Transmission Scheduling of Periodic Real-Time Traffic in Wireless Networks

by

Igor Kadota

BSc Electrical Engineering, Technological Institute of Aeronautics, 2010
SM Telecommunications, Technological Institute of Aeronautics, 2013

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2016

Author . . . .
Department of Aeronautics and Astronautics
July 29, 2016

Certified by . . . .
Eytan Modiano
Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by . . . .
Paulo C. Lozano
Associate Professor, Department of Aeronautics and Astronautics
Chair, Graduate Program Committee

Transmission Scheduling of Periodic Real-Time Traffic in Wireless Networks

by Igor Kadota

Submitted to the Department of Aeronautics and Astronautics on July 29, 2016,
in partial fulfillment of the requirements for the degree of
Master of Science

Abstract

An increasing number of applications rely on wireless networks for distributing information. Communicating time-sensitive data such as position, video, voice and telemetry, can be particularly challenging in wireless networks due to packet losses. In this thesis, we consider a single-hop wireless network, in which a base station is sending time-sensitive data packets to a set of clients. Our goal is to study transmission scheduling strategies for real-time traffic. Even though this problem has been explored in the literature, we present novel results that provide useful insight into the optimal scheduling problem.

Previous work considered the problem of maximizing the throughput of networks with instantaneous feedback and without feedback. We address the general case of delayed feedback. Delayed feedback is particularly important for communication systems in which the round trip delay is much greater than the packet transmission time, and it has a significant impact on the scheduling decisions and network performance.

In addition, we consider the case of clients receiving multiple parallel packet flows with heterogeneous deadlines. It is a well-known result that the Shortest Time to Extinction (STE) policy optimizes the throughput *in wired networks*. In this thesis, we establish a class of *wireless* networks for which the STE policy is throughput-optimal, i.e. minimizes the expected number of packets that expire due to the deadlines.

Finally, we study the wireless network from the perspective of the Age of Information (AoI). This recently proposed performance metric represents the freshness of the information at the clients. We use Dynamic Programming to formulate and solve the problem of characterizing the scheduling policy that minimizes the AoI. The AoI metric is compared with throughput, and insights are drawn from numerical results. Simulations suggest that AoI-optimal policies are always throughput-optimal, while the converse is not true.

Thesis Supervisor: Eytan Modiano
Title: Professor, Department of Aeronautics and Astronautics

for my loving family, Conceição, Jorge, Paola, Débora and Sakura.

Acknowledgments

Upon arriving at MIT, one of the first things I was told is that I was very fