A sample sequence of age at the BS and at the controller i. Such a sequence for $\Delta_i^{BS}(t)$ and $\Delta_i^{\mathcal{C}_i}(t)$ can be observed in our system when the UL transmissions scheduled at t_0-1, t_0, t_0+3, and t_0+5 are successful in the interval $t \in [t_0-1, t_0+5]$, respectively.

It is important to emphasize that in case of successful transmission, $\nu_i^{BS}(t+1)$ drops to t. This behavior is caused by the fact that the status update packets in the UL always contain the plant process' system state; or equivalently:

$$\Delta_i^{\mathcal{S}_i}(t) \triangleq t - \nu_i^{\mathcal{S}_i}(t) = 0, \quad \forall i, t. \tag{4.65}$$

The equation above also implies that the AoI at the BS followed by a successful transmission is always one:

$$\Delta_i^{BS}(t+1) = \begin{cases} \Delta_i^{BS}(t)+1 & \text{, if } \pi_i^{UL}(t)\cdot\delta_i^{UL}(t)=1, \\ 1 & \text{, if } \pi_i^{UL}(t)\cdot\delta_i^{UL}(t)=0, \end{cases} \tag{4.66}$$

The definition of the AoI at the controllers is analogous and can be characterized as:

$$\Delta_i^{\mathcal{C}_i}(t) \triangleq t - \nu_i^{\mathcal{C}_i}(t), \tag{4.67}$$

with:

$$\nu_i^{\mathcal{C}_i}(t+1) = \begin{cases} \nu_i^{BS}(t), & \text{, if } \pi_i^{UL}(t)\cdot\delta_i^{UL}(t)=1, \\ \nu_i^{\mathcal{C}_i}(t), & \text{, if } \pi_i^{UL}(t)\cdot\delta_i^{UL}(t)=0, \end{cases} \tag{4.68}$$

In simple words, if the controller does not receive any new information in slot t, the timestamp of the latest observation remains constant. Otherwise, upon receiving a packet, $\nu_i^{\mathcal{C}_i}(t+1)$ is overwritten with the generation time of the most recent update that has been available to the BS by t.

Fig. 4.19 illustrates a sample sequence of $\Delta_i^{BS}(t)$ and $\Delta_i^{\mathcal{C}_i}(t)$ for the interval $t \in [t_0, t_0+6]$ following the introduced AoI model as in (4.63) and (4.67). One should mention that due to

the start network topology, every packet that is available at the controller must have passed through the BS. Therefore, the AoI at the BS is always less than or equal to the age at the controller, i.e., $\Delta_i^{BS}(t) \leq \Delta_i^{\mathcal{C}_i}(t)$ for all t and i. Additionally, as we assume each transmissions to take one full slot duration, $\Delta_i^{\mathcal{C}_i}$ is always greater than or or equal to two due to the fact that sensor-controller pairs are two-hop away.

4.4.1.2 Control Model

We represent the behavior of each sub-system by the LTI model in discrete-time, i.e.:

$$\boldsymbol{x}_i[t] = \boldsymbol{A}_i \boldsymbol{x}_i[t] + \boldsymbol{B}_i \boldsymbol{u}_i[t] + \boldsymbol{w}_i[t], \tag{4.69}$$

with time-invariant system matrix $\boldsymbol{A}_i \in \mathbb{R}^{n_i \times n_i}$ and input matrix $\boldsymbol{B}_i \in \mathbb{R}^{n_i \times m_i}$. Moreover, $\boldsymbol{w}_i[t] \in \mathbb{R}^{n_i}$ is a zero-mean random noise vector modeling the external disturbances and follows a multi-variate Gaussian distribution with the covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{n_i \times n_i}$. $\boldsymbol{x}_i[t]$ and $\boldsymbol{u}_i[t]$ are vectors of appropriate sizes denoting the system state and control input. It is important to mention that in the following analysis the time slot t of the network corresponds to the t-th time step of control systems. In other words, the duration of a time slot equals to the sampling period for every sub-system. As we consider unit time in our model, the sampling period, as well as the sampling frequency equals to one[17].

At the beginning of each time step, the controller calculates the control input $\boldsymbol{u}_i[t]$ based on the available observation history. In order to compensate for the shortcomings of the wireless communication network in the feedback loop, the controller estimates the plant state remotely based on the most recent knowledge that it has acquired. Suppose that the freshest information until the moment when $\boldsymbol{u}_i[t]$ needs to be determined is $\boldsymbol{x}_i[t - \Delta_i^{\mathcal{C}_i}(t)]$. Thus, the conditional expectation of the state, which minimizes the mean squared estimation error is given as:

$$\begin{aligned} \hat{\boldsymbol{x}}_i[t] &\triangleq \mathbb{E}\left[\boldsymbol{x}_i[t] \mid \Delta_i^{\mathcal{C}_i}(t), \boldsymbol{x}_i[t - \Delta_i^{\mathcal{C}_i}(t)]\right] \\ &= \boldsymbol{A}_i^{\Delta_i^{\mathcal{C}_i}(t)} \boldsymbol{x}_i[t - \Delta_i^{\mathcal{C}_i}(t)] + \sum_{q=1}^{\Delta_i^{\mathcal{C}_i}(t)} \boldsymbol{A}_i^{q-1} \boldsymbol{B}_i \boldsymbol{u}_i[t-q]. \end{aligned} \tag{4.70}$$

The equation implies that the controller has to store the last $\Delta_i^{\mathcal{C}_i}(t)$ control inputs in the memory. However, it does not impose any additional communication effort as this information is already present at each $\mathcal{C}_i$.

[17]For legacy reasons, while we use round brackets for the network and age model, we choose square brackets for the control model. As the network and control models advance with the same speed, one could use the round brackets for the difference equation (4.69) as well

The goal of the closed-loop system is to minimize the LQG cost function $\mathcal{F}_i$:

$$\mathcal{F}_i \triangleq \limsup_{K\to\infty} \frac{1}{K} \sum_{t=0}^{K-1} (\boldsymbol{x}_i[t])^T \boldsymbol{Q}_i \boldsymbol{x}_i[t] + (\boldsymbol{u}_i[t])^T \boldsymbol{R}_i \boldsymbol{u}_i[t]. \tag{4.71}$$

A lower $\mathcal{F}_i$ indicates a higher QoC. The optimal state feedback matrix $\boldsymbol{L}_i \in \mathbb{R}^{m_i \times n_i}$ is obtained as if every link in the feedback loop were ideal. Therefore, the control law is formulated by the following matrix equation:

$$\boldsymbol{u_i}[t] = -\boldsymbol{L}_i \hat{\boldsymbol{x}}_i[t], \tag{4.72}$$

which corresponds to certainty equivalent control. In other words, the controller assumes a separation principle between estimation and control. The optimal matrix $\boldsymbol{L}_i$ minimizing the LQG cost is obtained from:

$$\boldsymbol{L}_i = \left(\boldsymbol{R}_i + \boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{B}_i\right)^{-1} \boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \tag{4.73}$$

Here, the matrix $\boldsymbol{P}_i$ of appropriate size is a solution of the DARE as in (2.18). Although, the DARE does not take any network characteristics into account, it has been shown in [Mai+22] that the certainty equivalent controller with $\boldsymbol{L}_i$ as in (4.73) is optimal despite the presence of packet loss and delay.

We define the (network-induced) estimation error as the difference between the real and estimated states, i.e.:

$$\boldsymbol{e}_i[t] \triangleq \boldsymbol{x}_i[t] - \hat{\boldsymbol{x}}_i[t]. \tag{4.74}$$

One can easily show that by plugging (4.69) and (4.70) in (4.74), the formula for the estimation error can be simplified to:

$$\boldsymbol{e}_i[t] = \sum_{q=1}^{\Delta_i^{\mathcal{C}_i}(t)} \boldsymbol{A}_i^{q-1} \boldsymbol{w}_i[t-q]. \tag{4.75}$$

The equation above consists of $\Delta_i^{\mathcal{C}_i}(t)$ addends. Each addend is a function of the time-invariant system matrix and a random noise vector $\boldsymbol{w}_i[t-q]$.

Consequently, the MSE at $\mathcal{C}_i$ can be expressed as:

$$\begin{aligned} MSE_i(\Delta_i^{\mathcal{C}_i}[t]) &\triangleq \mathbb{E}\left[(\boldsymbol{e}_i[t])^T \boldsymbol{e}_i[t]\right] \\ &= \sum_{q=1}^{\Delta_i^{\mathcal{C}_i}(t)} \mathsf{tr}\left((\boldsymbol{A}_i^T)^q \boldsymbol{A}_i^q \boldsymbol{\Sigma}_i\right) \end{aligned} \tag{4.76}$$

which maps the vector $\boldsymbol{e}_i[t] \in \mathbb{R}^{n_i}$ to a scalar value. It is important to mention that any scheduler utilizing MSE_i should be able to keep track of $\Delta_i^{\mathcal{C}_i}(t)$ remotely. Nevertheless, since in our scenario, all packets are relayed over the BS, where the scheduler is located

at, it would be feasible for the scheduler to obtain $\Delta_i^{\mathcal{C}_i}(t)$ through a simple acknowledgment mechanism in the DL. Moreover, the knowledge of $\boldsymbol{A}_i$ and $\boldsymbol{\Sigma}_i$ is necessary to obtain MSE_i. Due to the fact that these are time-invariant matrices, a one-time information exchange prior to operation would suffice. Note that (4.76) does not depend on the feedback gain matrix $\boldsymbol{L}_i$, hence the time evolution of the MSE is independent of the controller design.

4.4.2 Greedy Uplink and Downlink Scheduling

So far, we have introduced how the communication network and feedback loops are modeled in our considered scenario. In addition, the effects of increasing AoI at the controllers on the estimation error has been defined. In this subsection, we discuss various greedy scheduling policies that are relatively simple in design and implementation. Consider a scheduling policy that decides on the UL and DL schedules based on the transmission history on both links and broadcasts this information to every node in the network[18]. That is, at the beginning of each time slot t, the scheduler determines the joint schedule $\boldsymbol{a}[t] = [a^{UL}(t) \quad a^{DL}(t)]^T$ with $a^{UL}, a^{DL} \in \{\varnothing, 1, 2, \dots, N\}$. If a user i is scheduled in the UL, i.e., $a^{UL}(t) = i$, the sensor $\mathcal{S}_i$ initiates an uplink transmission containing the latest system state $\boldsymbol{x}_i[t]$. Note that $a^{UL}(t) = i$ implies $\pi_i^{UL}(t) = 1$. Similarly, $a^{DL}(t) = i$ means that a DL packet of sub-system i is sent by the BS, i.e., $a^{DL}(t) = i$ and $\pi_i^{DL} = 1$. The usage of the empty set $\varnothing$ is particularly important in this section. For instance, if the BS does not have any new packet in its transmission queue, the DL slot remains idle. Such a case is covered by $a^{DL}(t) = \varnothing$.

In order to decide whether a new packet is eligible for transmission by any of the nodes, the scheduler utilizes the *age difference* between source-destination pairs in each link. For the UL case, the eligibility criteria is given as the positive age difference, i.e., $\Delta_i^{BS}(t) > \Delta_i^{\mathcal{S}_i}(t) = 0$. As sensors always have a new packet to send due to their sampling frequency of one, all sensors are eligible for transmission in every slot. However, this is not the case for the DL. As stated in the previous paragraph, the BS may not have received new information from $\mathcal{S}_i$ since the corresponding controller $\mathcal{C}_i$ was last updated in an earlier slot. In Fig. 4.19, time slots $t_0 + 2$ and $t_0 + 3$ constitute an example to such a situation, where the ages at the BS and the controller equalize. In that case, we refer to a DL transmission towards $\mathcal{C}_i$ as ineligible, due to $\Delta_i^{BS}(t) = \Delta_i^{\mathcal{C}_i}(t)$. If none of the sub-systems are eligible in slot t, then the BS does not send any packet in the DL leading to $a^{DL}(t) = \varnothing$.

The scheduling decisions are tightly coupled with the estimation and control performances in the network. Depending on the scheduling policy, the AoI follows a different trajectory, which in return affects the network-induced estimation error. Furthermore, the estimation accuracy defines the degree of sub-optimality of control inputs that are obtained by using

[18]We assume that transmission of the broadcast messages is instantaneous and error-free

the state estimate instead of the actual state, as in (4.72)[19]. Therefore, depending on how the wireless network resources are distributed among multiple users, defined by the selected scheduling policy $\boldsymbol{\pi}$, the network can directly affect control performance metrics, such as LQG cost and MSE from (4.71) and (4.76), respectively.

As an alternative to contention-free protocols, the contention-based protocols from the existing literature may also come into consideration, e.g., ALOHA [Abr70], Slotted ALOHA [Rob75a], and age-dependent random access protocol [CGL20]. However, despite requiring less implementation effort due to their decentralized nature, they are also known for low throughput in multi-user scenarios due to their high packet loss. Thus, we do not consider contention-based protocols in this section and focus only on contention-free protocols[20].

4.4.2.1 Application-Unaware Greedy Policies

We consider two heuristic scheduling policies that are well studied in wireless communications research: 1) *Round-Robin (RR)* and 2) *Maximum Throughput (MT)*. These two protocols are classified as application-unaware policies due to their negligence of application-layer metrics, such as AoI and MSE.

Round-Robin Policy: RR is one of the most frequently deployed scheduling policies in industry due to its ease of implementation. It has been used in cellular networks [Cap+13], as well as in the context of remote estimation use cases [Zou+19]. Under the RR policy, all nodes are scheduled in a predetermined fixed circular order. It can be argued that RR is a "good" heuristic for real-time applications as it offers fairness in time thanks to its periodic resource allocation nature. Note that the classical RR scheduler is channel-unaware by definition as it does not consider channel conditions, such as packet success probability.

Maximum Throughput Policy: The MT policy prioritizes users with the best channel quality, hence targeting a throughput maximization [Cap+13]. The MT scheduler is also known for its unfair nature as it may lead to the starvation of those users with bad channel conditions. Given our system model, in which the channel quality is represented by the packet loss probability, the scheduling under the MT policy is expressed as:

$$\begin{aligned} a^{UL}(t) &= \underset{i \in \mathcal{A}^{UL}(t)}{\arg\min} \quad p_i^{UL}(t), \\ a^{DL}(t) &= \underset{i \in \mathcal{A}^{DL}(t)}{\arg\min} \quad p_i^{DL}(t). \end{aligned} \tag{4.77}$$

[19]The control law is optimal if $\boldsymbol{x} = \hat{\boldsymbol{x}}$ since the optimal feedback gain matrix $\boldsymbol{L}_i$ minimizes $\mathcal{F}_i$. Due to the impairment between the real and estimated states caused by the network, the estimation error leads to the sub-optimality of $\boldsymbol{u}$ w.r.t. the LQG cost.

[20]The following chapter studies the control performance achieved by ALOHA and Slotted ALOHA protocols in a practical setup

Here, $\mathcal{A}^{UL}(t)$ and $\mathcal{A}^{DL}(t)$ denote the feasible actions and are given as:

$$\begin{aligned}\mathcal{A}^{UL}(t) &\triangleq \{i:\ i \in \{1,\dots,N\}, \Delta_i^{BS}(t) > \Delta_i^{\mathcal{S}_i}(t)\} \\ &= \{1,2,\dots,N\}\end{aligned} \tag{4.78}$$

$$\mathcal{A}^{DL}(t) \triangleq \{i:\ i \in \{1,\dots,N\}, \Delta_i^{\mathcal{C}_i}(t) > \Delta_i^{BS}(t)\} \cup \{\varnothing\} \tag{4.79}$$

Ties are broken arbitrarily. The necessity for $\mathcal{A}^{DL}$ arises in order to prevent sub-system i from being scheduled, although the BS does not have any novel update to send. As we are going to show later, by defining the set of feasible actions, we are able to reduce the computational complexity of our proposed algorithm.

4.4.2.2 Application-Aware Greedy Policies

As an alternative to application-unaware policies, we introduce two centralized greedy scheduling mechanisms from the state-of-the-art: 1) *Maximum Age First (MAF)* and 2) *Maximum Error First (MEF)*. Their implementation relies on the propagation of the application layer metrics such as AoI and MSE down to the data link layer; thus, they can be considered as more challenging and complex to implement when compared to RR and MT.

Maximum Age First Policy: As the name suggests, the MAF policy is a greedy strategy that prioritizes the user with the highest instantaneous age [Bed+19]. It has also been called "maximum age first" in the literature [Chi+22]. When the MAF policy is applied, the users are scheduled according to the following rule:

$$a^{UL}(t) = \underset{i \in \mathcal{A}^{UL}(t)}{\arg\min} \quad \Delta_i^{BS}(t), \tag{4.80}$$

$$a^{DL}(t) = \underset{i \in \mathcal{A}^{DL}(t)}{\arg\min} \quad \Delta_i^{\mathcal{C}_i}(t). \tag{4.81}$$

$$\tag{4.82}$$

The main difference between MAF and RR is that the MAF is adaptive to the outcome of the past transmissions, whereas RR is not. In particular, if a user is scheduled in the UL but the transmission fails, i.e., $\pi_i^{UL}(t) = 1$, $\delta_i^{UL}(t) = 0$, one can expect that the same scheduling decision is made in the subsequent slot $t+1$ as well. However, under the RR strategy, the scheduling actions are taken in a fixed order, independent of the outcome of transmissions.

Maximum Error First Policy: The MEF policy is firstly proposed by Walsh et al. in [WY01] for single-hop multi-loop scenarios. In its original form, it prioritizes the user with the highest network-induced error, i.e., $a(t) = \arg\max_i \|\boldsymbol{e}_i[t]\|$. However, since the actual system state $\boldsymbol{x}_i[t]$ is unknown to the centralized scheduler, it is not feasible to assume a global knowledge

of $\boldsymbol{e}_i[t]$. Therefore, we rule out a scheduling design based on the actual error $\boldsymbol{e}_i[t]$ and limit our study to the utilization of the MSE from (4.76).

Moreover, we normalize the MSE by the value it takes when AoI is one. The necessity for normalization can be explained with the help of an example: Consider two control applications of different types sharing the wireless communication network. By definition, each element in $\boldsymbol{e}_i(t) \in \mathbb{R}^{n_i}$ has the same unit as the system state $\boldsymbol{x}_i(t)$, e.g., kelvin, meter, radians. Thus, comparing the MSE in its raw form may potentially lead to comparison of numbers in different orders of magnitude, hence would only be possible if all sub-systems' units were identical. To overcome this issue, we suggest normalizing the MSE as:

$$nMSE_i(\Delta_i(t)) \triangleq \frac{MSE_i(\Delta_i(t))}{MSE_i(1)}, \tag{4.83}$$

such that the resulting metric is dimensionless. The normalized MSE (nMSE) captures how rapidly the estimation inaccuracy grows relative to the baseline case of MSE at age of one. We utilize the nMSE for scheduling instead of MSE and in consequence, our implementation of the MEF scheduler can be expressed by:

$$a^{UL}(t) = \underset{i \in \mathcal{A}^{UL}(t)}{\arg\max} \quad nMSE_i(\Delta_i^{BS}(t)), \tag{4.84}$$

$$a^{DL}(t) = \underset{i \in \mathcal{A}^{DL}(t)}{\arg\max} \quad nMSE_i(\Delta_i^{\mathcal{C}_i}(t)). \tag{4.85}$$

The MEF scheduler as defined by (4.85) is a modified version of the greedy scheduler from section 4.1, where the MSE is normalized. It is important to mention that the MAF and MEF schedulers ar channel-unaware as they are defined and implemented in this section.

4.4.3 Optimal Joint Uplink and Downlink Finite Horizon Scheduling

The previously introduced greedy scheduling algorithms are simple heuristics but do not guarantee optimality in task- and application-specific performance. As a solution, we formulate the centralized scheduling problem as a finite horizon FH optimization problem and propose a policy $\boldsymbol{\pi}$ that is optimal for a given horizon H. Our approach is based on minimizing the expected cost over H. In brief, the H-optimal scheduler generates a tree structure including all possible trajectories of the system, incorporating all possible states and costs that could appear within H steps from now. As already stated in section 4.3, the longer H is, the more "farsighted" is the algorithm, as we increase the time horizon that our scheduler takes into account while deciding on the optimal action.

4.4.3.1 States and Actions

Let $\boldsymbol{\nu}(t)$ be a column vector of size $3N$ containing the generation time of the freshest information at sensors, BS, and controllers at time t:

$$\boldsymbol{\nu}(t) \triangleq \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}(t) \\ \boldsymbol{\nu}^{BS}(t) \\ \boldsymbol{\nu}^{\mathcal{C}}(t) \end{bmatrix}, \tag{4.86}$$

with:

$$\boldsymbol{\nu}^{\mathcal{S}}(t) \triangleq \begin{bmatrix} t \\ t \\ \vdots \\ t \end{bmatrix}, \qquad \boldsymbol{\nu}^{BS}(t) \triangleq \begin{bmatrix} \nu_1^{BS}(t) \\ \nu_2^{BS}(t) \\ \vdots \\ \nu_N^{BS}(t) \end{bmatrix}, \qquad \boldsymbol{\nu}^{\mathcal{C}}(t) \triangleq \begin{bmatrix} \nu_1^{\mathcal{C}_1}(t) \\ \nu_1^{\mathcal{C}_2}(t) \\ \vdots \\ \nu_1^{\mathcal{C}_N}(t) \end{bmatrix}. \tag{4.87}$$

We refer to the vector $\boldsymbol{\nu}(t)$ as the *network state* and to the scheduling decision $\boldsymbol{a}[t] = [a^{UL}(t) \quad a^{DL}(t)]^T$ as *action*. One should mention that it is possible to map each network state $\boldsymbol{\nu}(t)$ to an age vector $\boldsymbol{\Delta}(t)$ with:

$$\boldsymbol{\Delta}(t) \triangleq \begin{bmatrix} 0 & \dots & 0 & \Delta_1^{BS}(t) & \dots & \Delta_N^{BS}(t) & \Delta_1^{\mathcal{C}_1}(t) & \dots & \Delta_N^{\mathcal{C}_N}(t) \end{bmatrix}^T. \tag{4.88}$$

Note that mapping a given $\boldsymbol{\Delta(t)}$ to $\boldsymbol{\nu}(t)$ is not possible as a $\boldsymbol{\Delta}(t)$ does not imply a unique $\boldsymbol{\nu}(t)$. We will exploit this relationship between $\boldsymbol{\Delta}(t)$ and $\boldsymbol{\nu}(t)$ later to use dynamic programming for complexity reduction. Having said that, we continue with $\boldsymbol{\nu}(t)$ throughout the following analysis.

Given a network state $\boldsymbol{\nu}(t) = \boldsymbol{\nu}_\tau$ and an action $\boldsymbol{a}(t) = [i \; j]$ with $i \in \mathcal{A}^{UL}(t)$, $j \in \mathcal{A}^{DL}(t)$, we can obtain the transition probability to a next state as a function of the transmission outcome on respective links, i.e., $\boldsymbol{\nu}(t+1) = f(\boldsymbol{\nu}(t), \boldsymbol{a}(t), \delta_i^{UL}(t), \delta_j^{DL}(t))$. In particular, if $j \neq \varnothing$, there are four possible next states depending on the transmission outcome on each link. Given the network state $\boldsymbol{\nu}(t)$ as defined in (4.86), we can formulate the transition probabilities to the four possible next states as:

$$\begin{aligned}
&\Pr\left[\boldsymbol{\nu}(t+1) = \boldsymbol{\nu}_{\tau+1}^{\mathbf{11}} \mid \boldsymbol{\nu}(t) = \boldsymbol{\nu}_\tau, \boldsymbol{a}(t) = [i \quad j]^T\right] = \left(1 - p_i^{UL}(t)\right)\left(1 - p_j^{DL}(t)\right) \\
&\Pr\left[\boldsymbol{\nu}(t+1) = \boldsymbol{\nu}_{\tau+1}^{\mathbf{10}} \mid \boldsymbol{\nu}(t) = \boldsymbol{\nu}_\tau, \boldsymbol{a}(t) = [i \quad j]^T\right] = \left(1 - p_i^{UL}(t)\right) p_j^{DL}(t) \\
&\Pr\left[\boldsymbol{\nu}(t+1) = \boldsymbol{\nu}_{\tau+1}^{\mathbf{01}} \mid \boldsymbol{\nu}(t) = \boldsymbol{\nu}_\tau, \boldsymbol{a}(t) = [i \quad j]^T\right] = p_i^{UL}(t)\left(1 - p_j^{DL}(t)\right) \\
&\Pr\left[\boldsymbol{\nu}(t+1) = \boldsymbol{\nu}_{\tau+1}^{\mathbf{00}} \mid \boldsymbol{\nu}(t) = \boldsymbol{\nu}_\tau, \boldsymbol{a}(t) = [i \quad j]^T\right] = p_i^{UL}(t) p_j^{DL}(t)
\end{aligned} \tag{4.89}$$

with:

$$\boldsymbol{\nu}^{\mathbf{11}}_{\tau+1} = \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}(t+1) \\ \nu_1^{BS}(t) \\ \vdots \\ \nu_N^{\mathcal{S}_i}(t) \\ \vdots \\ \nu_N^{BS}(t) \\ \nu_1^{\mathcal{C}_1}(t) \\ \vdots \\ \nu_j^{BS}(t) \\ \vdots \\ \nu^{\mathcal{C}_N}(t) \end{bmatrix}, \boldsymbol{\nu}^{\mathbf{10}}_{\tau+1} = \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}(t+1) \\ \nu_1^{BS}(t) \\ \vdots \\ \nu_N^{\mathcal{S}_i}(t) \\ \vdots \\ \nu_N^{BS}(t) \\ \boldsymbol{\nu}^{\mathcal{C}}(t) \end{bmatrix}, \boldsymbol{\nu}^{\mathbf{01}}_{\tau+1} = \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}(t+1) \\ \boldsymbol{\nu}^{BS}(t) \\ \nu_1^{\mathcal{C}_1}(t) \\ \vdots \\ \nu_j^{BS}(t) \\ \vdots \\ \nu^{\mathcal{C}_N}(t) \end{bmatrix}, \boldsymbol{\nu}^{\mathbf{00}}_{\tau+1} = \begin{bmatrix} \boldsymbol{\nu}^{\mathcal{S}}(t+1) \\ \boldsymbol{\nu}^{BS}(t) \\ \boldsymbol{\nu}^{\mathcal{C}}(t) \end{bmatrix}.$$

Note the differentiation between vectors written in bold, e.g., $\boldsymbol{\nu}^{BS}(t) \in \mathbb{Z}^N$ and scalar values $\nu_i^{BS}(t) \in \mathbb{Z}$. Here, a "1" in the superscript indicates success in the corresponding link with the first position being the UL and the second being the DL. For instance, $\boldsymbol{\nu}^{\mathbf{10}}_{\tau+1}$ denotes the next state when the UL transmission is successful, but the DL transmission fails. In case of an idle DL slot, $a^{DL}(\tau) = \varnothing$, the number of next possible states decreases to two, i.e., $\boldsymbol{\nu}^{\mathbf{10}}_{\tau+1}$ and $\boldsymbol{\nu}^{\mathbf{00}}_{\tau+1}$.

4.4.3.2 The H-Stage Problem and Finite Horizon Cost

We consider a cost function $g : \mathbb{Z}^{3N} \mapsto \mathbb{R}$ mapping a network state to an immediate cost of the form:

$$g(\boldsymbol{\nu}(t)) \triangleq \sum_i g_i(\nu_i^{\mathcal{S}_i}(t), \nu_i^{BS}(t), \nu_i^{\mathcal{C}_i}(t)). \tag{4.90}$$

Here, $g_i : \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \mapsto \mathbb{R}$ characterizes the contribution of sub-system i to $g(\boldsymbol{\nu}(t))$. A simple example of such a function would be the weighted sum of AoI at each node:

$$g_i\left(\nu_i^{\mathcal{S}_i}(t), \nu_i^{BS}(t), \nu_i^{\mathcal{C}_i}(t)\right) = w_i^{\mathcal{S}_i} \underbrace{(t - \nu_i^{\mathcal{S}_i}(t))}_{=0} + w_i^{BS} \underbrace{(t - \nu_i^{BS}(t))}_{=\Delta_i^{BS}(t)} + w_i^{\mathcal{C}_i} \underbrace{(t - \nu_i^{\mathcal{C}_i}(t))}_{=\Delta_i^{\mathcal{C}_i}(t)}$$

with $w_i^{\mathcal{S}_i}, w_i^{BS}, w_i^{\mathcal{C}_i} > 0$, $\forall i$. The selection of the cost function g plays a crucial role in the achieved performance and should capture the task-specific targets fairly well that the network aims at. In the results section, we will discuss the effect of selecting different cost functions on control performance.

Let us now define an additive cost of the form:

$$\mathcal{J}_t^{\boldsymbol{\pi}_H}(\boldsymbol{\nu}(t)) \triangleq \mathbb{E}\left[\sum_{\tau=t}^{t+H} g(\boldsymbol{\nu}(\tau)) \;\middle|\; \boldsymbol{\pi}_\tau\right] \tag{4.91}$$

for an initial state $\boldsymbol{\nu}(t)$ and possible future states, i.e., $\boldsymbol{\nu}(\tau+1) = f(\boldsymbol{\nu}(\tau), \boldsymbol{a}(\tau), \delta_i^{UL}(\tau), \delta_j^{DL}(\tau))$. $\mathcal{J}_t^{\boldsymbol{\pi_H}}$ denotes the expected H-stage cost when the scheduling policy $\boldsymbol{\pi_H} = \{\boldsymbol{\pi}_t, \boldsymbol{\pi}_{t+1}, \ldots, \boldsymbol{\pi}_{t+H-1}\}$ is employed over the horizon H. Here, $\boldsymbol{\pi}_\tau$ are functions that map a state in stage τ to a scheduling decision, i.e., $\boldsymbol{\pi_\tau}(\boldsymbol{\nu}(\tau)) = \boldsymbol{a}(\tau)$. The expectation is taken w.r.t. $\delta_i^{UL}(t)$ and $\delta_j^{DL}(t)$ that together with the scheduling decision $\boldsymbol{a}(t)$ define the occurrence probability of any next state. In consequence, we aim to find the optimal scheduling policy $\boldsymbol{\pi}_{\boldsymbol{H}}^*$ for the *H-stage problem* with the optimal cost:

$$\mathcal{J}_t^{\boldsymbol{\pi}_H^*}(\nu(t)) = \min_{\boldsymbol{\pi_H}} \quad \mathcal{J}_t^{\boldsymbol{\pi_H}}(\boldsymbol{\nu}(t)) \tag{4.92}$$

Note that although $\boldsymbol{\pi_H}$ looks H slots into the future, the H-stage problem is solved every time slot, and the optimal action $\boldsymbol{\pi}_{\boldsymbol{t}}^*(\boldsymbol{\nu}(t)) = \boldsymbol{a}^*(t)$ is executed both for the UL and DL.

4.4.3.3 The Finite Horizon Scheduling Algorithm

As it has been shown in [Ber95], the H-stage problem starting at t and ending at $t+H$ can be solved optimally by minimizing the RHS of the equation (4.92) for $\tau \in [t, t+H)$:

$$\mathcal{J}_\tau(\boldsymbol{\nu}(\tau)) = \min_{\boldsymbol{a}(\tau)\in\boldsymbol{\mathcal{A}}(\tau)} \quad \mathbb{E}\left[g(\boldsymbol{\nu}(t)) + \mathcal{J}_{\tau+1}\left(f(\boldsymbol{\nu}(t), \boldsymbol{a}(\tau), \delta_i^{UL}(\tau), \delta_j^{DL}(\tau))\right)\right], \tag{4.93}$$

while we start from the terminal cost given as:

$$\mathcal{J}_H(\boldsymbol{\nu}(t+H)) = g(\boldsymbol{\nu}(t+H)). \tag{4.94}$$

We refer to [Ber95, p. 25] for proof. The equations (4.93) and (4.94) can be explained in plain words as follows: the optimal cost $\mathcal{J}^{\boldsymbol{\pi}_H^*}(\boldsymbol{\nu}(t))$ can be obtained by iterating backwards in time from stage $H-1$ to stage zero, while at each iteration step τ the optimal action $\boldsymbol{a}^*(t)$ solving (4.93) is taken. Once the 0-th stage is solved, the resulting scheduling policy $\boldsymbol{\pi}_{\boldsymbol{H}}^*$ achieving the minimum expected cost $\mathcal{J}_t^{\boldsymbol{\pi}_H^*}$ is optimal.

Similar to section 4.3, we visualize the H-stage problem through an H-level tree structure. Each node in the tree represents a network state occurring within the finite horizon. The 0-th level consists of a single node with state $\boldsymbol{\nu}(t)$ and is called the root node. Moreover, the nodes in the last level are called leaf nodes.Starting from the root node, all feasible state-action pairs define the remaining levels with appropriate transition probabilities. In particular, a joint UL and DL action $\boldsymbol{a}(t)$ can lead to at most four possible next states, as previously explained in (4.89), where each transmission is modeled by an outgoing edge. Moreover, each edge is assigned a transition probability depending on the action, i.e., scheduling decision, and the transmission success probability of the respective links. Once the whole tree is generated, each state's cost is assigned according to (4.93) and (4.94) starting from the leaf nodes. The following steps summarize the resulting algorithm of the FH scheduler:

Finite Horizon Scheduling Algorithm

1. Initialize the current state $\boldsymbol{\nu}[t]$ as the root of the tree.
2. Starting from the root node, determine the feasible actions $\mathcal{A}^{UL}(\tau)$ and $\mathcal{A}^{DL}(\tau)$ and subsequently all possible next states $\boldsymbol{\nu}(\tau+1)$ given an action $\boldsymbol{a}[\tau]$.
3. Add the obtained next states as child nodes to the next level of the tree and initialize the edges between the parent and child nodes with corresponding transition probabilities.
4. Repeat steps 2. and 3. until the H-th level of the tree is constructed.
5. Assign the minimum cost and best action to each node as in (4.93) and (4.94) starting from the leaf nodes.

Once the FH scheduling algorithm completes, the scheduler executes the optimal action $\boldsymbol{a}^*(t) = \boldsymbol{\pi}_{\boldsymbol{t}}^*(\boldsymbol{\nu}(t)$ obtained for the root node. We emphasize that even though the FH scheduling algorithm has obtained the optimal action for all possible future states within the horizon H, the FH scheduler algorithm must be repeated after every time slot. The reason is the modification of the tree's root node and the dynamically changing edge transition probabilities caused by time-varying channel conditions. We would like to remind the reader that the scheduler knows each link's current loss probability but is unaware of future channel conditions, e.g., $p_i^{UL}(t+1)$, $p_i^{DL}(t+2)$. Therefore, the expected cost calculation is performed as if the current packet success probabilities would remain constant throughout the following H slots.

One can easily deduct by looking at the algorithm that constructing such a tree structure is a heavy task in terms of computation complexity. In particular, the complexity strictly depends on the selected FH parameter H and the scale of the network governed by N. A detailed discussion on the FH algorithm's complexity follows in the remainder of this section.

Selecting the cost function: We propose using an additive weighted cost function for immediate state cost given as:

$$g\left(\boldsymbol{\nu}(t)\right) = \sum_{i=1}^{N} w_i^{BS}\, nMSE_i\left(\Delta_i^{BS}(t)\right) + w_i^{\mathcal{C}_i}\, nMSE_i(\Delta_i^{\mathcal{C}_i}(t)). \tag{4.95}$$

$\Delta_i^{BS}(t)$ and $\Delta_i^{\mathcal{C}_i}(t)$ are defined as in (4.63) and (4.67), respectively. w_i^{BS} and $w_i^{\mathcal{C}_i}$ are positive weighting factors for a sub-system i. It is evident that the selected cost function does not only consider the normalized MSE at controllers but also the nMSE at the BS. The reason behind the consideration of Δ_i^{BS} can be explained with the help of a toy example: Suppose a

newly initialized network with $N = 1$ and $H = 1$. The initial AoI at each node is given as $\Delta_1^{\mathcal{S}_1}(0) = 0$ and $\Delta_1^{BS}(0) = \Delta_1^{\mathcal{C}_1}(0) = 5$. In such a setting, if we only reward a cost reduction at the controller, or equivalently, $w_1^{BS} = 0$ but $w_1^{\mathcal{C}_1} > 0$, then a successful UL transmission does not lead to any change in cost within the considered horizon, i.e., by $t + 1$. As a result, the scheduler is indifferent between $a^{UL}(0) = 1$ and $a^{UL}(0) = \varnothing$ since it is not better-off by scheduling any sub-system within $H = 1$ slots. Thus, in order to incorporate a hidden future reward enabled through updating the BS, we choose a positive w_i^{BS} in our framework.

It is important to add that one may use different cost functions than the normalized MSE. Such approaches exist in the relevant existing literature. Through the numerical evaluation that we will present in section 4.4.4, the FH scheduling algorithm can operate with various age-penalty functions. For instance, the scheduling algorithm can employ $g_i(t) = \Delta_i^{BS}(t)+\Delta_i^{\mathcal{C}_i}(t)$ can target information freshness by solving the H-stage problem. Similarly, other common functions can be utilized to penalize increasing age such as $g_i(t) = e^{\alpha_i \Delta_i^{BS}(t)} + e^{\alpha_i \Delta_i^{\mathcal{C}_i}(t)}$ and $g_i(t) = \alpha_i \Delta_i^{BS}(t) + \alpha_i \Delta_i^{\mathcal{C}_i}(t)$ with a design parameter $\alpha_i > 0$ as a multiplier constant in front of the age. Throughout the following section, we simplify the notation by dropping the superscript. As an example, when we say that the scheduler employs the metric $\alpha_i \Delta_i(t)$, it implies $g_i(t) = \alpha_i \Delta_i^{BS}(t) + \alpha_i \Delta_i^{\mathcal{C}_i}(t)$.

4.4.4 Numerical Results and Discussion

In this section, we evaluate the performance of our pro- posed FH scheduler in terms of the long-term average AoI quantifying information freshness, the average MSE capturing the estimation performance, and average LQG cost quantifying the control performance in the network. Our evaluation is not only limited to the presentation of key performance indicators (KPIs) when various selected cost functions from the existing literature are employed, but additionally, we investigate the relationship between the scheduler's far-sightedness and achieved performance.

4.4.4.1 Simulation Details

Our simulation consists of $N = 4$ feedback control loops of heterogeneous type. In particular, one of the sub-systems is an inverted pendulum, which is a well-known real-life application in control theory textbooks [rM08]. As depicted in Fig. 4.20, it consists of a pendulum mounted on a motorized cart. The controller's objective is to hold the pendulum in upright position by moving the cart back and forward. While the inverted pendulum (IP) has a four-by-four system matrix, i.e., $\boldsymbol{A}_{IP} \in \mathbb{R}^{4\times4}$, the remaining applications are a scalar control loop each, i.e., $\boldsymbol{A}_{i\neq IP} \in \mathbb{R}$.

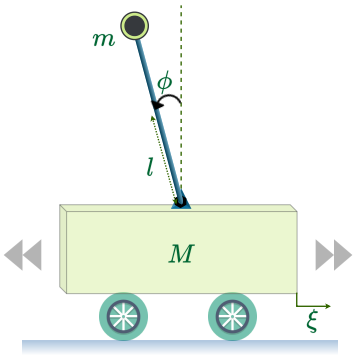


Figure 4.20: An inverted pendulum with motorized cart. The primary goal of the controller is to hold the pendulum in upright position, i.e., to keep $|\phi|$ close to zero as much as possible.

The parameter selection of our control model can be summarized as:

$$\boldsymbol{A}_2 = \boldsymbol{A}_{IP} = \begin{bmatrix} 1.0000 & 0.0100 & 0.0001 & 0.0000 \\ 0.0000 & 0.9983 & 0.0191 & 0.0001 \\ 0.0000 & 0.0000 & 1.0017 & 0.0100 \\ 0.0000 & -0.0049 & 0.3351 & 1.0017 \end{bmatrix},$$

$$\boldsymbol{B}_2 = \boldsymbol{B}_{IP} = \begin{bmatrix} 0.0001 \\ 0.1706 \\ 0.0002 \\ 0.0488 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_{IP} = \begin{bmatrix} 6.4\ 10^{-7} & 0 & 0 & 0 \\ 0 & 4.9\ 10^{-7} & 0 & 0 \\ 0 & 0 & 2.74\ 10^{-5} & 0 \\ 0 & 0 & 0 & 4.87\ 10^{-5} \end{bmatrix},$$

$$\boldsymbol{A}_1 = \begin{bmatrix} 1.0 \end{bmatrix}, \boldsymbol{A}_3 = \begin{bmatrix} 1.2 \end{bmatrix}, \boldsymbol{A}_4 = \begin{bmatrix} 1.3 \end{bmatrix},$$
$$\boldsymbol{B}_1 = \boldsymbol{B}_3 = \boldsymbol{B}_4 = \begin{bmatrix} 1.0 \end{bmatrix}, \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1.0 \end{bmatrix}.$$

Moreover, the selected parameters to determine the stabilizing feedback gain are given as:

$$\boldsymbol{Q}_1 = \boldsymbol{Q}_3 = \boldsymbol{Q}_4 = \begin{bmatrix} 100.0 \end{bmatrix}, \boldsymbol{Q}_2 = \begin{bmatrix} 5000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{R}_1 = \boldsymbol{R}_2 = \boldsymbol{R}_3 = \boldsymbol{R}_4 = \begin{bmatrix} 1.0 \end{bmatrix}.$$

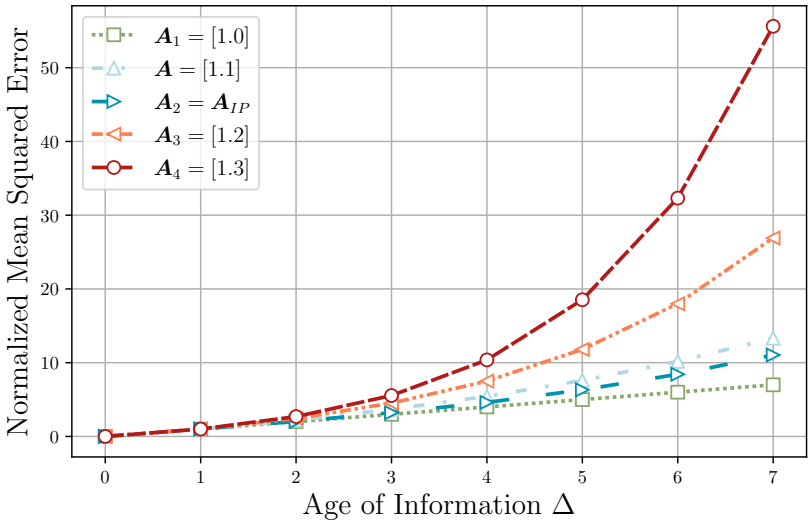


Figure 4.21: The evolution of the normalized MSE for increasing AoI. For the scalar systems, i.e., $i \in \{1, 3, 4\}$, as the system matrix $\boldsymbol{A}_i$ increases, a given Δ leads to a higher squared error in expectation. Additionally, the inverted pendulum's (IP) normalized MSE behavior is comparable to an imaginary scalar sub-system with $\boldsymbol{A}_i = [1.1]$ and $\boldsymbol{\Sigma}_i = [1.0]$.

Our results consist of 50 simulation runs that are each $T = 180\ 000$ time slots long. The finite horizon H is varied from zero to five. When $H = 0$ is selected, we employ the greedy MAF policy described in section 4.4.2. Similarly, the greedy MEF scheduler corresponds to a combination of $H = 0$ with the metric $nMSE_i(\Delta_i(t))$.

As our primary goal is reducing the cost at the destination, for the cost function (4.95), we penalize controller's contribution more than the cost at BS by using a higher weight as $w_i^{BS} = 1$, $w_i^{\mathcal{C}_i} = 2$ for all i. In addition, the following parameters are used for the Gilbert-Elliott channel model: $p_G = 0.1$, $p_B = 0.4$, $p_{G2B}^{UL} = p_{g2B}^{DL} = p_{B2G}^{UL} = p_{B2G}^{DL} = 0.1$.

Fig. 4.21 illustrates the evolution of the normalized Normalized Mean Squared Error (NMSE) for increasing AoI. As depicted in the figure, each sub-system considered in our simulations has a different expected NMSE trajectory for increasing AoI. For instance, at $\Delta = 7$, the NMSE is expected to be much higher for $\boldsymbol{A}_4$ than for A_1, indicating a higher estimation error normalized by their respective default state $\Delta = 1$. In addition, we observe that the inverted pendulum application lies between $\boldsymbol{A}_1$ and $\boldsymbol{A}_3$ w.r.t. NMSE. In fact, it is comparable to an imaginary[21] scalar sub-system with $\boldsymbol{A} = [1.1]$ and $\boldsymbol{\Sigma} = [1.0]$. This brings us to the selection α_i in our simulations and in the following discussion. That is, we select $\alpha_i = \boldsymbol{A}_i$ for $i \in \{1, 3, 4\}$ and $\alpha_2 = \alpha_I P = 1.1$ when applicable. We would like to stress that α_i appears as a multiplicative factor in front of AoI for some of the considered cost metrics, such as $g_i(t) = w_i^{BS}\alpha_i\Delta_i^{BS}(t) + w_i^{\mathcal{C}_i}\alpha_i\Delta_i^{\mathcal{C}_i}(t)$.

[21]By using the word "imaginary", our goal is to emphasize that this sub-system is not considered in our simulations. It is included in the figure only for comparison.

4.4.4.2 Key Performance Indicators

We select the long-term average AoI as the primary KPI to quantify information freshness in the network[22]. The average AoI during a simulation run is measured per time slot and user; hence it can be obtained by:

$$\overline{\Delta} = \frac{1}{T}\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1}\Delta_i^{\mathcal{C}_i}(t). \tag{4.96}$$

$\overline{\Delta}$ tells us, how outdated the most recent information at the controllers in average is. Despite being an application layer metric, the average age does not state anything about the application-specific performance beyond freshness. To that end, we measure the average NMSE and LQG cost, calculated per time slot and per user as:

$$\overline{nMSE} = \frac{1}{T}\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1} w_i \, nMSE_i(\Delta_i(t)), \tag{4.97}$$

$$\overline{\mathcal{F}} = \frac{1}{N}\sum_{i=1}^{N} w_i \mathcal{F}_i, \tag{4.98}$$

with $\mathcal{F}_i$ as in (4.71) and sub-system weight factor $w_i \geq 0, \forall i$. In this study, we assume equal weight for each loop, i.e., $w_i = 1, \forall i$. Note that a lower $\overline{nMSE}$ points to a higher estimation performance, whereas a lower $\overline{nMSE}$ indicates an increased control performance in the network.

4.4.4.3 Results

Fig. 4.22 shows the resulting information freshness performance of the considered scheduling policies. It plots $\overline{\Delta}$ against increasing H, while each curve belongs to a different scheduling policy, including MT, RR, and our proposed FH scheduler operating with various age-penalty functions. One can clearly see from the figure that the best-performing schedulers w.r.t. $\overline{\Delta}$ are the FH scheduler using linear functions of age, i.e., $\Delta_i[t]$ and $\alpha_i\Delta_i[t]$. Moreover, the MT scheduler fails to provide information freshness as it prioritizes those users with the best channel conditions, irrespective of the timeliness aspect. Although being a channel-unaware policy, RR performs significantly better than MT due to its periodic scheduling pattern. However, if we look at the FH scheduler using NMSE as the scheduling metric, we can observe that it falls behind the RR scheduler. Please note that the results for MT and RR are included multiple times in the figure for presentation purposes, although they are independent of H. In consequence, they appear as horizontal lines parallel to the x-axis.

[22]Performing better or worse in information freshness is not a success indicator for QoC. Average AoI is presented solely for more insightful discussion.

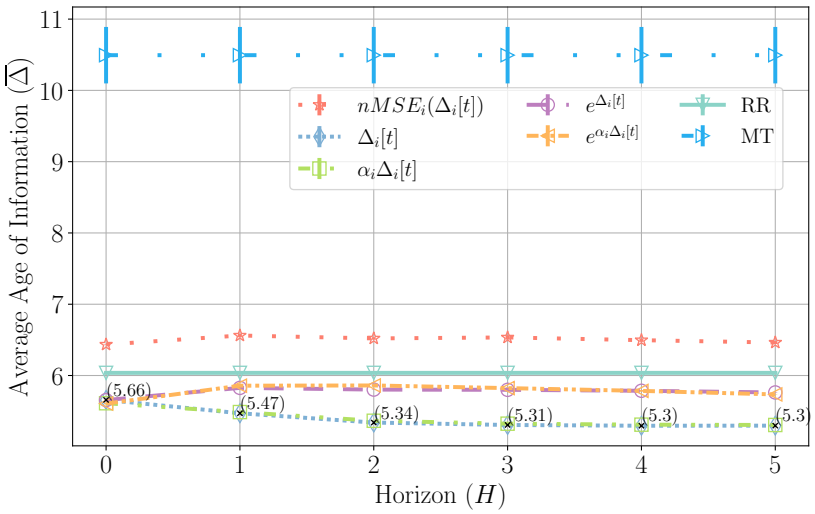


Figure 4.22: Average age of information (AoI), i.e., $\overline{\Delta}$, in the network. While maximum throughput (MT) scheduler performs the worst w.r.t. $\overline{\Delta}$, using linear age-penalty functions, i.e., $\alpha_i\Delta_i[t]$, $\Delta_i[t]$ leads to the lowest average AoI in the network. Vertical error bars represent 95% confidence interval.

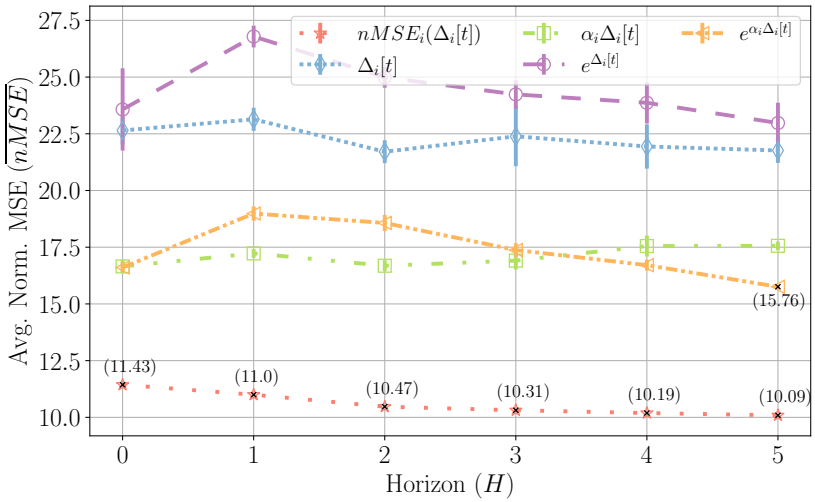


Figure 4.23: Long-term average normalized mean squared error (NMSE) in the network, as defined in (4.97). In the x-axis, the horizon parameter $0 \leq H \leq 5$ is varied. $\overline{nMSE}$ captures the mean estimation performance in the network. Vertical error bars represent 95% confidence interval.

Fig. 4.23 presents the average estimation performance for varying H captured by $\overline{nMSE}$. For presentation purposes, we intentionally left RR and MT out as they perform significantly worse than any other FH schedulers considered in this work[23]. We can easily deduct from the figure that the FH scheduler using $nMSE_i(\Delta_i(t))$ as cost function, outperforms all other schedulers. This can be accounted for the ability of the metric NMSE to identify the most valuable transmission in terms of the estimation error reduction than the other selected

[23]The average NMSE for RR and MT are $12.9 \cdot 10^4$ and $34.4 \cdot 10^{10}$, respectively.

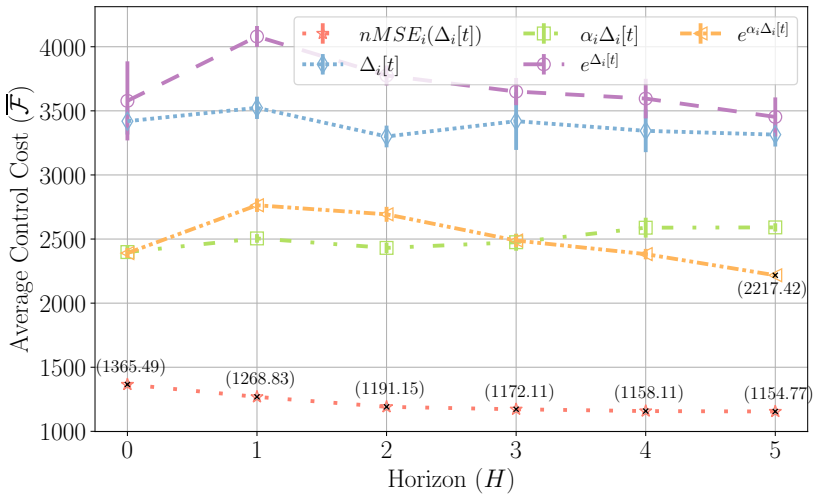


Figure 4.24: The figure shows the average control cost per sub-system given in (4.98). $\mathcal{F}$ captures the average control performance in the network. Vertical error bars represent 95% confidence interval.

functions, such as $\Delta_i(t)$. In particular, applying the MEF scheduler from [Aya+19], which coincides with the greedy version of our proposed policy, i.e., $H = 0$, outperforms its closest competitor, i.e., the combination of $H = 5$ with $e^{\alpha_i\Delta_i[t]}$, by more than 27%. Moreover, by increasing the far-sightedness of our scheduler beyond $H = 0$, we are able to improve the estimation performance further by more than 11%, compared to the MEF policy from [Aya+19]. It is important to mention that the approximating functions $\alpha_i\Delta_i$ and $e^{\alpha_i\Delta_i(t)}$ outperform the FH age-optimal $\Delta_i(t)$ scheduler. This reveals the potential in approximating the task criticality by appropriate functions for those scenarios where the exact modeling is not feasible.

Fig. 4.24 illustrates the average control cost $\overline{\mathcal{F}}$ in the network for varying H. Similar to $\overline{nMSE}$, the best-performing scheduling policy regarding the control performance is the FH scheduler using $nMSE_i(\Delta_i(t))$. In particular, when $H = 5$ is selected, the achieved $\overline{\mathcal{F}}$ is lower than the greedy scheduler proposed in [Aya+19] by approximately 15%. As in Fig. 4.23, we deduct that the utilization of $\Delta_i[t]$ and $e^{\Delta_i(t)}$, falls behind carefully selected heterogeneous cost functions, i.e., $\alpha_i\Delta_i$, $e^{\alpha_i\Delta_i(t)}$, $nMSE_i(\Delta_i(t))$. Notice that none of the selected schedulers incorporate control cost into their decision-making. However, as in the case of NMSE-based FH scheduler, the control performance can be indirectly improved as a side-product of optimizing estimation performance.

It is important to emphasize that the availability of network resources remained unchanged throughout our numerical evaluation. Only by controlling how the network resources are distributed among the users, we are able to create significant performance differences between the considered resource allocation policies. Hence, we can claim that by introducing control-awareness into decision-making, e.g., into the MAC layer protocols, one can significantly

vary the quality of the offered service. Both figures presenting the achieved $\overline{nMSE}$ and $\overline{F}$ confirm the importance of such an approach and reveal the potential of considering semantics of information and control-awareness as an alternative to increasing the bandwidth to offer the same level of service.

Remarks on inverted pendulum's performance: To demonstrate the benefit of adopting a control-aware approach over conventional MAC protocols in a more tangible way, we focus on the inverted pendulum's performance. First, let us define three success criteria specific or the maximum allowed pendulum angle $|\phi[t]|$ in degrees:

I) $|\phi[t]| < 15°|$ for $0 \leq t < T$,

II) $|\phi[t]| < 30°$ for $0 \leq t < T$,

III) $|\phi[t]| < 90°$ for $0 \leq t < T$.

The table 4.3 summarizes the number of measurement runs out of fifty, in which $|\phi[t]|$ complies with the selected criterion for all $t \in [0, T)$. It contains RR, MT, and the NMSE-based FH scheduler with $H = 5$. In simple words, the FH scheduler is able to keep the pendulum

	MT	RR	FH
I)	$0/50$	$41/50$	$50/50$
II)	$5/50$	$50/50$	$50/50$
III)	$7/50$	$50/50$	$50/50$

Table 4.3: Number of measurement runs, in which the maximum $|\phi|$ is below the upper bound.

angle within $\pm 15°$ throughout the fifty simulation runs. On the other hand, when the MT policy is selected, only seven out of fifty runs comply with criterion III). This suggests that a human observer, looking at the inverted pendulum visualized in 4.20 would see the pendulum falling in 43 runs as the pendulum angle exceeds $\pm 90°$. Moreover, the RR scheduler leads to satisfactory results in all runs only according to the second and third criteria.

Discussion on complexity: The selection of the horizon parameter H plays a central role in our approach. By increasing H, one can control the time horizon, in which the FH scheduler offers optimality for the selected age-penalty metric. However, this comes with an exponential complexity increase for the joint UL and DL scheduling problem. Particularly, the maximum number of states defining the H-stage problem for N sub-systems is formulated by the following equation:

$$\#nodes(H) = \frac{(N^2 + 2N + 1)^{H+1} - 1}{N^2 + 2N}. \tag{4.99}$$

The H parameter allows us to control up to how many nodes the tree structure should consist of, if the number of users is given. Note that when the MEF policy with $H = 0$, the tree consists of a single root node representing the current network state. The following table compares the number of nodes in the worst-case obtained by (4.99) to the average node count measured during our simulations. The difference between the worst-case and simulation

H	1	2	3	4	5
worst-case	26	651	16276	4.1×10^5	10.17×10^6
simulations	16.3	238.6	3428	4.9×10^4	7.26×10^5

Table 4.4: Node count comparison between simulations and worst-case complexity.

is caused by constraining the set $\mathcal{A}^{DL}$. As we do not consider ineligible DL actions as a candidate for the optimal policy, due to equal knowledge at the BS and a certain controller $\mathcal{C}_i$, the number of tree nodes as well as our search space is reduced.

Learning Optimal Action(s): In addition to reducing the complexity as described above, we propose a further technique to lower the barrier to adoption in practical deployment. In a static network with an invariant number of control loops, the FH scheduler can learn the optimal actions over time as the same states are revisited repeatedly. That is, if the same H-stage problem has been solved and the optimal action(s) has been determined before, it can be stored in the memory for future reference. This approach is similar to the Dynamic Programming (DP) method used in computer science, which stores solutions to sub-problems for future usage.

One way to speed up the learning process is parallel computing. In particular, our simulations employ the DP approach by running multiple threads that collectively gather a solution set in the shared memory. Fig. 4.25 demonstrates the potential usage of this technique. The upper sub-figure shows the ratio of an H-stage problem being already in the solution set, which we call a *DP hit*. If the H-stage problem is identified as novel, referred to as a *DP miss*, the optimal solution is calculated and added to the solution set afterwards. As the network learns more solutions over time, the occurrence ratio of a DP hit takes over, and a new execution of the FH scheduling algorithm becomes unnecessary. It is needless to say that storing previous solutions to already visited problems comes with an increased memory cost over time. In brief, one can say that the high computational complexity is traded for increased memory demand.

As an important remark, we would like to mention that we are able to identify a problem reappearing due to the limited variance in the link quality, thanks to the Gilbert-Elliott model. That is, users experience a dynamic packet loss probability on each link that alternates between

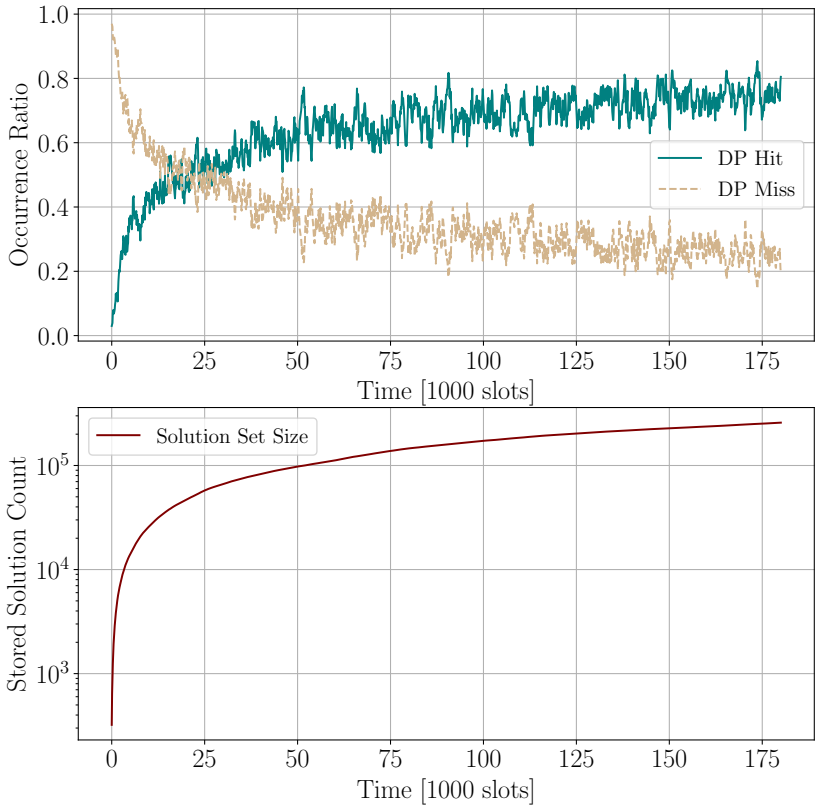


Figure 4.25: The upper sub-figure shows the ratio of a problem being already in the solution set by using dynamic programming method (DP) and the occurrence ratio of a new problem. The lower sub-figure shows the growing solution set size.

two values, i.e., p_G and p_B depending on the link state. However, each of these probabilities could take any value between zero and one in a real network. Consequently, the H-stage problem is highly likely to be a novel problem, leading to a DP miss. Nevertheless, this issue motivates the consideration of approaches similar to [HH08]. In [HH08], the authors adjust the parameters of a two-state Gilbert-Elliott model through adapting the coefficient of variation identified in the real network trace. They demonstrate that the model they obtain after the parametrization step captures the loss pattern of the real network fairly well.

Discussion on the effect of H: As we have just mentioned, the growth in computational complexity is exponentially increasing in H. However, we cannot claim the same for the improvement in performance. Fig. 4.26 shows the achieved estimation performance for our NMSE-based FH scheduler in detail. We observe that for increasing H, the estimation performance is indeed improved, identified by a decrease in $\overline{nMSE}$. However, a significant portion of the gain is achieved already when H reaches three. As we go beyond $H = 3$, the benefit of looking into a longer time horizon diminishes. Therefore, the right H value should be identified prior to deployment if the complexity constitutes a bottleneck for our scenario.

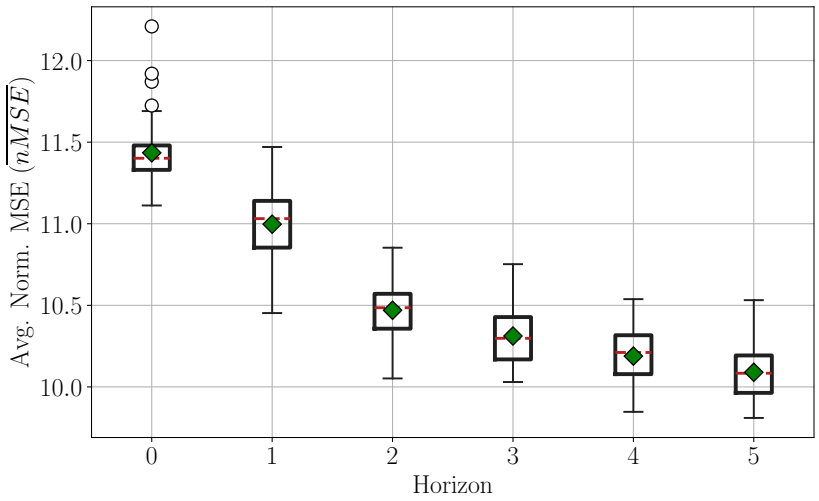


Figure 4.26: Detailed average normalized mean squared error NMSE results for the FH scheduler with $g_i(t) = w_i^{BS}\alpha_i\ nMSE_i(\Delta_i^{BS}(t)) + w_i^{\mathcal{C}_i}\alpha_i\ nMSE_i(\Delta_i^{\mathcal{C}_i}(t))$. As the scheduler becomes more far-sighted, the estimation performance captured by $\overline{nMSE}$ improves.

Simulation versus analytic methods: As an alternative to comparing scheduling policies through simulations, one could argue for finding an analytical solution to the cost minimization problem characterized in (4.97). For instance, given a stationary scheduling policy, if we can obtain a stationary distribution of AoI, we can easily map it to the average NMSE, as we have shown in section 3.2. However, doing so, we can only get partial answers, e.g., unclear mapping between AoI distribution and the average LQG cost from (4.98). In addition, analytic methods may introduce significant difficulties without making substantial contributions to the core problem. On the contrary, a simulation-based study, as we have conducted in this work, gives definitive answers to the KPIs of interest. Additionally, simulations serve as a proof that it is practically feasible to implement the proposed algorithm in software.

4.5 Related Work

The existing literature related to the content of this chapter can be roughly divided into two main blocks. The first group of works study the age of information and further semantics of information beyond age for real-time applications communicating over networks. The second group focuses on networking protocols specifically for networked control systems.

Age of information and beyond: The AoI has emerged as a novel metric in real-time networked systems as a metric capturing the freshness aspect of information and has received great attention by the research community [Yat+21; Pop+22; Kad+18]. It has been used in remote monitoring and control-oriented tasks, such as UAVs and autonomous driving

[Sor+21; Han+22; AED19]. Note that a detailed literature review on AoI has already been provided in chapter 2.

Although it has been widely adopted by the research community, the AoI has been shown to be sub-optimal for decision-making in certain settings. In particular, by definition, the AoI is agnostic to the value and content of the transmitted bits beyond their freshness property. Hence, derivatives of AoI has emerged, to capture further semantics of information beyond age. To name a few examples, the *value of information of update (VoIU)* has been proposed in [Kos+17] to overcome the shortcomings of AoI by considering *non-linear functions of AoI*. [SC19; Sun+17; Maa+20] are other examples using derivatives and non-linear functions of AoI, e.g., $f(\Delta(t)) = e^{a\Delta(t)}$, in order to approximate the importance of transmitting the next status update packet. In [Maa+20], the authors propose a novel metric called the *age of incorrect information (AoII)* that considers an update "informative" only if the system state has altered since the most recent update. The works mentioned above consider information semantics, e.g., VoIU, AoII, as an alternative to aging as a linear process. Generally speaking, using derivatives of AoI can be seen as *heuristics* towards system-dependent and task-specific metrics.

An alternative approach is using derivatives of age that depend on the time dynamics of the monitored process', as it has been done in [SPU20; Kl19; Cha+19]. In contrast to these works, the content of this chapter is targeted for multi-user scenarios, i.e., when there are multiple feedback control loops sharing a communication network. For example, [Cha+19] focuses on control performance in a single-loop scenario, in which the authors target the sampling problem under constant loss probability and propose a greedy policy based on the estimation error.

Networked control systems: The vast majority of the existing works on networked control systems focuses on stability conditions in the presence of a network, including but not limited to [Cer03; Dre+05; LFJ15; Mai+22]. An alternative methodology is followed by [LG04; GRP15; MH14; Vil+16], which study the achieved control performance under contention-based MAC protocols. However, contention-free access to the shared wireless medium has been found to outperform contention-based strategies through a numerical study in [LG04], hence motivating the centralized scheduling in multi-loop WNCS scenarios.

Centralized resource allocation for WNCS have been studied in [WY01; WYB02; MEJ19; ZZN20]. In [ZZN20], a multi-user scenario is considered, and a new metric called *urgency of information* has been proposed. While [WY01] and [WYB02] propose a greedy policy, scheduling the sub-system with the highest instantaneous estimation error, [MEJ19] solves the problem for a given time horizon optimally, when there are no delays or losses in the network. It is important to mention that all of these works assume global knowledge for

decision-making. In other words, the centralized scheduler that is not co-located with the sensor devices is assumed to have global and perfect knowledge about the content of the information to be sent. On the contrary, the scheduling algorithm presented in this chapter bases its decision-making on the expected value of the MSE or NMSE given the AoI.

4.6 Summary

This chapter has focused on the centralized scheduling of wireless network resources for networked control systems. The core research question that this chapter aims to answer is the following: How to distribute the limited network resources among multiple heterogeneous control loops in order to keep performance deterioration at the minimum? Our starting point has been the application of AoI in centralized scheduling, e.g., [Kad+18]. Motivated by the application-unawareness of AoI, in section 4.1, we have employed the expected value of the mean-squared (estimation) error as the scheduling metric. We have shown through simulations that although the MSE-based scheduling lacks behind the pure AoI-based scheduling in terms of information freshness, it achieves a higher performance w.r.t. control-oriented tasks accounted for its ability to capture the real value behind the transmitted bits. In section 4.2, we have introduced constant packet loss into the network and solved the problem optimally using the value iteration technique. The assumption of packet loss to be time-invariant has been relaxed in section 4.3, in which we obtained the optimal scheduling policy for a given finite horizon. We have shown through simulations that the estimation performance can be improved by increasing the finite horizon for the price of increased computational complexity. Section 4.4 extends the results of section 4.3 by considering a two-hop star network instead of a single-hop link between the source-destination pairs. Additionally, the last section contains a comparison between our proposed AoI-based control-dependent metric to heuristic derivatives of age, e.g., $f(\Delta(t)) = e^{a\Delta(t)}$ that have been used in the existing literature. We have shown through simulations that if designed carefully, such heuristic functions can outperform pure AoI-based scheduling w.r.t. control performance but lacks behind using metrics that are specific to the control system dynamics.

Chapter 5

Implementation of Control-Aware Scheduling for WNCS

We have shown in the previous chapter that semantics of information such as freshness and value are particularly relevant in scenarios with limited resources, e.g., for WNCS, when multiple feedback control loops share the wireless network. The previous chapter has revealed that the consideration of the VoI for network and protocol design is a very effective tool for minimizing the deterioration in control systems' performance caused by the limitations of the imperfect communication network. However, incorporating the SoI into the network design may unfold new challenges that may not be straightforward to tackle in practice.

For instance, suppose a wireless network that is shared by multiple feedback control loops. As in the previous section, the allocation of network resources is done by a centralized scheduler. In such a setting, having the QoC maximization as the primary goal in mind, the scheduler does not only need to measure the channel quality or keep track of the offered throughput, but it also requires a control-specific evaluation mechanism to be able to prioritize the most valuable information. An example to such has been shown in the previous section and the results have shown that a significant performance improvement can be achieved. However, when it comes to the realization of such a cross-layer and complex algorithm in practice, the random nature of the wireless channel, approximation and hardware effects, as well as timing errors, may obscure the performance improvement promised by the simulation results. Hence, the additional effort spent on the protocol design may be superfluous.

In this chapter, we extend our theoretical results especially from section 4.1 through an extensive experimental study focusing on existing MAC protocols. More specifically, in a multi-loop scenario, we study the effect of various scheduling policies on the resulting control performance. The core question of this chapter is the following: *Despite all the non-idealities and randomness introduced by the physical channel and hardware components, is*

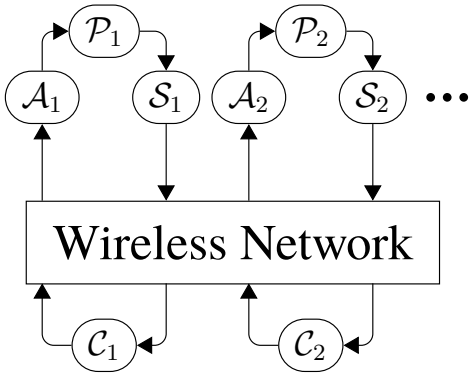


Figure 5.1: Multiple feedback control loops share a wireless communication network in order to exchange information between their components. We consider that each plant $\mathcal{P}_i$, sensor $\mathcal{S}_i$, and actuator $\mathcal{A}_i$ are co-located, whereas the controller $\mathcal{C}_i$ operates remotely.

it still possible to achieve a notable control performance improvement in practice through control-aware scheduling policies that heavily rely on theoretical and control-specific system modeling? In the remainder of this chapter we seek an answer to this question by conducting experimental measurements on a practical testbed. For the information exchange between sensor-controller pairs, we utilize SDRs that are programmed using the GNU Radio library [Rad].

Section 5.1 investigates the effect of the selected queueing strategy at the source devices on information freshness and control performance. The results of section 5.1 are based on the publication [AOK21]. Section 5.1 serves as a baseline for the following section, i.e., section 5.2, in which we conduct an extensive study on selected MAC protocols. The results of section 5.2 have been published in [Aya+22]. A related work section on relevant practical research is provided at the end of this chapter.

5.1 Effect of the Selected Queueing Strategy on Information Freshness and Quality of Control

5.1.1 Scenario

We consider multiple control systems that are closed over a shared wireless network. As visualized in Fig. 5.1, each loop consists of a plant $\mathcal{P}_i$, a sensor $\mathcal{S}_i$, a controller $\mathcal{C}_i$, and an actuator $\mathcal{A}_i$. Each sensor measures the plant state periodically and transmits the observed state over the shared wireless channel to $\mathcal{C}_i$. The transmissions occur in the form of status update packets, while each packet contains a single state information. Having received a new update, each controller responds with a packet carrying the newly calculated control input to

$\mathcal{A}_i$. The commanded input is then applied by the actuator to the plant with the goal of driving the system state to zero.

We assume a request-response system in which the controller transmits only upon a reception of a sensor measurement, but not triggering any transmissions itself. Each sensor-to-controller link, as well as the controller-to-actuator link requires a single transmission, hence the loop is closed over a two-hop wireless communication network. The formation control of UAVs from a ground station can be given as a practical example of such a topology with the UAVs being the plant, its positioning sensors being the sensors, the ground station corresponding to the controller, and a set of motors installed on the UAVs serving as the actuators.

The control sub-systems are characterized by a set of LTI matrix equalities in discrete-time:

$$\boldsymbol{x}_i[k+1] = \boldsymbol{A}_i\boldsymbol{x}_i[k] + \boldsymbol{B}_i\boldsymbol{u}_i[k] + \boldsymbol{w}_i[k], \tag{5.1}$$

with the variables defined as in (4.69). Upon receiving a system state $\boldsymbol{x}_i[k - \Delta_i^{\mathcal{C}_i}[k]]$, the controller calculates the control input according to the following control law:

$$\boldsymbol{u}_i^{\mathcal{C}_i}[k] = -\boldsymbol{L}_i\boldsymbol{x}_i[k - \Delta_i^{\mathcal{C}_i}[k]]. \tag{5.2}$$

Here, $\boldsymbol{L}_i \in \mathbb{R}^{m_i \times n_i}$ denotes the time-invariant feedback gain matrix. Note that in contrast to the previous chapters, the control law is kept simple in the sense that there is no ongoing state estimation at the controller side. That is, although $\mathcal{C}_i$ may have received an outdated observation of the state that is already $\Delta_i^{\mathcal{C}_i}[k]$ sampling periods older than k, the latest received state is directly used to obtain the next control input. Additionally, we do not assume any intelligence at the actuators, thus the latest received control input held constant until a new packet is received. In other words, $\boldsymbol{u}_i[k] = \boldsymbol{u}_i[k-1]$ holds in case there has not been any successful reception in the controller-to-actuator link during the sampling period $k-1$. The behavior of the input applied to the plant can be characterized by the following equation:

$$\boldsymbol{u}_i[k] = \begin{cases} \boldsymbol{u}_i[k-1] & \text{, if no reception during the sampling period } k-1 \\ -\boldsymbol{L}_i\boldsymbol{x}_i[k - \Delta_i[k]], & \text{, if a new transmission on the } \mathcal{C}_i\text{-to-}\mathcal{A}_i \text{ link} \end{cases} \tag{5.3}$$

with $\boldsymbol{x}_i[k - \Delta_i[k]]$ being the system state that was used to obtain the latest control command received by the actuator. Put differently, $\boldsymbol{x}_i[k - \Delta_i[k]]$ denotes the freshest plant state that $\mathcal{C}_i$ has successfully received, processed, and forwarded to $\mathcal{A}_i$ by the beginning of the k-th sampling period[1]. We emphasize that the information content changes from state to control input along the $\mathcal{S}_i$-to-$\mathcal{C}_i$-to-$\mathcal{A}_i$ path and only the final applied control input, i.e., $\boldsymbol{u}_i[k]$, has an effect on the state evolution. Equations (5.2) and (5.3) imply that the control signal at $\mathcal{C}_i$, as well as the actuated signal at $\mathcal{A}_i$ may be obtained using an outdated knowledge of the state.

[1]Throughout the following analysis, we assume that a control input can be used by $\mathcal{A}_i$ in time step k only if it has been received before the beginning of the k-th sampling period.

5.1.2 Design and Implementation

Our goal is to operate the LTI feedback control loops introduced in the previous section through a physical channel. To that end, we develop a framework programmed in Python language implementing the control applications in real-time. For the networking part, we take the framework presented in [Blo+13] as a starting point, which contains various GNU Radio modules destined to run on SDRs. In particular, the framework replicates the physical layer of the IEEE 802.15.4 standard and can interfaces any other application process, e.g., a Python code realizing the plant process, through User Datagram Protocol (UDP) sockets.

5.1.2.1 Medium Access Control

The framework that has been proposed in [Blo+13] does not contain any MAC considerations. More specifically, the MAC layer processing is implemented in a way that any incoming packet is immediately forwarded down to the physical layer processing block that is responsible for their transmission on the wireless link. Such a MAC protocol is also called ALOHA in the literature. As we know from the fundamental rules of wireless communications that the ALOHA protocol causes collisions due to simultaneous access to the shared channel, consequently leading to high packet loss. In order to minimize the amount of such collisions, we implement an second protocol that relies on centralized resource allocation based on time slots. The transmission schedule is communicated using *beacon packets* that are broadcasted by a gateway node periodically. Among other fields required for data-link layer processing, such as packet header, Cyclic Redundancy Check (CRC), each beacon packet contains the duration of a time slot, denoted by T_D, and the transmission schedule for the next L slots forming a *frame* together[2].

The beacon packets are also used for time synchronization among the SDRs. In particular, having received a new beacon packet, each sensor aligns its timing in order to synchronize with the slot structure. When a new time slot begins, the node that is allowed to transmit according to the latest broadcasted beacon packet initiates the transmission of the next packet in its transmission buffer. The gateway node, also implementing the centralized scheduler in our scenario, allocates one sensor node at a time. Therefore, all other SDRs except the scheduled user refrain from accessing the channel. The controllers run as parallel processes and are co-located with the gateway node. Alternatively, one can think of controllers and the scheduler as one node. The scheduling policy considered in this analysis is the *round-robin*

[2]If a sensor node fails to decode a beacon packet, the transmission schedule from the previous frame is assumed to apply for the current frame.

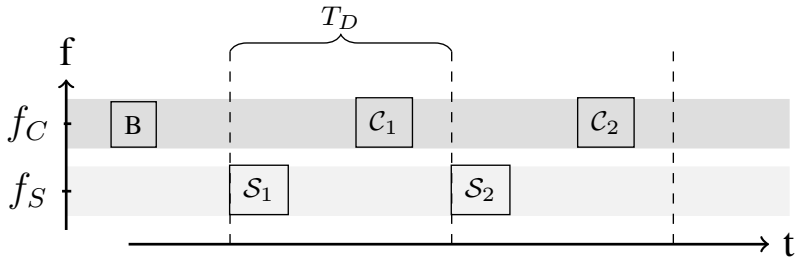


Figure 5.2: An example packet exchange between the gateway and two control sub-systems. In the figure, $\mathcal{S}_1$ and $\mathcal{S}_2$ stand for the status update packets sent by the two sensor nodes. Similarly, $\mathcal{C}_1$ $\mathcal{C}_2$ represent the control inputs that are sent as a response to the both status update packets. Note that a slot duration is long enough to accommodate a status update, as well as a control input packet. B stands for the beacon packet. While the gateway node uses the center frequency of f_C, the sensors are sent on a different frequency f_S.

(RR) scheduler. This means that each slot is allocated to a sensor according to a predetermined order[3].

In addition to the beacon packets, there are two other packet types in our network. The *status update packets* are transmitted from sensors to the controller processes. They contain a unique identifier of the sub-system, i.e., i, the generation timestamp indicating the sampling period κ, to which they correspond to, and the system state $\boldsymbol{x}_i[\kappa]$. Each received status update packet triggers the transmission of a new *control input packet* carrying the controller identifier i, the sampling period κ, and the control input to be applied by the respective actuator $\boldsymbol{u}_i[\kappa] = -\boldsymbol{L}_i\boldsymbol{x}_i[\kappa]$. Our implementation utilizes Frequency Division Duplex (FDD) technique, i.e., both the beacon and control packets are transmitted on a distinct frequency from status update packets. Fig. 5.2 depicts an example information exchange between the gateway node and two sensor nodes. An initial beacon packet is followed by a status update packet transmitted by $\mathcal{S}_1$. In the same slot, a control packet is generated by $\mathcal{C}_i$ and sent via the gateway node to the sensor. The same procedure repeats in the subsequent slot.

5.1.2.2 Packet Management Strategies

To understand the connection between the selected packet management policy and the resulting performance, we consider various strategies that are commonly used in the literature:

- First Come First Serve with Tail Drop (FCFS-TD)
- First Come First Serve with Head Drop (FCFS-HD)
- Last Come First Serve with Packet Drop (LCFS-PD)

[3]We use the sub-system identifiers for that purpose and schedule the sub-systems in the increasing order, i.e., sub-system two with $i = 2$ is scheduled following the first sub-system, $i = 1$.

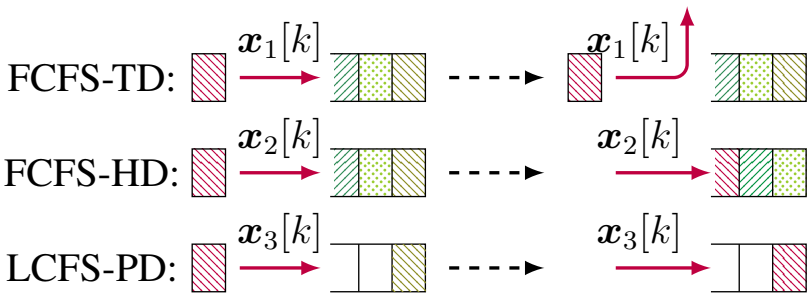


Figure 5.3: The visualization of the FCFS-TD, FCFS-HD, and LCFS-PD strategies with a transmission buffer size of three packets. While the FCFS-TD and FCFS-HD strategies discard the most recent and oldest packet available, respectively, according to the LCFS-PD policy, any older packet is replaced with a new one.

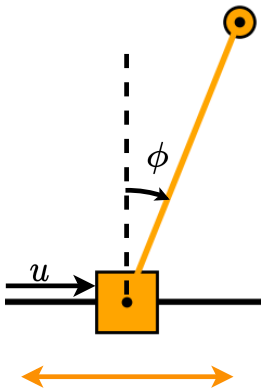


Figure 5.4: Illustration of an inverted pendulum. The control input $\boldsymbol{u}$ is applied to move the cart back and forward in order to keep the pendulum vertical. The dynamics of the system can be linearized around the equilibrium point, which is located at $\phi = 0$.

In our framework, all of these policies are implemented using transmission queues of finite length. In case of the FCFS-TD strategy, if the queue is full and there is a new packet received from the application layer, then the new packet is dropped without further consideration. When the FCFS-HD policy is applied, a new arrival to a full queue leads to the discard of the oldest packet. Note that the oldest packet is located at the head of the transmission queue. Both of these strategies policies inject any incoming packet to the back of the queue, also called the *tail* of the queue. However, when the the LCFS-PD policy is employed, the packet at the queue's head is discarded prior to the admission of the new arrival. This way, the LCFS-PD serves only the most recent packet and discards all admitted ones. Fig. 5.3 depicts the operation of the packet management policies.

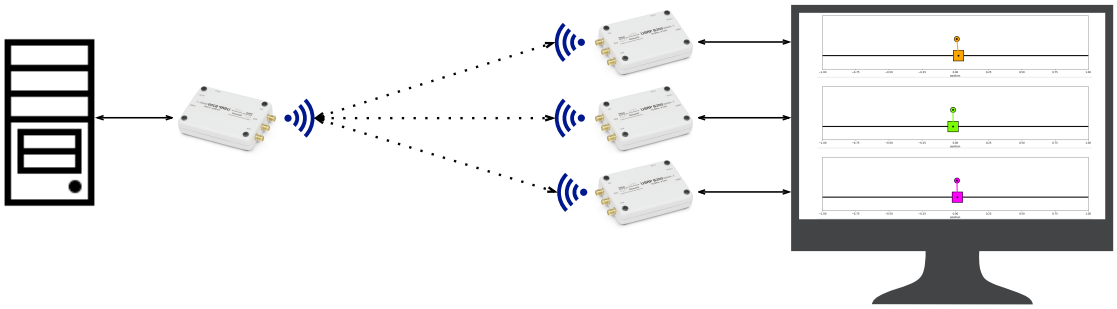


Figure 5.5: A sketch of our measurements setup composed of four USRP B200mini-i SDRs. Three SDRs are used for sensor-to-controller data transmission and one SDR is used for the controller-to-actuator traffic. On the monitor, the current state of the inverted pendulums are visualized via a graphical interface.

5.1.2.3 Case Study: Inverted Pendulum

As a case study, we choose the task of controlling an inverted pendulum over the wireless network. Fig. 5.4 shows a screenshot taken from our graphical user interface emulating an inverted pendulum. The controller's objective is to hold the pendulum in upright position through the movements of the cart. Accounted for its unstable nature, as the pendulum would fall if not controlled appropriately, it is an example control application that requires fresh information to operate correctly. The discrete-time behavior is characterized by a set of matrix equations as given in section 4.4.4.1. It is important to mention that the specified matrices describe the behavior of the system for a sampling frequency of 100 Hz. Equivalently, in our framework, the system state is sampled every ten milliseconds and a new status update packet is admitted into the network.

Our setup consists of three inverted pendulum processes that run in parallel on a desktop computer. Each process is mapped to a different Software-Defined Radio (SDR) and the traffic that is generated by the inverted pendulum process is sent over the shared wireless channel to the gateway node. Fig. 5.5 sketches our measurement setup with three applications running on one computer and the controller running on a separate computer. As SDR Ettus Research's USRP B200mini-i is used.

Each plant $\mathcal{P}_i$ is sampled with a constant frequency of 100 Hz and as slot duration 8 ms is selected. The length of the frame is selected as $L = 30$ meaning that the RR scheduler determines the transmission schedule for the next thirty slots. Note that each sub-system is granted channel access every three slots as a result of the RR policy.

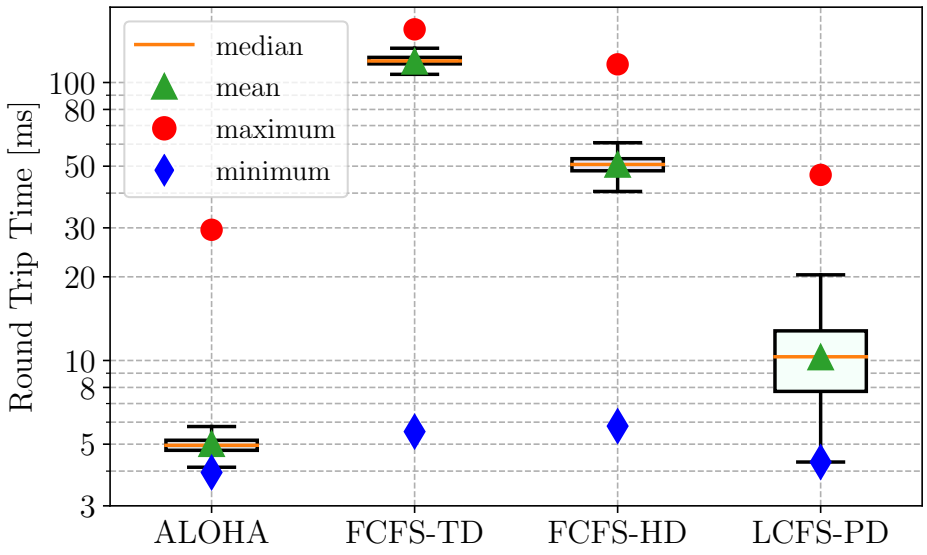


Figure 5.6: Round Trip Time (RTT) measured in application layer. ALOHA does not include any MAC layer considerations or queueing. FCFS-TD, FCFS-HD, and LCFS-PD are combined with RR scheduler. Outliers are not displayed to avoid visual clutter.

5.1.3 Results

We compare the selected packet management strategies, namely, FCFS-HD, FCFS-TD, and LCFS-PD in terms of performance. To that end, we measure average packet loss, Round Trip Time (RTT), and AoI, when a different policy is employed. Moreover, to verify that our RR scheduler helps reducing the packet loss rate, we keep track of the selected performance metrics when the ALOHA policy is used. Each configuration is measured for twenty seconds and is repeated forty times.

Fig. 5.6 shows the RTT performance of the selected policies, while the measurements are recorded in the application layer and only for those packets that are responded by a control packet. Thus, we define RTT as the time between the generation of a status update packet and the reception of the corresponding control packet with the same sequence number[4]. The results show that ALOHA performs the best in terms of delay accounted for its zero waiting time. It is followed by the LCFS-PD strategy accounted for its packet replacement feature. The worst strategy in terms of delay is FCFS-TD, as it discards the most recent packet if the queue is not empty.

If we look at average packet loss, the ALOHA protocol performs the worst, as shown in Fig. 5.7. More specifically, while the overall average loss rate is approximately 40%, we observe that in at least one run, the ALOHA strategy has encountered an average loss rate

[4]We do not need any synchronization to measure the RTT as the sending, as well reception timestamps are taken on the same computer.

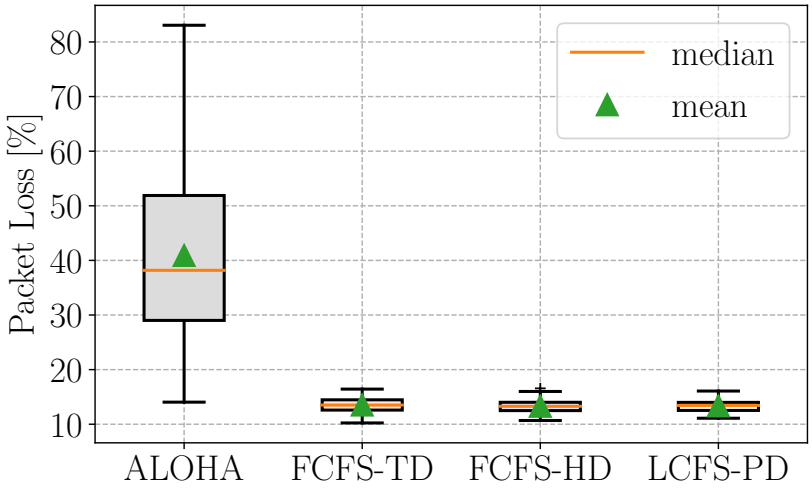


Figure 5.7: Packet loss rate for ALOHA and RR scheduler combined with FCFS-TD, FCFS-HD, and LCFS-PD strategies. The packet loss is significantly lower when the resources are allocated centrally.

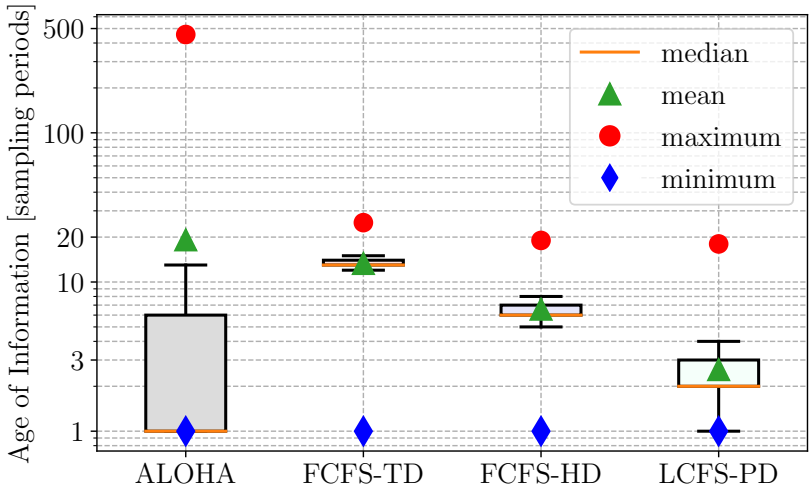


Figure 5.8: The performance in terms of AoI measured in units of sampling periods. The data set contains raw age values and not mean values only. Outliers are not displayed due to presentation purposes. The circle markers correspond to the maximum and minimum values that have been achieved throughout the measurements.

more than 80%. On the other hand, the centralized scheduler can achieve a significantly lower packet loss rate compared to ALOHA irrespective of the selected queueing policy. Note that the discarded packets do not contribute to packet loss statistics.

As we have seen in figures 5.7 and 5.6, employing the ALOHA protocol achieves a lower RTT, whereas the chance of a collision between two transmitters is significantly higher. Similarly, the additional effort put into the centralized scheduling protocol decreases the packet loss, but an additional queueing delay is introduced into the the network. Having said that, we present Fig. 5.8 that shows the average AoI in the network that is affected

both by packet loss and RTT. The results suggest that transmitting only the most recent information in a contention-free manner, i.e., RR combined with LCFS-PD, outperforms other policies prioritizing stale packets, such as FCFS-TD and FCFS-HD. Our experimental results coincide with the theoretical findings in [KYG12b]. Furthermore, providing regular updates is essential for reducing information staleness at the receiver, as it can be seen from the comparison between ALOHA and LCFS-PD with RR scheduling.

Let us now have a closer look at the performance beyond freshness and investigate the ability of inverted pendulum applications balancing themselves successfully. To do so, we select ± 20 degrees as the maximum allowed pendulum deviation from the upright position. Moreover, we call the time it takes the applications to violate this criterion *time to failure*. Table 5.1 summarizes the results with the infinity sign representing a successful control within the given bounds throughout the measurements. The results show that the RR scheduler with LCFS-PD is the only one among the considered techniques that is able to keep the pendulum angle within the specified bounds. In addition, when ALOHA is applied, the mean time to failure is 4.96 seconds and it was able to satisfy the control criterion at least once. On the other hand, the FCFS-HD and FCFS-TD policies were able to comply with the stability criterion at most 1.24 and 0.93 seconds long, respectively. Thus, we conclude that the performance and stability of our considered applications are more sensitive to receiving outdated information than infrequent exchange of data. However, we emphasize that this is strictly scenario and application dependent.

	Time To Failure		
Policies	***Mean***	***Median***	***Maximum***
ALOHA	$4.96\ s$	$2.73\ s$	∞
FCFS-TD	$0.55\ s$	$0.53\ s$	$0.93\ s$
FCFS-HD	$0.81\ s$	$0.81\ s$	$1.24\ s$
LCFS-PD	∞	∞	∞

Table 5.1: Mean, median, and maximum time to exceed ± 20 degrees of pendulum angle, referred as time to failure. To obtain the mean and median values, only the failed runs are taken into account. The RR scheduler with LCFS-PD policy was able to meet the requirements in each run.

5.2 Practical Study on Various MAC Protocols for Networked Control Systems

In the previous section, we have introduced an experimental framework that enables us to conduct measurements using real-life connections. The first results have shown the importance

of the selected packet management technique, but also of reducing the collisions through a centralized resource management scheme. In this section, we explore various scheduling mechanisms from the existing literature, such as maximum throughput, round-robin, slotted ALOHA, and few other strategies designed to increase information freshness. Moreover, as the primary contribution, we implement control-aware scheduling in practice and demonstrate its effectiveness over the control-unaware methods. The content of this section is primarily based on [Aya+22] and can be seen as an extension of the previous sections 4.1 and 5.1.

5.2.1 System Model

We consider N control applications of heterogeneous type sharing a physical wireless channel. Each application consists of a plant $\mathcal{P}_i$, a sensor $\mathcal{S}_i$, and a controller $\mathcal{C}_i$. Sensors observe the system state of their respective plant and transmit these to a Gateway (GW) node using the shared network resources. In particular, the status updates carrying the state information are sent via an SDR implementing a sensor $\mathcal{S}_i$ to another SDR implementing the GW[5]. The successfully received packets are then forwarded using a wired link to the corresponding controller $\mathcal{C}_i$. As a practical example of such a topology, we can name the camera-based control of an inverted pendulum, in which a camera observes the system state remotely and transmits these over a shared wireless network to the controller that is co-located with the plant. Such a topology can be viewed as N source nodes contending for wireless resources to transmit their status updates over a shared single-hop link. Fig. 5.9 visualizes the described scenario.

From the theoretical AoI research, e.g., [KYG12b], we know that under the assumption that the status is Markovian, having received an update, the controller does not benefit from receiving an older observation. Therefore, any older packet is considered as obsolete and non-informative, motivating the replacement of older packets by new information. Such a policy has been implemented in section 5.1 and referred to as LCFS-PD. In compliance with the theoretical findings, this approach has been found to outperform FCFS techniques, both in section 5.1.3 and recent prior work [KRM21].

We utilize digital representation of control applications running as independent parallel processes. Their behavior is modeled as discrete-time LTI systems, meaning that the system state evolves in discrete-time steps with a constant sampling period of $T_{i,s}$. In other words, if we index time steps using t, any consecutive time steps t and $t+1$ are $T_{i,s}$ seconds apart in continuous-time and the system state is considered constant between two subsequent sampling events. The discrete-time difference equation that characterizes the system state's evolution

[5]Note that the GW node is common for all sub-systems.

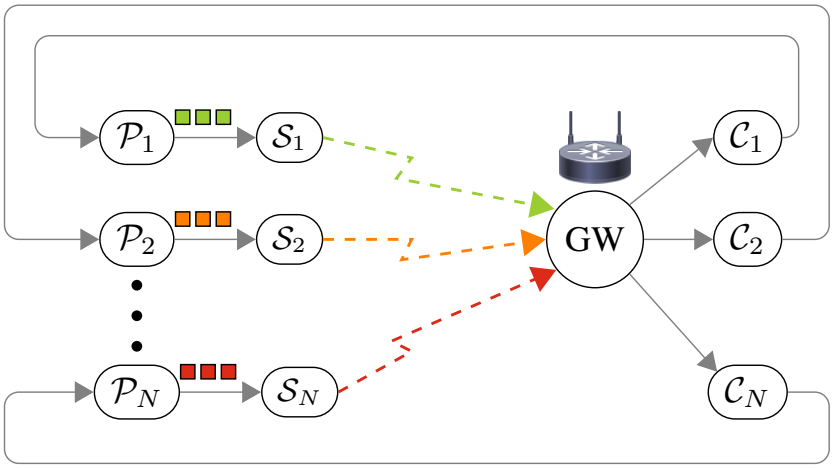


Figure 5.9: The considered scenario with N feedback control loops closed over a shared wireless link. Each sensor/SDR $\mathcal{S}_i$ is responsible for observing and transmitting the system state to the gateway GW, from where they are forwarded to the corresponding controller $\mathcal{C}_i$. While solid arrows stand for ideal links, the dashed lines indicate an imperfect wireless connection between the components.

is given as:

$$\boldsymbol{x}_i[t+1] = \boldsymbol{A}_i\boldsymbol{x}_i[t] + \boldsymbol{B}_i\boldsymbol{u}_i[t] + \boldsymbol{w}_i[t]. \tag{5.4}$$

Here, $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$ and $\boldsymbol{u}_i \in \mathbb{R}^{m_i}$ are column vectors denoting the plant state and control input, respectively. The system matrix $\boldsymbol{A}_i \in \mathbb{R}^{n_i \times n_i}$ defines the linear relationship between the current and next states. Furthermore, $\boldsymbol{B}_i \in \mathbb{R}^{n_i \times m_i}$ is the input matrix and $\boldsymbol{w}_i \in \mathbb{R}^{n_i}$ is the independent and identically distributed (i.i.d.) noise sequence that follows the zero-mean Gaussian distribution, i.e., $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_i)$. Here, $\boldsymbol{\Sigma}_i$ is the noise covariance matrix, which is a square diagonal matrix. To simplify the following analysis, we select the sampling period to be identical for all sub-systems, i.e., $T_{i,s} = T_s, \forall i$.

The calculation of $\boldsymbol{u}_i[t]$, which is needed to drive the state to the desired value, depends on the observation history at $\mathcal{C}_i$. However, due to the fact that the state information is communicated over the imperfect wireless link, namely the $\mathcal{S}_i$-to-$\mathcal{C}_i$ link, packet loss events and/or the resource scarcity prevents the controller to retrieve all generated data. Besides, the successfully received packets may be delivered with a non-negligible end-to-end delay that is caused by the entire communication stack between the plant and controller processes. In consequence, the controller's knowledge about the remote state gets outdated, leading to sub-optimal control inputs; hence to the degradation of the control performance.

In order to minimize the adverse effects caused by information staleness, each controller employs an estimator, the task of which is to estimate the actual system state remotely. The expected value of the remote state can be expressed given the latest received state information

$\boldsymbol{x}_i[\nu_i(t)]$ as follows:

$$\begin{aligned}\hat{\boldsymbol{x}}_i[t] &\triangleq \mathbb{E}\left[\boldsymbol{x}_i[t] \mid \boldsymbol{x}_i[\nu_i(t)]\right] \\ &= \boldsymbol{A}_i^{\Delta_i(t)}\boldsymbol{x}_i[\nu_i(t)] + \sum_{q=1}^{\Delta_i(t)} \boldsymbol{A}_i^{q-1}\boldsymbol{B}_i\boldsymbol{u}_i[t-q].\end{aligned} \tag{5.5}$$

$\Delta_i(t) \triangleq t - \nu_i(t)$ gives the number of elapsed sampling periods since the most recent information; hence, the *age of information* in terms of the sampling period. The estimation law is in alignment with (4.70). The simplified notation of AoI without the superscript is accounted for the single-hop case.

The controller is an LQR controller, designed independent of the network. It aims to minimize the infinite horizon quadratic cost function:

$$\mathcal{F}_i \triangleq \limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}(\boldsymbol{x}_i[t])^T\boldsymbol{Q}_i\boldsymbol{x}_i[t] + (\boldsymbol{u}_i)^T\boldsymbol{R}_i\boldsymbol{u}_i[t]\right]. \tag{5.6}$$

$\mathcal{F}_i$ is called the linear-quadratic-Gaussian (LQG) cost function. The matrices $\boldsymbol{Q}_i$ and $\boldsymbol{R}_i$ are symmetric positive semi-definite weighting matrices of appropriate dimensions penalizing the state error and the control effort, respectively[6]. A lower $\mathcal{F}_i$ indicates a higher control performance, i..e, an increased QoC.

The controller obtains the control input according to a linear time-invariant control law:

$$\boldsymbol{u}_i[t] = -\boldsymbol{L}_i\hat{\boldsymbol{x}}_i[t]. \tag{5.7}$$

The optimal matrix is obtained by solving the DARE, as already explained (2.17) in chapter 2. It is worth mentioning that although the LQG does not contain any network-specific considerations, as it has been shown in [Mai+22], the optimal $\boldsymbol{L}_i$ matrix solving the DARE leads to optimal control law if the network is prone to delays and dropouts. The network-induced imperfections are reflected in the estimation process.

In simple words, the operation of the controller can be summarized as follows: After each estimation step performed according to (5.5), the controller uses the estimated state $\hat{\boldsymbol{x}}_i[t]$ to calculate the control input according to (5.7). Then $\boldsymbol{u}_i[t]$ is applied to the plant $\mathcal{P}_i$ during the next sampling period t. If a new data is received during the t-th sampling period, it is fed into the the estimation algorithm to determine $\hat{\boldsymbol{x}}_i[t+1]$ and the same procedure is repeated to obtain the next control input.

[6]We assume that the set-point to be zero. As a result, the state $\boldsymbol{x}_i[t]$ is also the state error, i.e. the deviation of the state vector from the desired point.

5.2.1.1 Information Staleness and Effects on Control Performance

As explained in the previous section, each controller runs an estimation process based on its latest knowledge about the system state. However, especially in a real networking scenario, it is common to experience delays caused by data processing and transmission. Besides, a subset of the generated data may either be discarded before any transmission or "lost" due to bad link quality or simultaneous access. All these effects combined lead to information staleness and, consequently, inaccuracies in the estimated state. In such a setting, the controllers take sub-optimal actions, which then increase the state error and control effort. As a result, the deterioration of the QoC is inevitable that is captured by an increase in the LQG cost.

As in (5.5), let $\boldsymbol{x}_i[\nu_i(t)]$ denote the most recent information available at the controller[7], indicating $\Delta_i(t) = t - \nu_i(t)$. Next, consider two real-time processes that are sampled with a frequency of 100 Hz, thus, every 10 milliseconds, e.g., the temperature of an office room and the location of a highly mobile Unmanned Aerial Vehicle (UAV). Generally speaking, the room temperature is much less volatile than the location information. Suppose that we are monitoring the states of these two systems via a communication network and we can afford to transmit the latest state every thousandth packets, i.e., once per ten seconds. In that case, we can intuitively see that the age cannot capture the growing uncertainty at the monitor over time between two consecutive status updates and the value of transmitting the next packet when the AoI reaches 1000 differs for these two applications, since they are unlike in state dynamics.

One way of capturing the uncertainty at the monitor is to use the estimation error. The *estimation error* is defined as the difference between the actual and estimated system states, i.e.:

$$\boldsymbol{e}_i[t] \triangleq \boldsymbol{x}_i[t] - \hat{\boldsymbol{x}}_i[t] = \sum_{d=1}^{\Delta_i[t]} \boldsymbol{A}_i^{d-1} \boldsymbol{w}_i[t-d]. \tag{5.8}$$

The closed form equation for $\boldsymbol{e}_i[t]$ can be obtained by subtracting (5.5) from (5.4). The mean squared error (MSE), which is a widely used metric in the literature to quantify estimation performance, can be calculated by taking the expectation of the quadratic form as:

$$MSE_i[t] \triangleq \mathbb{E}\left[(\boldsymbol{e}_i[t])^T \boldsymbol{e}_i[t]\right] \tag{5.9}$$

$$= \sum_{d=1}^{\Delta_i(t)} \mathsf{tr}\left((\boldsymbol{A}_i^T)^{d-1} \boldsymbol{A}_i^{d-1} \boldsymbol{\Sigma}_i\right), \tag{5.10}$$

with the trace operator $\mathsf{tr}(.)$. (5.10) can be interpreted as a mapping function from age AoI to MSE.

[7] $\nu_i(t)$ is always smaller than t because in our implementation, the calculation of $u_i[t]$ happens directly subsequent to sampling. As it is infeasible to have "almost zero" delay in a practical setup, in our mathematical model, we do not allow the equality case, thus, $\nu_i(t) < t,\ \forall t$ follows.

$\boldsymbol{e}_i[t] \in \mathbb{R}^{n_i}$ is a multi-variate Random Variable (RV) defined as the deviation of the system state from its expectation. The first property of $\boldsymbol{e}_i[t]$ is that it is a zero-mean multivariate RV, i.e., $\mathbb{E}\left[\boldsymbol{e}_i[t]\right] = \mathbf{0}$, with $\mathbf{0}$ being a column vector of length n_i that contains only zeros. This can easily be shown by taking the expectation of the RHS of (5.8) and applying the linearity property of expectation. Moreover, $\boldsymbol{e}_i[t]$ is a normally distributed multi-variate RV since each addend in (5.8) is a linear transformation of the multivariate normal RV $\boldsymbol{w}_i[t-d] \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i)$ with $1 \leq d \leq \Delta_i(t)$. In fact, each addend follows a normal distribution with the covariance matrix $\boldsymbol{\Sigma}_d = \boldsymbol{A}_i^{d-1}\boldsymbol{\Sigma}_i(\boldsymbol{A}_i^{d-1})^T$.

Proof. Given any $d \geq 1$, $\boldsymbol{y}_d[t] = \boldsymbol{A}_i^{d-1}\boldsymbol{w}_i[t-d]$ is the d-th addend of (5.8) with $\boldsymbol{y}_d[t] \in \mathbb{R}^{n_i}$ and $\mathbb{E}\left[\boldsymbol{y}_d[t]\right] = \mathbf{0}$. We obtain the covariance matrix $\boldsymbol{\Sigma}_d$ as follows:

$$\begin{aligned}
\boldsymbol{\Sigma}_d &\triangleq \mathbb{E}\left[(\boldsymbol{y}_d - \mathbb{E}\left[\boldsymbol{y}_d\right])(\boldsymbol{y}_d - \mathbb{E}\left[\boldsymbol{y}_d\right])^T\right] \\
&= \mathbb{E}\left[\boldsymbol{A}_i^{d-1}\boldsymbol{w}_i[t-d](\boldsymbol{w}_i[t-d])^T(\boldsymbol{A}_i^{d-1})^T\right] \\
&= \boldsymbol{A}_i^{d-1}\mathbb{E}\left[\boldsymbol{w}_i[t-d](\boldsymbol{w}_i[t-d])^T\right](\boldsymbol{A}_i^{d-1})^T \\
&= \boldsymbol{A}_i^{d-1}\boldsymbol{\Sigma}_i(\boldsymbol{A}_i^{d-1})^T.
\end{aligned}$$

□

As a result, the total estimation error $\boldsymbol{e}_i[t]$ comprised of d independent addends, which are $\{\boldsymbol{y}_d[t] : 1 \leq d \leq \Delta_i(t)\}$, is characterized by the multivariate normal distribution $\boldsymbol{e}_i[t] \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_e)$. Since we are able to sum up the covariance matrices as the individual addends are independent RVs, it holds that:

$$\boldsymbol{\Sigma}_e = \sum_{d=1}^{\Delta_i(t)} \boldsymbol{A}_i^{d-1}\boldsymbol{\Sigma}_i(\boldsymbol{A}_i^{d-1})^T. \tag{5.11}$$

Here, it is important to emphasize that an increase in Δ_i creates a new positive semi-definite addend on the RHS. Note that if $\boldsymbol{A}_i$ is a scalar, this corresponds to an increase in the variance of $\boldsymbol{e}_i$'s distribution. Let us illustrate this with a numerical example that involves two scalar loops with $\boldsymbol{A}_i \in \{1.0, 1.2\}$ and $\boldsymbol{\Sigma}_i = 1.0$.

Fig. 5.10 depicts the probability density function Probability Density Function (PDF) of estimation error for different control systems when Δ_i ranges from one to eight. The figure shows how the PDFs become more stretched as the AoI at the estimator increases. In other words, the uncertainty of state estimation is amplified when Δ_i gets higher. However, the uncertainty grows at different rates for each sub-system as information becomes more stale. In fact, the sub-system with $\boldsymbol{A}_i = 1.2$ depicted at the bottom has a much wider distribution of the squared error at $\Delta_i = 8$ than the one with $\boldsymbol{A}_i = 1.0$ shown at the top. Fig. 5.10 can

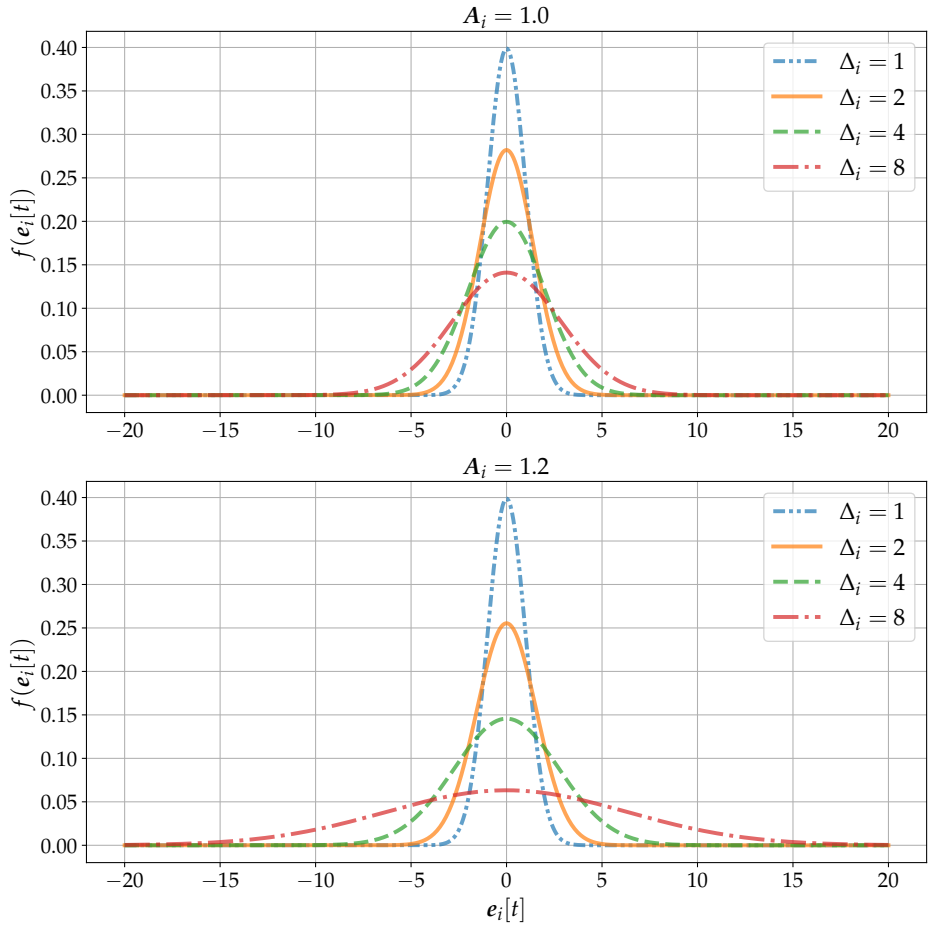


Figure 5.10: The probability density function of the estimation error $f(\boldsymbol{e}_i[t])$ for varying AoI values. The distribution is characterized by $\boldsymbol{e}_i[t] \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_e)$ with $\boldsymbol{\Sigma}_e = \sum_{d=1}^{\Delta_i[t]} \boldsymbol{A}_i^{d-1} \boldsymbol{\Sigma}_i (\boldsymbol{A}_i^{d-1})^T$. Here, $\boldsymbol{A}_i = \{1.0, 1.2\}$ and $\boldsymbol{\Sigma}_i = 1.0$ are used.

be interpreted as an illustration of how the importance of updating the controller relates to the freshness property of information and to its context, i.e., who is sending and receiving the information, what is the communication purpose, etc. In our toy example illustrated in the figure, the context is defined by the goal of uncertainty reduction at two destinations that are monitoring two remote processes with distinct system dynamics.

Although the estimation error is not a direct measure of control performance, it strongly affects the control inputs' accuracy. With the growing uncertainty at $\mathcal{C}_i$, the utilized control inputs become sub-optimal due to the mismatch between $\boldsymbol{x}_i[t]$ and $\hat{\boldsymbol{x}}_i[t]$. As a result, $\boldsymbol{u}_i[t]$ is not able to drive the state towards the reference value correctly. This leads to a higher LQG cost, i.e., $\mathcal{F}_i$, since the state error, as well as the spent control effort grows. This phenomenon is shown in Fig. 5.11, in which the relationship between the inaccurate estimate and the sub-optimal control is depicted.

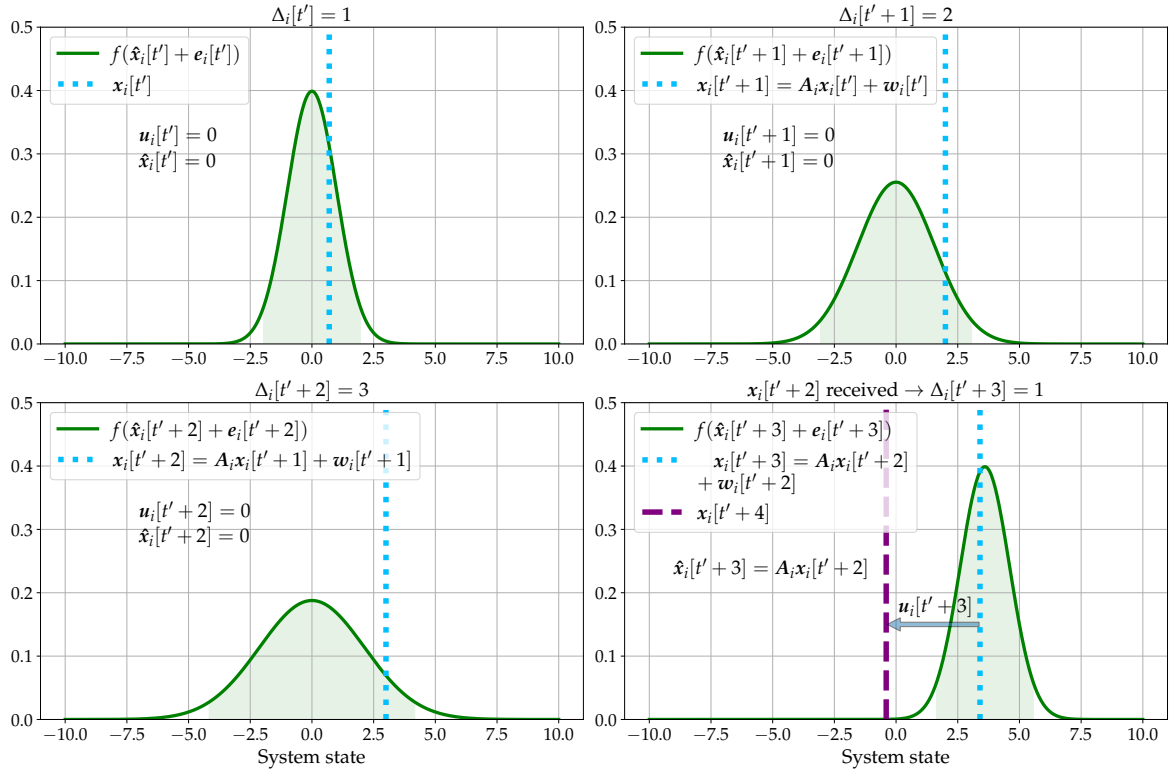


Figure 5.11: An example snapshot of the system state $\boldsymbol{x}_i[t]$, control input $\boldsymbol{u}_i[t]$ and estimated state $\hat{\boldsymbol{x}}_i[t]$. The figures illustrate how the state drifts away from the reference value due to missing status updates about recent changes. Please notice that the distribution of the estimation error is more stretched as Δ_i increases.

In Fig. 5.11, we are able to see the interplay between age, state, estimation, and the control input. Here, the controller has stale knowledge and expects $\boldsymbol{x}_i[t']$ to be correctly driven to the equilibrium point of $\boldsymbol{x}_i = 0$. As the AoI increases further, the controller does not take any immediate action, as it lacks recent data, leading to zero control input, i.e., $\boldsymbol{u}_i[t] = 0$ for $t \in [t', t'+2]$. However, after receiving a successful update, the controller improves its estimation and generates a non-zero control input at $t = t'+3$ to drive the state back to zero. Both the state deviation and the following control effort contribute to the LQG cost and lead to a degradation in the control performance.

5.2.1.2 Task-Oriented Communications and Problem Statement

The optimal state feedback gain matrix $\boldsymbol{L}_i$ from (5.7) assumes perfect communication between the components of a feedback control loop. However, this contradicts our setup, in which the state observations are to be sent over a physical wireless link. Therefore, to limit the deviation of controller design from optimality, the network should aim to reduce the network-induced effects, thus bringing it closer to the ideal case.

If the network consists of multiple time-critical applications and the available bandwidth cannot accommodate the entire traffic, it becomes essential to identify the most urgent transmissions to use the limited resources in an efficient manner. One way of doing so, is to select the highest expected uncertainty reduction; or, equivalently scheduling the user with the highest MSE[8]. In contrast to capturing estimation quality, the control theory uses the LQG cost to measure the level of success in accomplishing the control objective. Although it is challenging to formulate the exact relationship between the MSE and LQG cost analytically, these are strongly intertwined.

Having said that, we seek to exploit the indirect relationship between estimation and control performances. In other words, by targeting an MSE reduction, we expect to decrease the control cost, as we have seen in the previous sections 3.2 and 4.4. Thus, we propose implementing a customized wireless MAC protocol $\boldsymbol{\pi}$ using SDRs to minimize the average LQG cost, i.e.:

$$\boldsymbol{\pi} = \arg\min_{\boldsymbol{\pi}} \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{\pi})\right], \tag{5.12}$$

with:

$$\mathcal{F}_i(\boldsymbol{\pi}) \triangleq \frac{1}{T}\sum_{t=0}^{T-1}(\boldsymbol{x}_i[t])^T \boldsymbol{Q}_i \boldsymbol{x}_i[t] + (\boldsymbol{u}_i[t])^T \boldsymbol{R}_i \boldsymbol{u}_i[t]. \tag{5.13}$$

As in (5.6), $\mathcal{F}_i(\boldsymbol{\pi})$ is the linear quadratic cost when $\boldsymbol{\pi}$ is employed. Section 5.2.2.2 presents two examples of such protocols utilizing MSE as a scheduling metric.

5.2.2 MAC Protocols for Real-Time NCSs

This section presents various MAC protocols that have been implemented and tested in our experimental study. We start by explaining three contention-based protocols in 5.2.2.1. Then, we briefly present three contention-free techniques: 1) slotted Round-Robin RR scheduler, 2) *WiFresh* from [KRM21], and 3) Maximum Error First from 4.1.2. As we are going to show through practical measurements later, the enhanced version of the MEF scheduler stands out in terms of QoC.

5.2.2.1 Contention-based Protocols

ALOHA: The ALOHA protocol, originally proposed in [Abr70], is based on the simple idea of sending any incoming data upon arrival, without any sophisticated MAC layer considerations. As we know from the fundamentals of wireless communications that such a strategy leads to

[8]Note that the uncertainty reduction happens only if the transmission is successful. This requires the consideration of channel conditions. In Sec. 5.2.2.2 we discuss in detail how the link reliability is incorporated into scheduling decisions in our setup.

high packet loss if the network traffic load is high and if the same resources are shared by multiple users.

Slotted ALOHA: Slotted ALOHA (SA) have been proposed in [Rob75b] as an improvement over ALOHA. It divides time into equally long slots, hence, is a *time-slotted* protocol. When a user has data to transmit, it waits for the next slot to begin and then either transmits with a time-invariant Channel Access Probability (CAP) p_i or backs off with the remaining probability $1-p_i$. Similar to other time-slotted protocols, the SA assumes a universal time structure, thus, time synchronization among users. In our framework, this is realized through beacon packets that are transmitted periodically. Further details for synchronization are given in section 5.2.3.

SA has recently been studied in the context of information freshness in [CGL20; Han+20][9]. In particular, in [Han+20], the authors conduct an experimental study using SDRs. As derived in [CGL20] analytically, when the SA protocol is employed, each sub-system achieves a mean AoI $\bar{\Delta}_{SA}$ as follows:

$$\bar{\Delta}_{SA} = \frac{1}{p(1-p)^{N-1}}. \tag{5.14}$$

$N \geq 3$ is the number of users in the network. As proven in the same work, the age-optimal CAP p^* for SA is $p_i^* = p^* = \frac{1}{N}, \forall i$. In the remainder of this section, we always assume that the SA uses the optimal channel access probability. The analysis in [CGL20], as well as the equation (5.14) are based on the assumption that the nodes continuously transmit the most recent state information. We ensure that this requirement is fulfilled and thus, adopt the LCFS-PD strategy instead of a FCFS queue. Moreover, the slot frequency and the frequency of the aging process have to coincide, which is also what we assume in our study.

Age-dependent Random Access Protocol: Chen et al. proposed the Age-Dependent Random Access (ADRA) protocol in [CGL20] as an optimal age-dependent stationary randomized policy for large-scale wireless networks. ADRA is a threshold-based policy defining a fixed CAP $p = p_i, \forall i$ for each user only if its instantaneous age is not below a certain threshold value $\delta_i = \delta, \forall i$, i.e.:

$$p_i(t) = \begin{cases} 0 & \text{, if } \Delta_i(t) < \delta \\ p & \text{, if } \Delta_i(t) \geq \delta \end{cases} \tag{5.15}$$

(5.15) implies that each source needs to know the instantaneous AoI at the receiver to decide whether the next update packet is eligible for transmission. However, since the sensors may not have the perfect knowledge about the observation history at the monitor, the instantaneous age has to be estimated remotely. To that end, despite being not directly addressed in [CGL20], our framework utilizes Acknowledgment (ACK) packets transmitted by the GW SDR upon a successful reception. Doing so, each sensor SDR estimates the

[9] In [CGL20; Han+20] authors refer to slotted ALOHA as "Age-Independent Random Access (AIRA)".

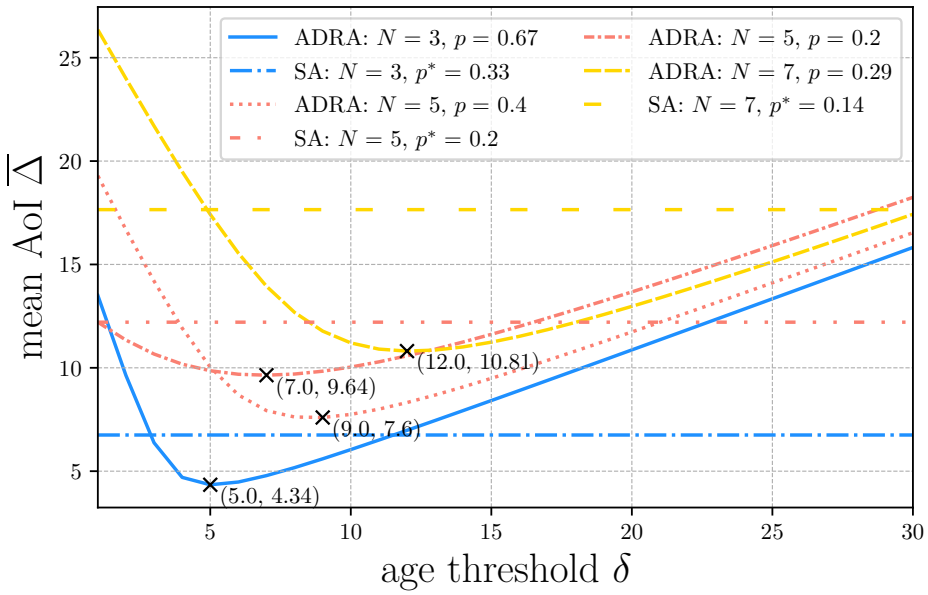


Figure 5.12: Network-wide mean AoI $\bar{\Delta}$ is plotted against age-threshold δ for selected number of users, $N = \{3, 5, 7\}$. p denotes the channel access probability for the age-dependent random access (ADRA) protocol. The horizontal lines show the minimum achievable AoI for slotted ALOHA (SA) with the age-optimal CAP $p^* = \frac{1}{N}$.

instantaneous AoI at the controller assuming that every unacknowledged packet is lost. In case of an unreliable control channel causing either frequent loss of ACK packets or their significantly delayed reception, the ADRA protocol is expected to overestimate the monitor's AoI, thus, leading to more frequent and unnecessary activation of users.

[CGL20] derives the mean AoI for the ADRA protocol as:

$$\bar{\Delta}_{ADRA} = \frac{\delta}{2} + \frac{1}{pq} - \frac{\delta}{2(\delta pq + 1 - pq)}, \tag{5.16}$$

with the successful status update probability q. To obtain the value for q we refer to the original paper. Moreover, the optimal values for δ and p, i.e., δ^* and p^*, can be found using the bisection method, as suggested by the authors[10]. Fig. 5.12 depicts the network-wide mean AoI for varying user count $N = \{3, 5, 7\}$ and δ up to 30. One can clearly conclude that if the right parameters are selected, the ADRA protocol outperforms the SA in terms of information freshness.

5.2.2.2 Contention-free Protocols

Round-Robin (RR): The RR is a widely used scheduling policy prioritizing each user in a given order. Therefore, it is neither a channel-aware nor an application-aware strategy. In our

[10]In our results section, when comparing the ADRA protocol to others, we have used the optimal values for δ and p.

implementation, the users are prioritized in the same order as their unique control loop ID i. Given that at any time slot $t \in \mathbb{N}$ only a single source node $i \in \{1, 2, \ldots, N\}$ is scheduled, the next node to schedule can simply be found according to the following rule:

$$i^*(t) = \arg\min_i \{t + N - i \mod N\}, \tag{5.17}$$

with the modulo operator $\mod$. There is always a single user i that makes $t+N-i \mod N$ zero, e.g., $i^*(t) = 1$, $i^*(t) = 2$, etc. In our framework, we enforce synchronization among users with the help of beacon packets as in SA and ADRA protocols. Therefore, each source node i can track the current time slot index t, through which the (5.17) can be solved.

For a given N, the RR scheduler leads to periodical transmission from each user's perspective. Therefore, if there is no packet loss in the network, the discrete-time AoI curve converges to a periodical pattern, i.e., rising up to N and then dropping to one. Put differently, whenever the AoI reaches $\Delta_i(t') = N$, it is followed by a reset to $\Delta_i(t'+1) = 1$ in the subsequent slot. As a result, the long-term mean AoI of each source node is equivalent to its mean AoI throughout N slots, which can be derived as a sum of arithmetic sequence as:

$$\bar{\Delta}_{RR} = \frac{1}{N}\left(\frac{N}{2}(1+N)\right) = \frac{N+1}{2}. \tag{5.18}$$

Despite its simple operation, the RR scheduler comes with some challenges in practical deployment. In addition to requiring time synchronization, it may cause underutilization of the network resources. In particular, the RR reserves a certain amount of resource units, e.g., a time slot, exclusively to a user. This also implies that the remaining portion of the resource is wasted if the transmission of packet takes shorter than the allocated time span. Especially in connected robotics and remote monitoring scenarios, in which the data packets are small-sized, finding a suitable slot duration to accommodate exactly a single transmission is a hidden challenge. Our experimental results reveal the performance loss caused by this issue. However, we do not tackle the slot duration adaptation problem in this thesis and fix the slot duration to ten milliseconds.

WiFresh: One of the most prominent examples of practical AoI research is *WiFresh* [KRM21], which is a polling-based protocol, i.e., it does not rely on time synchronization. In particular, the channel access is initiated by a *poll request* transmitted by the GW node. As a response, the user, from which the information is requested, replies with the most recent measurement that is available in the buffer.

In their work, the authors consider multiple sources transmitting to a GW using SDRs. The GW tracks the AoI of each source process and asks for a status update packet by sending a poll request. Additionally, it estimates the channel reliability $r_i^{ch}(t)$ between each source

and the BS by the following equation:

$$r_i^{ch}(t) = \frac{RX_i^D(t) + 1}{TX_i^P(t) + 1}, \tag{5.19}$$

where $RX_i^D(t)$ and $TX_i^P(t)$ denote the number of successfully received data packets and transmitted poll packets in the last 0.5 seconds, respectively. The next user to poll is then defined by the max-weight policy, i.e:

$$i^*(t) = \arg\max_i \{r_i^{ch}(t)\tilde{\Delta}_i(t)\}, \tag{5.20}$$

with $\tilde{\Delta}_i(t)$ being the estimated age of the freshest information about source node i. The need for the age estimation arises, due to source and destination nodes being separated, and the exact timing of sampling events at the source process are unknown to the GW. Therefore, the GW has to estimate the AoI by tracking the sampling periods that have elapsed since the latest reception. Our approach is similar to the one considered in [KRM21]. We would like to mention that this is another example of hidden challenges in systems research that is revealed only prior to practical deployment[11].

Accounted for its channel awareness, the WiFresh protocol is particularly a good fit for high mobility scenarios, in which the nodes experience dynamic link reliability. Besides, as it does not rely on time synchronization, one can argue for its lower complexity than implementing the SA or ADRA in practice. However, in contrast to SA or ADRA, that are contention-based, the packet success ratio is expected to be much higher as simultaneous channel access is avoided by virtue of the centralized polling mechanism.

Maximum Error First Scheduler (MEF): As introduced in section 4.1, the MEF scheduler is a greedy scheduler that allocates resources by starting with the user with the highest instantaneous estimation error. In particular, it suggests employing the MSE from (5.10) as the scheduling metric in a time-slotted radio access network. Hence, the next user to schedule is determined by:

$$i^*(t) = \arg\max_i \{MSE_i[t]\}. \tag{5.21}$$

Note that $i^*(t)$ determines the only user that is allowed to access the shared wireless channel in time slot t. It goes without saying the the MEF scheduler relies on time synchronization.

When it comes to implementing such an age-dependent control-aware strategy on real HW, there are a few design choices to make. One of these is the frequency of control messages carrying the information about the transmission schedule. Similar to section 5.1, we assume that the scheduling grant is communicated to sensor nodes via beacon packets before a frame

[11]For instance, the theoretical part of this thesis assumes perfect time synchronization, as well as, the exact knowledge about sampling instants

starts. In the following analysis, we set the length of a frame as $L = 20$ time slots, whereas a time slot is ten milliseconds long. It is important to mention that since the scheduling grant is sent once in every 200 ms, the scheduler does not get the chance to modify any decision that has been made at the beginning of a frame. In our implementation, the GW neglects the packet loss probability and allocates each of those 20 slots in advance as if all transmissions were to be successful. One of the disadvantages of this approach is the incapability of the scheduler to dynamically modify a scheduling decision during the frame, e.g., after detecting a loss event.

The scheduling metric employed in (5.21) is strongly system-dependent; hence, its unit varies from one control application to another. In a practical scenario comprised of multiple heterogeneous applications, one cannot employ the RHS of (5.21) in its raw form. More precisely, the scheduling decision based on the raw MSE may be comparing multiple numbers in different units and orders of magnitude. As a solution to this hidden issue, which may only reveal itself when it comes to deploying such protocols, we propose and employ the *normalized mean squared error (NMSE)* given by:

$$\|MSE_i(t)\| \triangleq \frac{MSE_i[t]}{MSE_{\Delta_i=1}} \tag{5.22}$$

where $MSE_{\Delta_i=1}$ is the MSE when the AoI is equal to one. Put differently, the NMSE captures the factor, by which the MSE is amplified as the age grows beyond one. As a result, NMSE is a dimensionless quantity. Please note that the normalization factor, i.e., $MSE_{\Delta_i=1}$, is equal to the trace of the covariance matrix, which is the only addend in the RHS of (5.10) when $\Delta_i(t)$ equals one. Similar to MSE, when the AoI is zero, the NMSE is zero as well. Consequently, the MEF algorithm implemented in our framework is characterized by the following scheduling policy:

$$i^*(t) = \arg\max_i \{\|MSE_i[t]\|\}. \tag{5.23}$$

Polling-based Maximum Error First Scheduler (pMEF): In contrast to MEF, WiFresh does not operate in a slotted fashion. Thus, if the response to a poll packet comes earlier than the beginning of the next slot, the WiFresh is expected to reduce the amount of idle time between two consecutive transmissions[12]. Moreover, by definition, the MEF scheduler is a channel-unaware protocol. Therefore, we propose to combine the strengths of both strategies and propose a polling-based channel- and control-aware scheduling policy that makes its decision according to the following rule:

$$i^*(t) = \arg\max_i \{r_i^{ch}(t)\|MSE_i(t)\|\}, \tag{5.24}$$

[12]In section 5.2.4, we discuss the effect of this property of polling on the AoI and control performances in detail.

Figure 5.13: A photo of our testbed while taking measurements with twelve control sub-systems.

with $r_i^{ch}(t)$ as in (5.19). The NMSE is obtained by plugging the instantaneous estimated AoI $\tilde{\Delta}_i(t)$ into (5.10).

5.2.3 Design and Implementation

5.2.3.1 Hardware and Software

Our experimental setup comprises $N \in \{2, 3, \ldots, 15\}$ control applications programmed in Python programming language. Due to periodic sampling, each plant process outputs a packet in constant intervals that are forwarded to the corresponding SDR $\mathcal{S}_i$ using a UDP socket[13]. After being received by the SDR, each packet traverses through multiple packet processing blocks programmed in C++ with GNU Radio [Rad]. In contrast to [Han+20], we have not directed the data flow of multiple source processes into a single SDR. We present Fig. 5.13 to give an idea about our experimental setup. It consists of eight computers running Ubuntu 20.04.3 LTS operating system. Ettus Research's USRP™ B200mini-i and B205mini-i SDRs are used for wireless data transmission.

Our framework clearly separates of the application layer and the wireless communication stack. In particular, after their generation, the packets are written to a local UDP socket that is read by the GNU Radio environment's signal processing blocks. Therefore, the application process is entirely agnostic to the communication network behind the UDP socket. Similarly, the interfacing between the GNU Radio and the application layer is done on the receiver side using UDP sockets. Our purpose in choosing a clear separation between the wireless communication stack and the application layer is to simplify the integration of any internet protocol-based application into our framework, thereby removing the barrier to its adoption.

[13] Each plant process $\mathcal{P}_i$ and sensor/SDR $\mathcal{S}_i$ run on the same machine.

An automation script is used to minimize the influence of a human operator on the results, especially when the measurement runs are started, stopped, and repeated. In addition, we discard the data belonging to the first and last five seconds of each 30-seconds-long run to avoid transitional effects of the startup and completion phases.

5.2.3.2 Synchronization

Time synchronization is a prerequisite for realizing time-slotted MAC protocols, such as SA, ADRA, MEF, and RR. To that end, we follow a similar approach as in [Han+20] and make use of periodic beacon packets at the beginning of each *frame*. A beacon packet contains three main fields:

- **MAC header**: Contains information such as packet type, MAC sequence number, source and destination addresses.
- **Payload**: Contains information specific to the employed MAC protocol, such as frame length, slot duration, i.e., $T_D = 10$ ms, slot index, and the transmission schedule, if applicable, e.g., for the MEF scheduler.
- **CRC**: Contains the 16-bits long cyclic redundancy check (CRC) field used for error detection, mainly caused by packet collisions in our setup.

After receiving a beacon packet, each $\mathcal{S}_i$ marks the current time as the beginning of the next frame and sets the current slot to the time slot index indicated by the *Payload* field[14]. This is based on the assumption that the difference in processing delays at each $\mathcal{S}_i$ is negligible. GNU Radio's *high_res_timer* library has been used for time stamping purposes with high resolution.

5.2.4 Evaluation

We have selected scalar control loops of three different classes representing heterogeneous task criticalities. The least challenging class of systems are $\mathcal{I}_{easy} = \{1, 4, 7, 10, 13\}$ with the system matrix $\boldsymbol{A}_1 = \boldsymbol{A}_4 = \cdots = \boldsymbol{A}_{13} = 1.0$. The second and third categories, i.e., $\mathcal{I}_{mid} = \{2, 5, 8, 11, 14\}$ and $\mathcal{I}_{hard} = \{3, 6, 9, 12, 15\}$ have the system matrices $\boldsymbol{A}_2 = \cdots = \boldsymbol{A}_{14} = 1.1$ and $\boldsymbol{A}_3 = \cdots = \boldsymbol{A}_{15} = 1.2$, respectively. The proportionality between the system matrix and control difficulty can be deducted from (5.4) intuitively, which defines the relationship between the current state $\boldsymbol{x}_i[t]$ and the next state $\boldsymbol{x}_i[t+1]$. The input and noise covariance matrices are

[14]Beacon packets carry information about slot duration and frame length, although they are constant in our study. Doing so, we aim to increase the flexibility of our implementation and facilitate the study on the effect of varying slot on network and control performance.

selected to be equal among systems, i.e., $\boldsymbol{B}_i = 1.0, \forall i$ and $\boldsymbol{\Sigma}_i = 1.0, \forall i$. Moreover, the LQR controller design has been done using $\boldsymbol{Q}_i = 100.0$ and $\boldsymbol{R}_i = 1.0$ for all sub-systems. This means that the state error is penalized a hundred times more than the spent control effort.

As mentioned in section 5.2.3.1, we neglect the first and last five seconds to avoid transitional effects. Therefore, the evaluation of each metric starts after the 500-th time step and ends with the 2500-th time step. As a result, the network-wide average AoI is calculated as:

$$\overline{\Delta} = \frac{1}{2000}\frac{1}{N}\sum_{t=501}^{2500}\sum_{i=1}^{N}\Delta_i(t). \tag{5.25}$$

We capture estimation and control performances by the mean squared estimation error $\overline{MSE}$ and the LQG cost $\overline{\mathcal{F}}$ per user, i.e.:

$$\overline{MSE} = \frac{1}{2000}\frac{1}{N}\sum_{t=501}^{2500}\sum_{i=1}^{N}(\boldsymbol{e}_i[t])^T\boldsymbol{e}_i[t], \tag{5.26}$$

and:

$$\overline{\mathcal{F}} \triangleq \frac{1}{2000}\frac{1}{N}\sum_{i=1}^{N}\sum_{t=501}^{2500}(\boldsymbol{x}_i[t])^T\boldsymbol{Q}_i\boldsymbol{x}_i[t] + (\boldsymbol{u}_i[t])^T\boldsymbol{R}_i\boldsymbol{u}_i[t], \tag{5.27}$$

respectively. Note that as $\boldsymbol{\Sigma}_i = 1.0, \forall i$, the denominator on the RHS of (5.22) is equal to one. Therefore, the raw MSE and the NMSE are equivalent for sections 5.2.4.1 and 5.2.4.2, i.e., $\overline{MSE} = \overline{\|MSE\|}$. However, the same is not valid for section 5.2.5, in which we introduce a new type of application into the network.

5.2.4.1 Contention-Based Protocols' Performance

In Sec. 5.2.2.1, we have discussed three random access protocols from the literature, namely ALOHA, SA, and ADRA. Fig. 5.14 presents the measured mean AoI and its theoretical expectation, i.e., $\bar{\Delta}_{SA}$ and $\bar{\Delta}_{ADRA}$. We do not include ALOHA in the figure, because the ALOHA protocol performs significantly worse than the other two already for a very low number of users in the network. For instance, the instantaneous age up to 1900 was observed in one of the measurements for $N = 3$. Therefore, we omit ALOHA in the remaining evaluation since it is unsuitable for time-sensitive wireless networks with multiple users.

The figure verifies that the ADRA protocol outperforms the SA protocol, as expected. However, we can see a mismatch between the experimental results and the theoretical results from [Han+20]. In fact, our implementation achieves better results for both protocols than their analytical mean values. We believe that there are two main reasons behind this outcome: 1) two or more simultaneous transmissions do not necessarily lead to a packet loss. The authors of [Han+20] have observed the same phenomenon and raised this issue in their work.

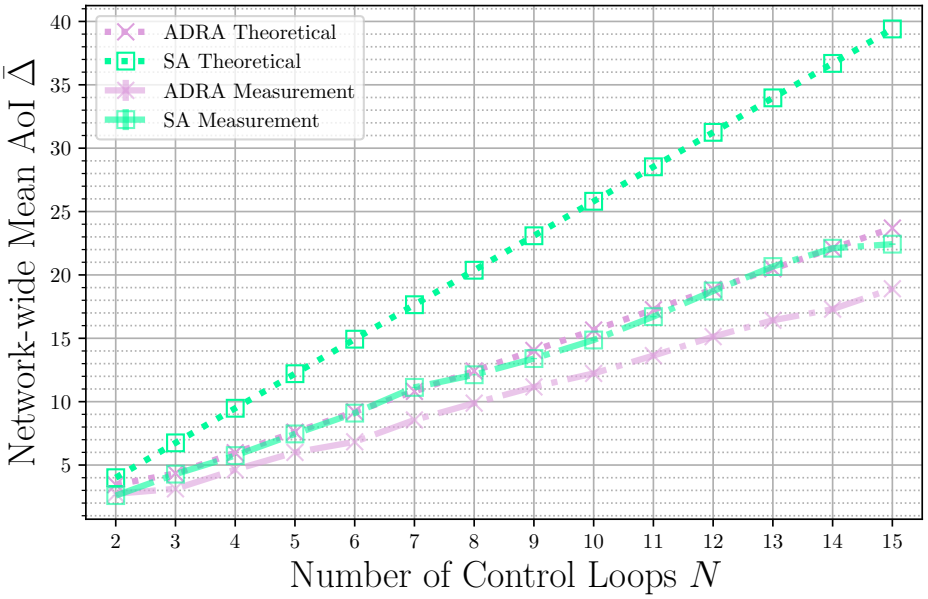


Figure 5.14: Mean AoI of contention-based access protocols, i.e., slotted ALOHA (SA) and age dependent random access (ADRA). Vertical bars illustrate 99% confidence intervals.

2) A data packet transmission does not occupy the entire slot, in contrast to the theory. In our framework, a slot is 10 ms long, whereas our measurements indicate an approximate transmission duration of 3 ms. As we do not force any synchronization in the application layer, this allows some of the packets to miss each other in time, although they are transmitted in the same slot. This phenomenon raises the packet delivery rate per slot above one, hence, improving $\overline{\Delta}$ beyond the theoretical expectation.

Fig. 5.15 presents that the LQG cost capturing the control performance shows divergent behavior for both contention-based protocols. Especially, already for $N = 8$, SA reaches an LQG cost up to 10^{16}, indicating an unstable behavior of the system state. The same applies to the ADRA protocol when there are fifteen users in the network, i.e., $N = 15$, showing the insufficiency of these protocols for multi-user scenarios with time-sensitive control applications.

5.2.4.2 Contention-Free Protocols' Performance

We know that the main strength of the contention-free protocols over random access protocols is their ability to reduce the packet loss rate. This comes at a price of increased complexity and communication overhead, as in the case of polling-based protocols. First, let us analyze the performance of contention-free protocols w.r.t. information freshness.

Fig. 5.16 presents the achieved $\overline{\Delta}$ when N is gradually increased from two to fifteen. We observe that the WiFresh delivers the best results by outperforming its closest competitor by

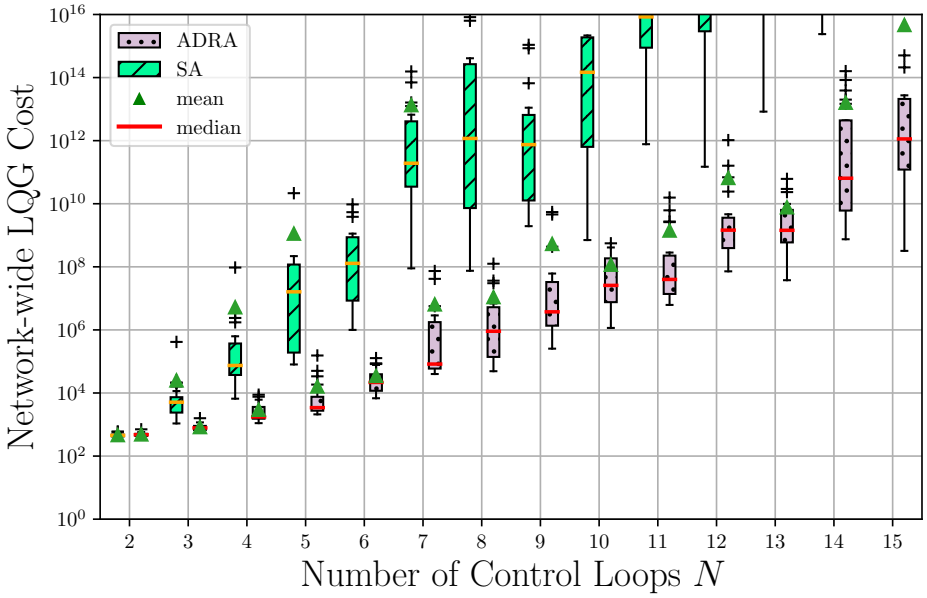


Figure 5.15: The achieved control performance by slotted ALOHA (SA) and age-dependent random access (ADRA) protocol. The control performance is quantified by the LQG Cost, i.e., $\overline{\mathcal{F}}$, defined in (5.27). A lower $\overline{\mathcal{F}}$ stands for a higher performance. The y-axis has been upper-bounded on account of improved presentation.

approximately 10%. The difference between WiFresh and Polling-based Maximum Error First (pMEF) is a result of the unbalanced prioritization of sub-systems by the pMEF algorithm accounted for its control awareness. Put differently, while WiFresh polls sub-systems in a RR fashion under constant channel conditions, the pMEF allocates a bigger portion of the network resources to the class of more challenging sub-systems $\mathcal{I}_{hard}$. This leads to an unbalanced age distribution in the network and increases $\bar{\Delta}$. Nevertheless, as we are going to show later in this section, pMEF is able to achieve better performance for the given control task by virtue of its ability to identify the most valuable transmission.

In our setup, the average polling time, defined as the time between a poll request and the reception of the corresponding data packet, is shorter than a time slot duration. Therefore, the beacon-based protocols, such as RR and MEF initiate fewer transmissions resulting in increased mean AoI than WiFresh and pMEF. This leads to longer idle periods for less critical sub-systems when MEF is used. As a result, the larger gap between control-aware and control-unaware protocols, i.e., MEF and RR, can be observed. Particularly, the MEF scheduler achieves $\bar{\Delta}$ beyond 20 for $N = 15$, while RR achieves less than 10 for the same number of sub-systems. Note that when RR is employed, the mean age performance matches its theoretical expectation derived in (5.18).

Fig. 5.17 shows the fraction of network resources allocated to each class of control applications, i.e., $\mathcal{I}_{easy}$, $\mathcal{I}_{mid}$, and $\mathcal{I}_{hard}$. The figures confirms that all classes are treated

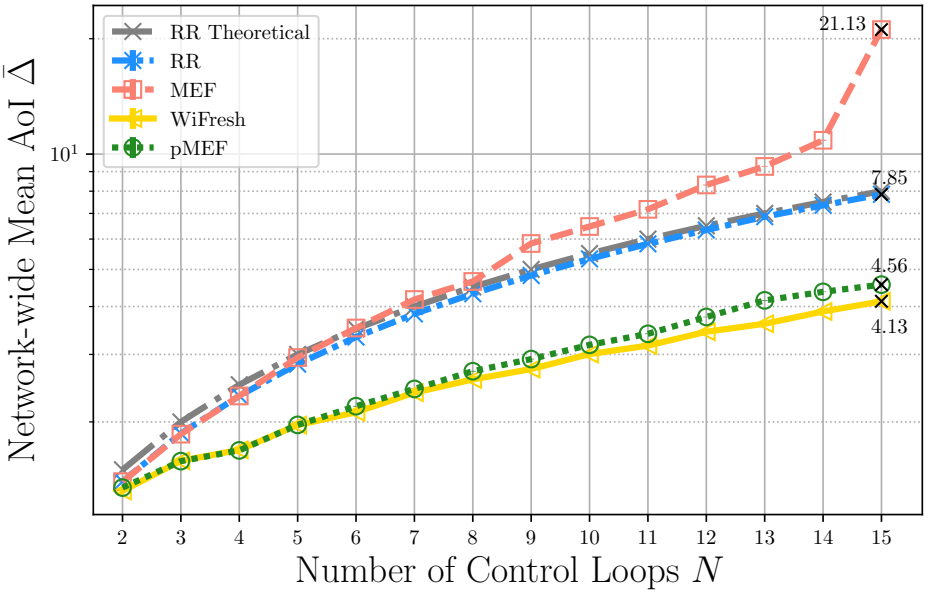


Figure 5.16: The achieved long-term average AoI by round-robin (RR), maximum error first (MEF), WiFresh and polling-based MEF (pMEF). Vertical bars illustrate 99% confidence intervals. The y-axis is drawn on logarithmic scale.

equally when RR and WiFresh are schedulers are employed. On the contrary, more time-critical applications are scheduled more frequently by the MEF and pMEF strategies.

Now, let us have a look at Fig. 5.18, which shows the estimation performance measured by $\overline{MSE}$. The results suggest that the control-aware protocols, i.e., MEF and pMEF, outperform their direct competitors, i.e., RR and WiFresh, respectively. Especially, as the resource scarcity grows due to increasing number of users, e.g., for $N = 15$, the importance of control awareness stands out. That is, MEF achieves relatively lower $\overline{MSE}$ than RR, although a lower $\overline{\Delta}$ has been achieved by RR. One can also say that the information freshness is traded for an increase in the estimation performance. We can observe a similar trend, if we compare pMEF and WiFresh. More specifically, pMEF outperforms WiFresh by up to 18% when there are 15 control sub-systems in the network.

Despite being an insightful task-specific application layer metric, the MSE does not capture the control performance to the fullest extent. Nevertheless, the QoC is strongly intertwined with the estimation accuracy, as discussed in detail in section 5.2.1.1. By virtue of this indirect relationship between the estimation and control performances, we observe a similar trend for the LQG cost in Fig. 5.19. From the figure, we can deduct that the pMEF strategy is able to outperform the WiFresh protocol by up to 21%. The beacon-based protocols' performance follows a similar trend as well, with MEF outperforming RR by 47%. It is very important to point out that there is a significant difference between the considered contention-free and contention-based protocols regarding the LQG cost.

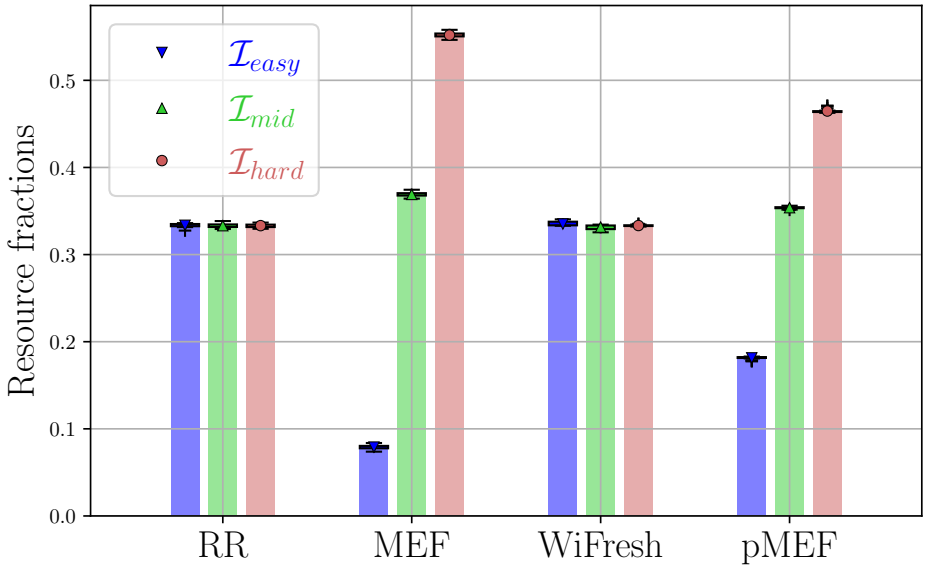


Figure 5.17: The fractions of network resources allocated to each control class. Control-unaware protocols. i.e., RR and WiFresh treat all system classes equally. On the other hand, the control-aware protocols, i.e., MEF and pMEF lead to an unbalanced distribution of resources.

5.2.5 A Real-Life Application Case Study: Inverted Pendulum

In the previous sections, we have shown the performance improvement of our proposed pMEF scheduler over conventional protocols. The selected scalar control systems were theoretical ones to illustrate this effect in a simple scenario. In the following analysis, we introduce a new class of control system, namely the inverted pendulum. The discrete-time state space representation has already peen presented in section 4.4.4.1. The same parameters have been used in this study, thus, we omit the detailed explanation of the inverted pendulum model and refer to section 4.4.4.1.

In order to see NMSE metric in action[15], we repeat our measurements with 15 control sub-systems, where we substitute all sub-systems of class $\mathcal{I}_{mid}$ with inverted pendulums, i.e., $\boldsymbol{A}_2 = \boldsymbol{A}_5 = \cdots = \boldsymbol{A}_{14} = \boldsymbol{A}_{IP}$. Analogously, we set the input and noise covariance matrices to $\boldsymbol{B}_{IP}$ and $\boldsymbol{\Sigma}_{IP}$.

Fig. 5.20 shows the evolution of NMSE with increasing AoI together with the MSE for inverted pendulum without the normalization step from (5.22). It illustrates the different growth rate of the NMSE for our considered applications. Besides, it reveals that the inverted pendulum lies between the $\mathcal{I}_{hard}$ and $\mathcal{I}_{easy}$ classes concerning the NMSE. Note that due to the significant difference in orders of magnitude between IP_{MSE} and other curves, the usage of the raw MSE would lead to resource starvation of inverted pendulums, in return, to their

[15]Here, we have to use NMSE instead of MSE, because the estimation error of a scalar loop and an inverted pendulum are not directly comparable.

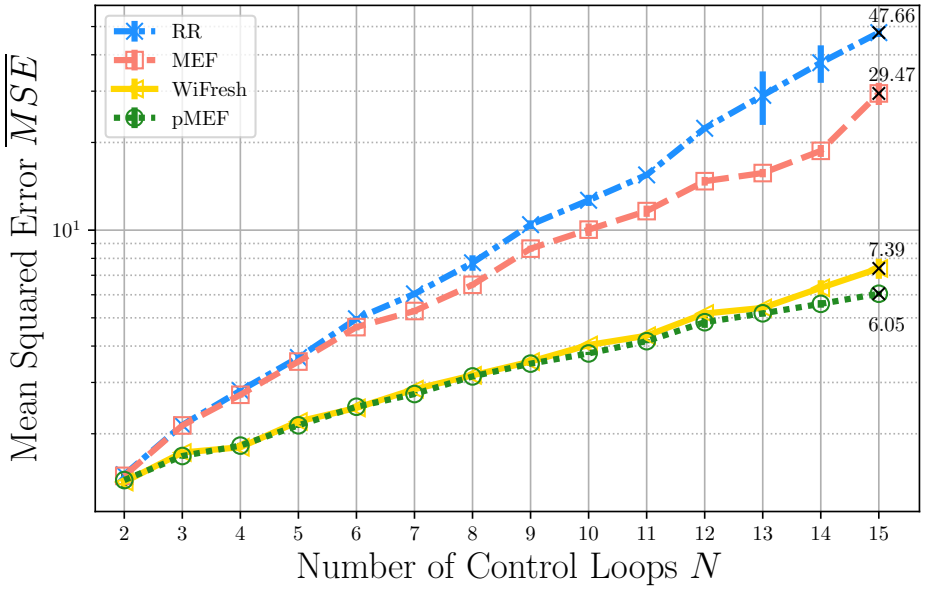


Figure 5.18: Average estimation performance achieved by the selected contention-free protocols, i.e., RR, MEF, WiFresh and pMEF. A lower $\overline{MSE}$ represents a performance raise. Vertical bars illustrate 99% confidence intervals. y-axis is drawn on logarithmic scale.

destabilization. Therefore, the following discussion considers only the usage of NMSE both for MEF and pMEF strategies.

Similar to the previous subsections, we measure the control KPIs to validate the applicability of the pMEF protocol for real-life applications. To that end, we have recorded pendulum angle ϕ and cart position ξ trajectories throughout 20 measurements. To narrow down the focus on IP, the following discussion is limited to the IP relevant metrics such as ϕ and the cart position ξ and does not contain the detailed state trajectories of other sub-systems of class $\mathcal{I}_{easy}$, i.e., $i \in \{1, 4, 7, 10, 13\}$ and of class $\mathcal{I}_{hard}$, i.e., $i \in \{3, 6, 9, 12, 15\}$.

In Fig. 5.21, we present an example trajectory of $\phi_i[t]$ in degrees for $t \in [500, 2500]$ and a randomly selected loop i. It has been recorded during one of the measurements when pMEF scheduler operating with NMSE was in use[16]. From the figure, we are able to observe that the pendulum angle is kept within ± 5 degrees. In addition, Fig. 5.21 shows the maximum and minimum values that are reached by all inverted pendulums in the network when MEF and pMEF is employed. Due to the higher sensor-to-controller delivery rate of pMEF compared to MEF, the pMEF achieves a better control performance w.r.t. ϕ. The same conclusion can be drawn if we look at Fig. 5.22 where the minimum and maximum ξ values are presented. In particular, we are able to observe larger spikes of ξ achieved by MEF than pMEF throughout

[16]The selection of the specific measurement run and loop have been made randomly, and they do not represent an outlier w.r.t. control performance.

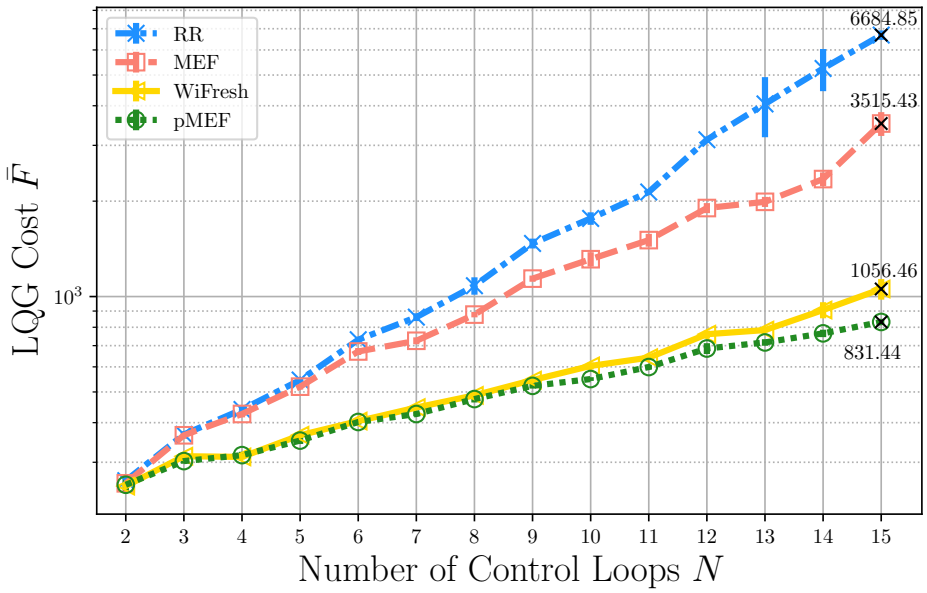


Figure 5.19: The achieved control performance when RR, MEF, WiFresh and pMEF are employed. A lower $\overline{\mathcal{F}}$ corresponds to an increase in control performance. Vertical bars illustrate 99% confidence intervals. The y-axis is drawn on logarithmic scale.

the measurements. To put it another way, the cart needed to move further away from its desired set point, i.e., $\xi = 0$ in order to keep the pendulum upright.

Last but not least, Fig. 5.23 depicts the average nMSE achieved when RR, MEF, WiFresh, and pMEF are used. Each boxplot represents a control class-scheduling strategy combination. In other words, it presents the contribution of each control class to the overall NMSE performance separately. The figure shows that control-unaware strategies, namely the RR and WiFresh strategies, lead to an increased NMSE for the $\mathcal{I}_{hard}$ class systems. This is an expected result of equal treatment of all sub-systems in the network, which lead to higher error values for more critical applications. On the other hand, as we know from Fig. 5.17, MEF and pMEF allocate more resources to $\mathcal{I}_{hard}$ systems than inverted pendulum and $\mathcal{I}_{easy}$. As a result, they are able to balance out the higher task criticality of those sub-systems through their awareness of NMSE displayed in Fig. 5.20.

5.3 Related Work

As listed in the previous chapters, the vast majority of the existing research on AoI and Networked Control Systems (NCS) have limited their validation to analytical results or simulations. In the NCS domain, [Ara+14; Zop+20; Mag+19; Gho+21; Bha+21] are the most prominent exceptions containing experimental measurements. For instance, [Ara+14] evaluates various triggering mechanisms, such as event-triggered control and self-triggered control,

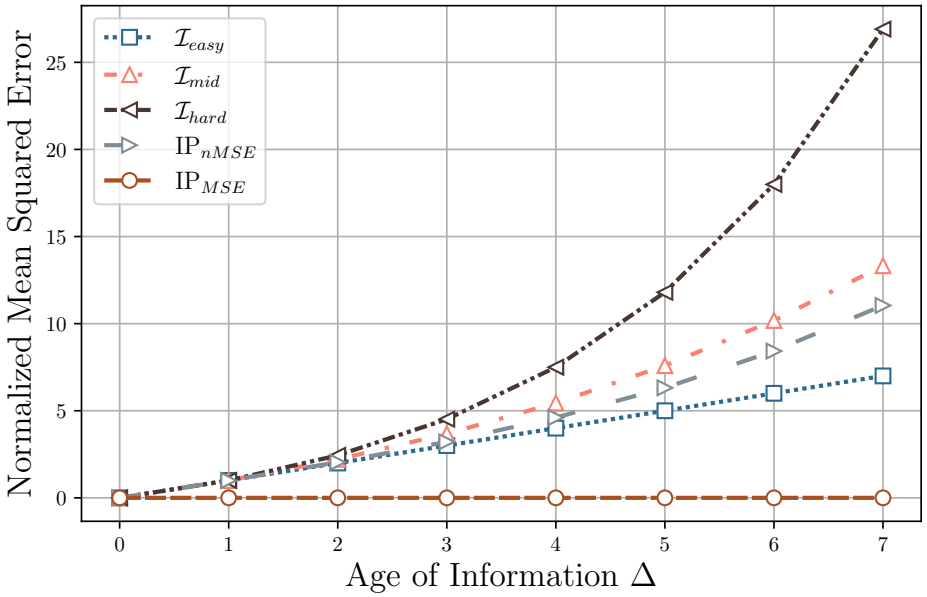


Figure 5.20: The normalized mean squared error plotted against age, Δ, for different control subsystems, i.e., inverted pendulum (IP), $\mathcal{I}_{easy}$, $\mathcal{I}_{mid}$, $\mathcal{I}_{hard}$. In addition, we present the raw MSE for IP prior to the normalization step from (5.22) to illustrate its necessity.

whereas the MAC layer realized using Time-Division Multiple Access (TDMA) with the transmission schedule determined a priori. Moreover, [Gho+21] suggests proactively sending multiple control inputs at once and shows that the control performance can be improved by doing so. In their work, the authors do not include any MAC layer considerations and their evaluation is limited to two control applications. On the other hand, [Bha+21] tackles collisions caused by simultaneous access. They propose a protocol in which the sensors send a transmission request by following a random access procedure. The transmission of status update packets occur in a contention-free way, while the users are scheduled according to the FCFS strategy. Note that none of the existing works on NCS study control-aware design for the medium access and hence can be distinguished from our work.

[Son+18; Bar+19; BBU19] can be named as pioneers of systems research on AoI studying information freshness using real HW. They present AoI performance using real-life connections without any modifications in the communication stack. On the contrary, [SKY19; KM21; KRM21; Han+20; Ogu+22; Pan+22] propose to adopt an age-aware network design while targeting different layers of the communication stack. For instance, [SKY19] proposes an adaptive age-aware transport layer protocol that improves average AoI than those of the widely-used UDP and Transmission Control Protocol (TCP) protocols. Furthermore, [KM21] and [Han+20] focus on random access protocols implemented using software-defined radios (SDRs). For instance, in [Han+20] the authors propose an age-threshold-based random access protocol for wireless networks that reduces the mean AoI compared to the well-known

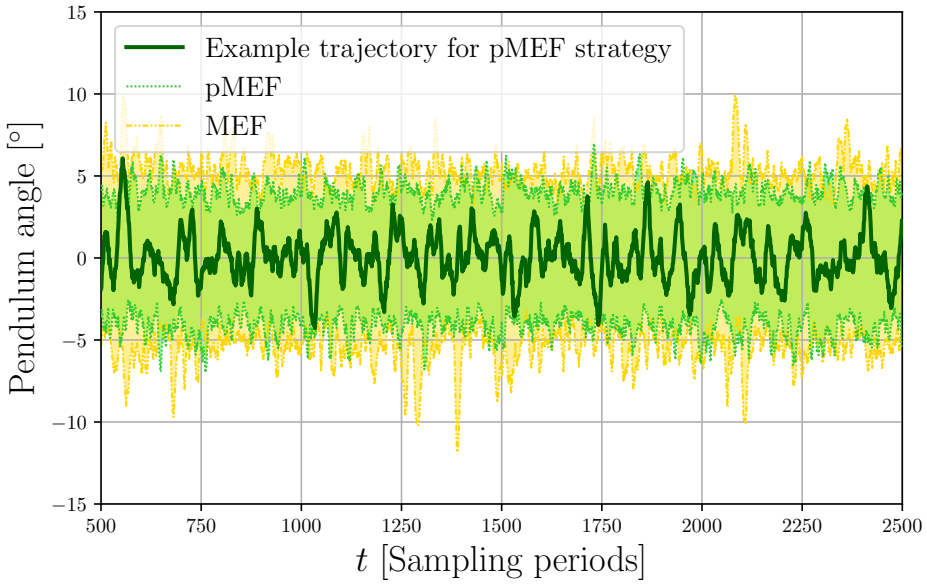


Figure 5.21: An example trajectory of the pendulum angle ϕ when maximum error first (MEF) and polling MEF schedulers are applied. ϕ is plotted in degrees.

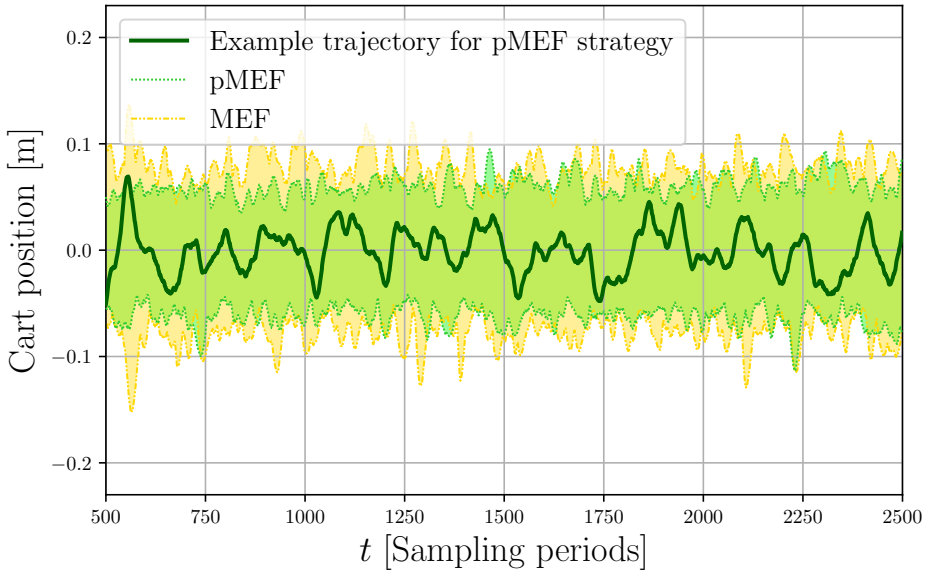


Figure 5.22: An example trajectory of the cart position ξ when maximum error first (MEF) and polling MEF schedulers are applied. ξ is plotted in meters.

slotted ALOHA protocol. The closest works to the content of this chapter are [KRM21] and [Ogu+22]. [KRM21] proposes a contention-free wireless MAC protocol, WiFresh, realized on SDRs and shows that the WiFresh protocol outperforms a standard WiFi network in terms of AoI performance. The authors of [Ogu+22] develop an SDR testbed with single transmitter and multiple receivers and compare the performance of conventional and age-minimizing MAC scheduling policies for push- and pull-based communication scenarios. In contrast

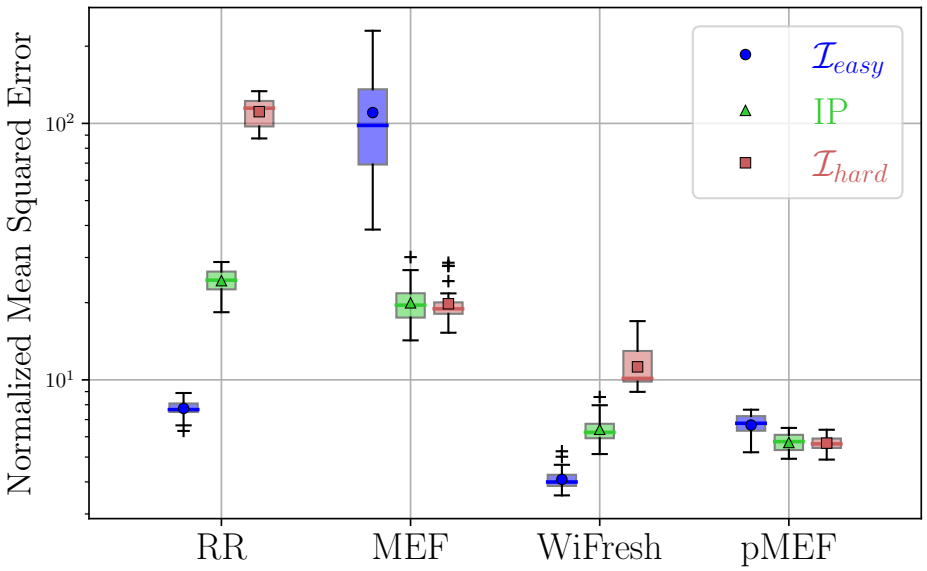


Figure 5.23: The normalized mean squared error (NMSE) achieved when round-robin (RR), maximum error first (MEF), WiFresh and polling MEF (pMEF) schedulers are employed. y-axis is drawn on logarithmic scale.

to the existing literature, we focus on an NCS scenario, in which the primary goal is not providing information freshness but also achieving a higher QoC. Our framework adopts an application-aware design, therefore, it can easily be distinguished from the state-of-the-art.

5.4 Summary

Connected robotics is envisioned as one of the driving applications for next-generation cellular networks [SBC20]. Having access to fresh information is beneficial in real-time monitoring and control scenarios. However, if the network serves control applications of different types and time-criticalities, aiming for freshness may not be optimal for improving performance. On the contrary, it had been proven sub-optimal in chapter 4 compared to customizing the network through the adoption of control-dependent metrics for decision-making. However, when it comes to practical deployment, the adverse and random effects introduced by the wireless channel and hardware components may conceal the performance improvement promised by theoretical results, making the additional effort spent on network design superfluous.

In order to validate our control-aware protocol design in a real network, we have studied realizations of various MAC protocols on hardware, contributing to systems research on wireless networked control systems. Through an extensive study of wireless MAC protocols in a network of multiple heterogeneous feedback control loops, we implement control-aware centralized scheduling using software-defined radios. Our results suggest that consider-

ing control-specific parameters for decision-making outperforms conventional methods, e.g., slotted ALOHA, round-robin scheduler, as well as the age-aware strategies, such as age-dependent random access [CGL20] and WiFresh [KRM21]. The results reveal the high potential in cross-layer design and show that the additional effort pays off with an improved control performance.

Chapter 6

Conclusion and Outlook

To date, offering wider bandwidth and increasing data rates have been the primary approach taken by the standardization and industry, although each new generation of mobile networks is driven by novel applications and use cases. However, due to the rapidly increasing number of connected devices and services, the scarcity of resources constitutes an inevitable bottleneck forcing the networking community to explore new solutions beyond one-size-fits-all approaches. One of the promising alternatives to offering higher data rates is the emerging field of semantic communications, which is built on the idea of including the significance of data relative to their transmission purpose in network and protocol design. Doing so allows the network to identify and prioritize relevant and important data to improve the offered service and filter out those with less contribution to the underlying task, thereby improving the efficiency in resource utilization.

The proposal of the age of information (AoI), which classifies real-time information according to its freshness, has paved the way for semantic communications. In particular, through the introduction of AoI, the network has gained the ability to assign a particular value to packets based on their ability to reduce the staleness at the destination. However, as the value of information (VoI) cannot be separated from the context and communication purpose, solely focusing on AoI may not be the optimal approach, especially when dealing with time-critical applications. Therefore, the age- and value of information belong to the key enablers of semantic communications while playing a vital role throughout this thesis.

6.1 Summary

This thesis has dealt with control-aware resource management for wireless networked control systems. In particular, the AoI allowed us to define the VoI in networked control, which quantifies the amount of uncertainty reduction at the destination. To that end, in the first

half of chapter 3, we have carried out a probability analysis providing us with the stationary distribution of age. In the second half of the same chapter, we have considered a multi-user scenario, with the help of which we have demonstrated an example usage of the age distribution to improve the control performance by solving an optimization problem based on the newly derived VoI metric. In chapter 3 we have solely looked into stationary randomized strategies and have not assumed any online decision-making mechanism.

Contrarily, in chapter 4, we have focused on cellular networks comprising multiple heterogeneous control applications in which a centralized scheduler orchestrates the network resources in an online fashion. Mainly, we have studied various scheduling policies that depend on the instantaneous network state determined by each user's current link reliability and transmission history. The main contributions of chapter 4 can be summarized in three points. First, it suggests employing control system-specific parameters to prioritize users. Thereby, it shows that the performance can be significantly improved through simple heuristics based on the semantics of information beyond freshness compared to conventional methods, including age-dependent scheduling. Second, it demonstrates how the centralized scheduling problem can be formulated as a finite or infinite horizon problem and can be solved optimally by applying stochastic optimization methods, such as value iteration and dynamic programming. Lastly, we have shown that the VoI can be approximated by non-linear age-penalty functions better than the AoI itself. However, the results suggest that one should use accurate models to capture the real value behind the transmitted bits, e.g., by taking system dynamics into account, if achieving a higher control performance is crucial.

Finally, in chapter 5, we have presented a proof-of-concept implementation of control-aware scheduling and validated the most important findings from chapter 4 through experimental results. In addition to our proposed control-aware strategy, our implementation covers some well-known MAC schemes, e.g., round robin, slotted ALOHA, but also relatively newer schemes from the state-of-the-art, such as the age-dependent random access and WiFresh protocols. Despite all imperfections that may appear by virtue of the practical implementation, e.g., hardware and timing errors, random nature of the wireless channel, the results reveal the domination of control-aware scheduling over conventional policies, hence, validating the potential of semantic communications for future generation wireless systems. Furthermore, our experimental study proves the feasibility of realizing such protocols in practice.

6.2 Future Work and Concluding Remarks

***Control-aware transport layer protocols and extended cross-layer design*:** In this thesis, we have shifted the entire intelligence down to the data link layer and neglected the opportunity

of a potential decision-making in other layers. A very good candidate for such an alternative would be a control-aware transport layer protocol directly at the source, inspired by the notion of event-triggering from the control domain [MEJ16]. However, to the best of our knowledge, a combination of control-aware congestion- and medium access control does not exist in the literature.

Throughout this thesis, the centralized scheduler located at the base station has been the only decision-making entity. Therefore, as the scheduler operates remotely to the source, we were only allowed to consider the estimated value of a certain data packet as a function of age, without knowing the exact packet content. However, a very promising idea would be implementing a value-aware transport layer protocol that is able to filter packets at the source based on their content and a signaling mechanism to inform the centralized scheduler whether there is a packet to transmit. Doing so, we believe that a notably higher control performance can be achieved than it has been done in this work.

***Optimal Control-Aware Scheduling with Performance Guarantees*:** Throughout the thesis, our goal has been the minimization of a particular cost function, e.g., MSE, LQG cost while designing the optimal scheduling policies. However, it is a very common approach in the industry and standardization to target a specific service guarantee in the form of minimum throughput, maximum end-to-end latency, and maximum allowed transmission interval [TS22.104]. Such service requirement could also be age-dependent, e.g., average peak age, maximum allowed age, etc. In such a setting, it becomes mandatory for the scheduler to satisfy these bounds while deciding on the next user to schedule. As a result, the consideration of service requirements for control-aware scheduling is one of the possible future directions to look into that is worth investigating.

***Time synchronization for age- and semantics-aware networking*:** The experimental study that we have carried out in chapter 5 has provided us with some lessons learned and hidden challenges, which have not entirely been addressed by the theoretical research on semantic communications and WNCS. An example to such is the necessity for tracking the instantaneous AoI within the MAC layer as it has been firstly discussed in (5.20). More specifically, although the knowledge of the AoI is one of the vital building blocks of semantic- and task-oriented communications, it may not be straightforward to obtain this knowledge at every location in the network. In fact, a significant portion of theoretical research assume the global knowledge of age and base their protocol design on this assumption. This reveals the importance of time synchronization protocols for age-aware networks, not only for the transmission and reception of packets (e.g. for time slots), but also for making decisions for medium access control, data forwarding, and routing.

Bibliography

Publications by the author

Journal publications

[Aya+20b] O. Ayan, H. Murat Gürsu, A. Papa, and W. Kellerer. "Probability Analysis of Age of Information in Multi-Hop Networks." In: *IEEE Networking Letters*. 2020, pp. 76–80. DOI: 10.1109/LNET.2020.2976991.

[Aya+22] O. Ayan, P. Kutsevol, H. Y. Özkan, and W. Kellerer. "Semantics- and Task-Oriented Scheduling for Networked Control Systems in Practice." In: *IEEE Access*. Vol. 10. 2022, pp. 115673–115690. DOI: 10.1109/ACCESS.2022.3218410.

[Kl20] M. Klügel, M. Mamduhi, O. Ayan, M. Vilgelm, K. H. Johansson, et al. "Joint Cross-Layer Optimization in Real-Time Networked Control Systems." In: *IEEE Transactions on Control of Network Systems* 7.4 (2020), pp. 1903–1915. DOI: 10.1109/TCNS.2020.3011847.

[Pap+22] A. Papa, A. Jano, S. Ayvaşık, O. Ayan, H. M. Gürsu, et al. "User-Based Quality of Service Aware Multi-Cell Radio Access Network Slicing." In: *IEEE Transactions on Network and Service Management* 19.1 (2022), pp. 756–768. DOI: 10.1109/TNSM.2021.3122230.

Conference publications

[AEK21] O. Ayan, A. Ephremides, and W. Kellerer. "Age of Information: An Indirect Way To Improve Control System Performance." In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2021, pp. 1–7. DOI: 10.1109/INFOCOMWKSHPS51825.2021.9484454.

[AOK21] O. Ayan, H. Y. Özkan, and W. Kellerer. "An Experimental Framework for Age of Information and Networked Control via Software-Defined Radios." In: *IEEE International Conference on Communications (ICC)*. 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500435.

[AVK20] O. Ayan, M. Vilgelm, and W. Kellerer. "Optimal Scheduling for Discounted Age Penalty Minimization in Multi-Loop Networked Control." In: *IEEE Consumer Communications & Networking Conference (CCNC)*. 2020, pp. 1–7. DOI: 10.1109/CCNC46108.2020.9045142.

[Aya+19] O. Ayan, M. Vilgelm, M. Klügel, S. Hirche, and W. Kellerer. "Age-of-Information vs. Value-of-Information Scheduling for Cellular Networked Control Systems." In: *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. 2019, pp. 109–117. DOI: 10.1145/3302509.3311050.

[Aya+20a] O. Ayan, H. M. Gürsu, S. Hirche, and W. Kellerer. "AoI-based Finite Horizon Scheduling for Heterogeneous Networked Control Systems." In: *IEEE Global Communications Conference (GLOBECOM)*. 2020, pp. 1–7. DOI: 10.1109/GLOBECOM42002.2020.9348053.

[DAK22] Y. Deshpande, O. Ayan, and W. Kellerer. "Improving AoI via Learning-based Distributed MAC in Wireless Networks." In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2022, pp. 1–8. DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798137.

[Mus+19] Z. Music, F. Molinari, S. Gallenmüller, O. Ayan, S. Zoppi, et al. "Design of a Networked Controller for a Two-Wheeled Inverted Pendulum Robot." In: *8th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NECSYS)* 52.20 (2019), pp. 169–174. DOI: https://doi.org/10.1016/j.ifacol.2019.12.153.

[Vil+17] M. Vilgelm, O. Ayan, S. Zoppi, and W. Kellerer. "Control-aware Uplink Resource Allocation for Cyber-Physical Systems in Wireless Networks." In: *23th European Wireless Conference*. 2017, pp. 1–7.

[Zop+20] S. Zoppi, O. Ayan, F. Molinari, Z. Music, S. Gallenmüller, et al. "NCSbench: Reproducible Benchmarking Platform for Networked Control Systems." In: *IEEE Annual Consumer Communications & Networking Conference (CCNC)*. 2020, pp. 1–9. DOI: 10.1109/CCNC46108.2020.9045199.

General publications

[Abr70] N. Abramson. "THE ALOHA SYSTEM: Another Alternative for Computer Communications." In: AFIPS '70 (Fall). Houston, Texas: Association for Computing Machinery, 1970, pp. 281–285. ISBN: 9781450379045. DOI: 10.1145/1478462.1478502. URL: https://doi.org/10.1145/1478462.1478502.

[AED19] M. A. Abd-Elmagid and H. S. Dhillon. "Average Peak Age-of-Information Minimization in UAV-Assisted IoT Networks." In: *IEEE Transactions on Vehicular Technology* 68.2 (2019), pp. 2003–2008. DOI: 10.1109/TVT.2018.2885871.

[Alu+09] R. Alur, A. D'Innocenzo, K. H. Johansson, G. J. Pappas, and G. Weiss. "Modeling and Analysis of Multi-hop Control Networks." In: *IEEE Real-Time and Embedded Technology and Applications Symposium*. 2009, pp. 223–232. DOI: 10.1109/RTAS.2009.40.

[Ara+14] J. Araujo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson. "System Architectures, Protocols and Algorithms for Aperiodic Wireless Control Systems." In: *IEEE Transactions on Industrial Informatics* 10.1 (2014), pp. 175–184. DOI: 10.1109/TII.2013.2262281.

[Aya+23] O. Ayan, S. Hirche, A. Ephremides, and W. Kellerer. "Optimal Finite Horizon Scheduling of Wireless Networked Control Systems." In: *IEEE/ACM Transactions on Networking* (2023), pp. 1–16. DOI: 10.1109/TNET.2023.3300949.

[Bar+19] B. Barakat, H. Yassine, S. Keates, I. Wassell, and K. Arshad. "How to Measure the Average and Peak Age of Information in Real Networks?" In: *European Wireless Conference*. 2019, pp. 1–5.

[BBU19] H. B. Beytur, S. Baghaee, and E. Uysal. "Measuring Age of Information on Real-Life Connections." In: *Signal Processing and Communications Applications Conference (SIU)*. 2019, pp. 1–4. DOI: 10.1109/SIU.2019.8806423.

[Bea+18] L. D. R. Beal, D. C. Hill, R. A. Martin, and J. D. Hedengren. "GEKKO Optimization Suite." In: *Processes* 6.8 (2018). ISSN: 2227-9717. DOI: 10.3390/pr6080106. URL: https://www.mdpi.com/2227-9717/6/8/106.

[Bed+19] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff. "Age-Optimal Sampling and Transmission Scheduling in Multi-Source Systems." In: *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. Mobihoc '19. 2019, pp. 121–130. DOI: 10.1145/3323679.3326510.

[Ber95] D. P. Bertsekas. *Dynamic programming and optimal control*. 4th ed. Vol. 1. Athena Scientific, 1995.

[Bha+21] L. Bhatia, I. Tomić, A. Fu, M. Breza, and J. A. Mccann. "Control Communication Co-Design for Wide Area Cyber-Physical Systems." In: *ACM Transactions on Cyber-Physical Systems* 5.2 (2021). DOI: 10.1145/3418528.

[Bil+15] A. Bildea, O. Alphand, F. Rousseau, and A. Duda. "Link quality estimation with the Gilbert-Elliot model for wireless sensor networks." In: *IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE. 2015, pp. 2049–2054.

[Blo+13] B. Bloessl, C. Leitner, F. Dressler, and C. Sommer. "A GNU radio-based IEEE 802.15. 4 testbed." In: *12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN 2013)* (2013), pp. 37–40.

[BSU19] B. Buyukates, A. Soysal, and S. Ulukus. "Age of information in multihop multicast networks." In: *Journal of Communications and Networks* 21.3 (2019), pp. 256–267. DOI: 10.1109/JCN.2019.000032.

[BT08] D. Bertsekas and J. N. Tsitsiklis. *Introduction to probability*. Vol. 1. Athena Scientific, 2008.

[Cap+13] F. Capozzi, G. Piro, L. Grieco, G. Boggia, and P. Camarda. "Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey." In: *IEEE Communications Surveys & Tutorials* 15.2 (2013), pp. 678–700. DOI: 10.1109/SURV.2012.060912.00100.

[CB21] E. Calvanese Strinati and S. Barbarossa. "6G networks: Beyond Shannon towards semantic and goal-oriented communications." In: *Computer Networks* 190 (2021), p. 107930. DOI: https://doi.org/10.1016/j.comnet.2021.107930.

[CCE14] M. Costa, M. Codreanu, and A. Ephremides. "Age of information with packet management." In: *IEEE International Symposium on Information Theory*. 2014, pp. 1583–1587. DOI: 10.1109/ISIT.2014.6875100.

[Cer03] A. Cervin. "Integrated Control and Real-Time Scheduling." In: *PhD Dissertation* (2003). Lund Institute of Technology, Lund, Sweden.

[CGL20] H. Chen, Y. Gu, and S.-C. Liew. "Age-of-Information Dependent Random Access for Massive IoT Networks." In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2020, pp. 930–935. DOI: 10.1109/INFOCOMWKSHPS50562.2020.9162973.

[Cha+19] J. P. Champati, M. H. Mamduhi, K. H. Johansson, and J. Gross. “Performance Characterization Using AoI in a Single-loop Networked Control System.” In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2019, pp. 197–203. DOI: 10.1109/INFCOMW.2019.8845114.

[Che+22] X. Chen, K. Gatsis, H. Hassani, and S. S. Bidokhti. “Age of Information in Random Access Channels.” In: *IEEE Transactions on Information Theory* (2022). DOI: 10.1109/TIT.2022.3180965.

[Chi+21] F. Chiariotti, O. Vikhrova, B. Soret, and P. Popovski. “Peak Age of Information Distribution for Edge Computing With Wireless Links.” In: *IEEE Transactions on Communications* 69.5 (2021), pp. 3176–3191. DOI: 10.1109/TCOMM.2021.3053038.

[Chi+22] F. Chiariotti, O. Vikhrova, B. Soret, and P. Popovski. “Age of Information in Multihop Connections With Tributary Traffic and No Preemption.” In: *IEEE Transactions on Communications* 70.10 (2022), pp. 6718–6733. DOI: 10.1109/TCOMM.2022.3202946.

[Cor+10] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, et al. “Environmental Wireless Sensor Networks.” In: *Proceedings of the IEEE* 98.11 (2010), pp. 1903–1917. DOI: 10.1109/JPROC.2010.2068530.

[Cue+19] A. Cuenca, D. J. Antunes, A. Castillo, P. Garcia, B. A. Khashooei, et al. “Periodic Event-Triggered Sampling and Dual-Rate Control for a Wireless Networked Control System With Applications to UAVs.” In: *IEEE Transactions on Industrial Electronics* 66.4 (2019), pp. 3157–3166. DOI: 10.1109/TIE.2018.2850018.

[DPS20] E. Dahlman, S. Parkvall, and J. Skold. *5G NR: The next generation wireless access technology*. Academic Press, 2020.

[Dre+05] M. Drew, X. Liu, A. Goldsmith, and K. Hedrick. “Networked Control System Design over a Wireless LAN.” In: *Proceedings of the 44th IEEE Conference on Decision and Control*. 2005, pp. 6704–6709. DOI: 10.1109/CDC.2005.1583239.

[FKB19a] S. Farazi, A. G. Klein, and D. R. Brown. “Age of Information with Unreliable Transmissions in Multi-Source Multi-Hop Status Update Systems.” In: *53rd Asilomar Conference on Signals, Systems, and Computers*. 2019, pp. 2017–2021. DOI: 10.1109/IEEECONF44664.2019.9048736.

[FKB19b] S. Farazi, A. G. Klein, and D. R. Brown. “Fundamental bounds on the age of information in multi-hop global status update networks.” In: *Journal of Communications and Networks* 21.3 (2019), pp. 268–279. DOI: 10.1109/JCN.2019.000038.

[FKRB19] S. Farazi, A. G. Klein, and D. Richard Brown. "Average Age of Information in Multi-Source Self-Preemptive Status Update Systems with Packet Delivery Errors." In: *53rd Asilomar Conference on Signals, Systems, and Computers*. 2019, pp. 396–400. DOI: 10.1109/IEEECONF44664.2019.9048914.

[GC10] R. A. Gupta and M.-Y. Chow. "Networked Control System: Overview and Research Trends." In: *IEEE Transactions on Industrial Electronics* 57.7 (2010), pp. 2527–2535. DOI: 10.1109/TIE.2009.2035462.

[Gho+21] S. Ghosh, A. Mondal, D. Roy, P. H. Kindt, S. Dey, et al. "Proactive Feedback for Networked CPS." In: *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. 2021, pp. 164–173. DOI: 10.1145/3412841.3441897.

[Gil60] E. N. Gilbert. "Capacity of a burst-noise channel." In: *Bell system technical journal* 39.5 (1960), pp. 1253–1265.

[GRP15] K. Gatsis, A. Ribeiro, and G. J. Pappas. "Decentralized Channel Access for Wireless Control Systems." In: *IFAC-PapersOnLine* 48.22 (2015), pp. 209–214. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.10.332.

[Han+20] Z. Han, J. Liang, Y. Gu, and H. Chen. "Software-Defined Radio Implementation of Age-of-Information-Oriented Random Access." In: *46th Annual Conference of the IEEE Industrial Electronics Society (IECON)*. 2020, pp. 4374–4379.

[Han+22] R. Han, Y. Wen, L. Bai, J. Liu, and J. Choi. "Age of Information Aware UAV Deployment for Intelligent Transportation Systems." In: *IEEE Transactions on Intelligent Transportation Systems* 23.3 (2022), pp. 2705–2715. DOI: 10.1109/TITS.2021.3117974.

[HDT13] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel. "Periodic Event-Triggered Control for Linear Systems." In: *IEEE Transactions on Automatic Control* 58.4 (2013), pp. 847–861. DOI: 10.1109/TAC.2012.2220443.

[HH08] G. Hasslinger and O. Hohlfeld. "The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet." In: *14th GI/ITG Conference - Measurement, Modelling and Evalutation of Computer and Communication Systems*. 2008, pp. 1–15.

[HMD17] Y.-P. Hsu, E. Modiano, and L. Duan. "Age of information: Design and analysis of optimal scheduling algorithms." In: *IEEE International Symposium on Information Theory (ISIT)*. 2017, pp. 561–565. DOI: 10.1109/ISIT.2017.8006590.

[HMD20] Y.-P. Hsu, E. Modiano, and L. Duan. “Scheduling Algorithms for Minimizing Age of Information in Wireless Broadcast Networks with Random Arrivals.” In: *IEEE Transactions on Mobile Computing* 19.12 (2020), pp. 2903–2915. DOI: 10.1109/TMC.2019.2936199.

[Ino+19] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka. “A General Formula for the Stationary Distribution of the Age of Information and Its Application to Single-Server Queues.” In: *IEEE Transactions on Information Theory* 65.12 (2019), pp. 8305–8324. DOI: 10.1109/TIT.2019.2938171.

[Kad+16] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano. “Minimizing the Age of Information in broadcast wireless networks.” In: *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2016, pp. 844–851. DOI: 10.1109/ALLERTON.2016.7852321.

[Kad+18] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano. “Scheduling Policies for Minimizing Age of Information in Broadcast Wireless Networks.” In: *IEEE/ACM Transactions on Networking* 26.6 (2018), pp. 2637–2650. DOI: 10.1109/TNET.2018.2873606.

[Kau+11] S. Kaul, M. Gruteser, V. Rai, and J. Kenney. “Minimizing age of information in vehicular networks.” In: *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. 2011, pp. 350–358. DOI: 10.1109/SAHCN.2011.5984917.

[Kl19] M. Klügel, M. H. Mamduhi, S. Hirche, and W. Kellerer. “AoI-Penalty Minimization for Networked Control Systems with Packet Loss.” In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2019, pp. 189–196. DOI: 10.1109/INFCOMW.2019.8845106.

[KM21] I. Kadota and E. Modiano. “Age of Information in Random Access Networks with Stochastic Arrivals.” In: *IEEE Conference on Computer Communications (INFOCOM)*. 2021, pp. 1–10. DOI: 10.1109/INFOCOM42981.2021.9488897.

[Kos+17] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis. “Age and value of information: Non-linear age case.” In: *IEEE International Symposium on Information Theory (ISIT)*. 2017, pp. 326–330. DOI: 10.1109/ISIT.2017.8006543.

[Kos+20] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis. “Non-linear Age of Information in a Discrete Time Queue: Stationary Distribution and Average Performance Analysis.” In: *IEEE International Conference on Communications (ICC)*. 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148775.

[KRM21] I. Kadota, M. S. Rahman, and E. Modiano. "WiFresh: Age-of-Information from Theory to Implementation." In: *International Conference on Computer Communications and Networks (ICCCN)*. 2021, pp. 1–11. DOI: 10.1109/ICCCN52240.2021.9522228.

[KYG12a] S. Kaul, R. Yates, and M. Gruteser. "Real-time status: How often should one update?" In: *IEEE Conference on Computer Communications (INFOCOM)*. 2012, pp. 2731–2735. DOI: 10.1109/INFCOM.2012.6195689.

[KYG12b] S. K. Kaul, R. D. Yates, and M. Gruteser. "Status updates through queues." In: *Annual Conference on Information Sciences and Systems (CISS)*. 2012, pp. 1–6. DOI: 10.1109/CISS.2012.6310931.

[LFJ15] K. Liu, E. Fridman, and K. H. Johansson. "Networked Control With Stochastic Scheduling." In: *IEEE Transactions on Automatic Control* 60.11 (2015), pp. 3071–3076. DOI: 10.1109/TAC.2015.2414812.

[LG04] X. Liu and A. Goldsmith. "Wireless medium access control in networked control systems." In: *American Control Conference*. Vol. 4. 2004, pp. 3605–3610. DOI: 10.23919/ACC.2004.1384471.

[Maa+20] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides. "The Age of Incorrect Information: A New Performance Metric for Status Updates." In: *IEEE/ACM Transactions on Networking* 28.5 (2020), pp. 2215–2228. DOI: 10.1109/TNET.2020.3005549.

[MAE18] A. Maatouk, M. Assaad, and A. Ephremides. "The Age of Updates in a Simple Relay Network." In: *IEEE Information Theory Workshop (ITW)*. 2018, pp. 1–5. DOI: 10.1109/ITW.2018.8613408.

[MAE20] A. Maatouk, M. Assaad, and A. Ephremides. "On the Age of Information in a CSMA Environment." In: *IEEE/ACM Transactions on Networking* 28.2 (2020), pp. 818–831. DOI: 10.1109/TNET.2020.2971350.

[Mag+19] F. Mager, D. Baumann, R. Jacob, L. Thiele, S. Trimpe, et al. "Feedback Control Goes Wireless: Guaranteed Stability over Low-Power Multi-Hop Networks." In: *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*. 2019, pp. 97–108. DOI: 10.1145/3302509.3311046.

[Mai+22] D. Maity, M. H. Mamduhi, S. Hirche, and K. H. Johansson. "Optimal LQG Control of Networked Systems Under Traffic-Correlated Delay and Dropout." In: *IEEE Control Systems Letters* 6 (2022), pp. 1280–1285. DOI: 10.1109/LCSYS.2021.3091492.

[Mam+17] M. H. Mamduhi, A. Molin, D. Tolić, and S. Hirche. "Error-dependent data scheduling in resource-aware multi-loop networked control systems." In: *Automatica* 81 (2017), pp. 209–216.

[Mam17] M. Mamduhi. *State-dependent Medium Access Control for Resource-aware Networked Control Systems*. 2017.

[Mam+22] A. Mamane, M. Fattah, M. E. Ghazi, M. E. Bekkali, Y. Balboul, et al. "Scheduling Algorithms for 5G Networks and Beyond: Classification and Survey." In: *IEEE Access* 10 (2022), pp. 51643–51661. DOI: 10.1109/ACCESS.2022.3174579.

[MEJ16] A. Molin, H. Esen, and K. H. Johansson. "Event-triggered Scheduling for Infrastructure-supported Collaborative Vehicle Control." In: *IFAC-PapersOnLine* 49.22 (2016). 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NECSYS), pp. 31–36. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2016.10.368.

[MEJ19] A. Molin, H. Esen, and K. H. Johansson. "Scheduling networked state estimators based on Value of Information." In: *Automatica* 110 (2019). DOI: https://doi.org/10.1016/j.automatica.2019.108578.

[MH14] A. Molin and S. Hirche. "Price-Based Adaptive Scheduling in Multi-Loop Control Systems With Resource Constraints." In: *IEEE Transactions on Automatic Control* 59.12 (2014), pp. 3282–3295. DOI: 10.1109/TAC.2014.2351892.

[Mun21] A. Munari. "Modern Random Access: An Age of Information Perspective on Irregular Repetition Slotted ALOHA." In: *IEEE Transactions on Communications* 69.6 (2021), pp. 3572–3585. DOI: 10.1109/TCOMM.2021.3060429.

[Ogu+22] T. K. Oguz, E. T. Ceran, E. Uysal, and T. Girici. "Implementation and Evaluation of Age-Aware Downlink Scheduling Policies in Push-Based and Pull-Based Communication." In: *Entropy*. Vol. 24. 5. 2022. DOI: 10.3390/e24050673.

[PAJ11] P. Park, J. Araujo, and K. H. Johansson. "Wireless networked control system co-design." In: *2011 International Conference on Networking, Sensing and Control*. 2011, pp. 486–491. DOI: 10.1109/ICNSC.2011.5874926.

[Pan+22] H. Pan, Y. Zhou, T.-T. Chan, M. Tang, J. Li, et al. "Improving Information Freshness via Backbone-Assisted Cooperative Access Points." In: *IEEE Internet of Things Journal* (2022). DOI: 10.1109/JIOT.2022.3218948.

[Par+18] P. Park, S. Coleri Ergen, C. Fischione, C. Lu, and K. H. Johansson. "Wireless Network Design for Control Systems: A Survey." In: *IEEE Communications Surveys & Tutorials* 20.2 (2018), pp. 978–1013. DOI: 10.1109/COMST.2017.2780114.

[Pop+20] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Şahin. "Semantic-Effectiveness Filtering and Control for Post-5G Wireless Connectivity." In: *Journal of the Indian Institute of Science* (2020), pp. 435–443.

[Pop+22] P. Popovski, F. Chiariotti, K. Huang, A. E. Kalor, M. Kountouris, et al. "A Perspective on Time Toward Wireless 6G." In: *Proceedings of the IEEE* 110.8 (2022), pp. 1116–1146. DOI: 10.1109/JPROC.2022.3190205.

[Å12] K. J. Åström. *Introduction to Stochastic Control Theory*. Courier Corporation, 2012.

[Rad] G. Radio. *GNU Radio: The free & open software radio ecosystem*. URL: https://www.gnuradio.org (Last Accessed: 18.07.2022).

[rM08] K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. 2nd ed. Vol. 2. Princeton University Press, 2008.

[Rob75a] L. G. Roberts. "ALOHA packet system with and without slots and capture." In: *ACM SIGCOMM Computer Communication Review* 5.2 (1975), pp. 28–42.

[Rob75b] L. G. Roberts. "ALOHA packet system with and without slots and capture." In: *ACM SIGCOMM Computer Communication Review* 5.2 (Apr. 1975), pp. 28–42.

[SBC20] W. Saad, M. Bennis, and M. Chen. "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems." In: *IEEE Network* 34.3 (2020), pp. 134–142. DOI: 10.1109/MNET.001.1900287.

[SC19] Y. Sun and B. Cyr. "Sampling for data freshness optimization: Non-linear age functions." In: *Journal of Communications and Networks* 21.3 (2019), pp. 204–219. DOI: 10.1109/JCN.2019.000035.

[Sen09] L. I. Sennott. *Stochastic dynamic programming and the control of queueing systems*. John Wiley & Sons, 2009.

[SHD98] G. Sharma, A. A. Hassan, and A. Dholakia. "Performance evaluation of burst-error-correcting codes on a Gilbert-Elliott channel." In: *IEEE transactions on communications* 46.7 (1998), pp. 846–849.

[SKY19] T. Shreedhar, S. K. Kaul, and R. D. Yates. “An Age Control Transport Protocol for Delivering Fresh Updates in the Internet-of-Things.” In: *IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. 2019, pp. 1–7. DOI: 10.1109/WoWMoM.2019.8793011.

[Sol+22] T. Soleymani, J. S. Baras, S. Hirche, and K. H. Johansson. “Value of Information in Feedback Control: Global Optimality.” In: *IEEE Transactions on Automatic Control* (2022), pp. 1–7. DOI: 10.1109/TAC.2022.3194125.

[Son+18] C. Sonmez, S. Baghaee, A. Ergişi, and E. Uysal-Bıyıkoğlu. “Age-of-Information in Practice: Status Age Measured Over TCP/IP Connections Through WiFi, Ethernet and LTE.” In: *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. 2018, pp. 1–5. DOI: 10.1109/BlackSeaCom.2018.8433695.

[Sor+21] I. Sorkhoh, C. Assi, D. Ebrahimi, and S. Sharafeddine. “Optimizing Information Freshness for MEC-Enabled Cooperative Autonomous Driving.” In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–14. DOI: 10.1109/TITS.2021.3119961.

[SPU20] Y. Sun, Y. Polyanskiy, and E. Uysal. “Sampling of the Wiener Process for Remote Estimation Over a Channel With Random Delay.” In: *IEEE Transactions on Information Theory* 66.2 (2020), pp. 1118–1135. DOI: 10.1109/TIT.2019.2937336.

[SSM18] S. M. Shimly, D. B. Smith, and S. Movassaghi. “Cross-Layer Designs for Body-to-Body Networks: Adaptive CSMA/CA with Distributed Routing.” In: *IEEE International Conference on Communications (ICC)*. 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422792.

[Sun+17] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff. “Update or Wait: How to Keep Your Data Fresh.” In: *IEEE Transactions on Information Theory* 63.11 (2017), pp. 7492–7508. DOI: 10.1109/TIT.2017.2735804.

[TKM17] R. Talak, S. Karaman, and E. Modiano. “Minimizing age-of-information in multi-hop wireless networks.” In: *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2017, pp. 486–493. DOI: 10.1109/ALLERTON.2017.8262777.

[TM19] V. Tripathi and E. Modiano. “A Whittle Index Approach to Minimizing Functions of Age of Information.” In: *57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2019, pp. 1160–1167. DOI: 10.1109/ALLERTON.2019.8919842.

[TM21] V. Tripathi and E. Modiano. "Age Debt: A General Framework for Minimizing Age of Information." In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2021, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS51825.2021.9484621.

[Tro+21] M. Trobinger, G. de Albuquerque Gleizer, T. Istomin, M. Mazo, A. L. Murphy, et al. "The Wireless Control Bus: Enabling Efficient Multi-Hop Event-Triggered Control with Concurrent Transmissions." In: *ACM Transactions on Cyber-Physical Systems* 6.1 (2021). DOI: 10.1145/3485467.

[TS22.104] 3GPP. *Service requirements for cyber-physical control applications in vertical domains (Release 18)*. Tech. rep. 2021.

[TS22.261] 3GPP. *Technical Specification Group Services and System Aspects: Service requirements for the 5G System*. Tech. rep. 2022.

[TS23.501] 3GPP. *System architecture for the 5G System (5GS) (Release 18)*. Tech. rep. 2022.

[TTM22] V. Tripathi, R. Talak, and E. Modiano. "Information Freshness in Multihop Wireless Networks." In: *IEEE/ACM Transactions on Networking* (2022), pp. 1–16. DOI: 10.1109/TNET.2022.3201751.

[Uys+22] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, et al. "Semantic Communications in Networked Systems: A Data Significance Perspective." In: *IEEE Network* 36.4 (2022), pp. 233–240. DOI: 10.1109/MNET.106.2100636.

[Vil+16] M. Vilgelm, M. H. Mamduhi, W. Kellerer, and S. Hirche. "Adaptive Decentralized MAC for Event-Triggered Networked Control Systems." In: *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. HSCC '16. Vienna, Austria, 2016, pp. 165–174. DOI: 10.1145/2883817.2883829.

[WC96] H. S. Wang and P.-C. Chang. "On verifying the first-order Markovian assumption for a Rayleigh fading channel model." In: *IEEE Transactions on Vehicular Technology* 45.2 (1996), pp. 353–357.

[WM95] H. S. Wang and N. Moayeri. "Finite-state Markov channel-a useful model for radio communication channels." In: *IEEE transactions on vehicular technology* 44.1 (1995), pp. 163–171.

[WY01] G. Walsh and H. Ye. "Scheduling of networked control systems." In: *IEEE Control Systems Magazine* 21.1 (2001), pp. 57–65. DOI: 10.1109/37.898792.

[WYB02] G. Walsh, H. Ye, and L. Bushnell. “Stability analysis of networked control systems.” In: *IEEE Transactions on Control Systems Technology* 10.3 (2002), pp. 438–446. DOI: 10.1109/87.998034.

[Yan+22] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. V. Poor. “Spatiotemporal Analysis for Age of Information in Random Access Networks Under Last-Come First-Serve With Replacement Protocol.” In: *IEEE Transactions on Wireless Communications* 21.4 (2022), pp. 2813–2829. DOI: 10.1109/TWC.2021.3116041.

[Yat18] R. D. Yates. “Age of information in a network of preemptive servers.” In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2018, pp. 118–123. DOI: 10.1109/INFCOMW.2018.8406966.

[Yat21] R. D. Yates. “The Age of Gossip in Networks.” In: *IEEE International Symposium on Information Theory (ISIT)*. 2021, pp. 2984–2989. DOI: 10.1109/ISIT45174.2021.9517796.

[Yat+21] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, et al. “Age of Information: An Introduction and Survey.” In: *IEEE Journal on Selected Areas in Communications* 39.5 (2021), pp. 1183–1210. DOI: 10.1109/JSAC.2021.3065072.

[YK17] R. D. Yates and S. K. Kaul. “Status updates over unreliable multiaccess channels.” In: *IEEE International Symposium on Information Theory (ISIT)*. 2017, pp. 331–335. DOI: 10.1109/ISIT.2017.8006544.

[YWB00] H. Ye, G. Walsh, and L. Bushnell. “Wireless local area networks in the manufacturing industry.” In: *American Control Conference (ACC)*. Vol. 4. 2000, pp. 2363–2367. DOI: 10.1109/ACC.2000.878603.

[ZGK13] L. Zhang, H. Gao, and O. Kaynak. “Network-Induced Constraints in Networked Control Systems - A Survey.” In: *IEEE Transactions on Industrial Informatics* 9.1 (2013), pp. 403–416. DOI: 10.1109/TII.2012.2219540.

[Zou+19] L. Zou, Z. Wang, Q.-L. Han, and D. Zhou. “Moving Horizon Estimation for Networked Time-Delay Systems Under Round-Robin Protocol.” In: *IEEE Transactions on Automatic Control* 64.12 (2019), pp. 5191–5198. DOI: 10.1109/TAC.2019.2910167.

[ZZN20] X. Zheng, S. Zhou, and Z. Niu. “Urgency of Information for Context-Aware Timely Status Updates in Remote Control Systems.” In: *IEEE Transactions on Wireless Communications* 19.11 (2020), pp. 7237–7250. DOI: 10.1109/TWC.2020.3009881.

List of Figures

List of Tables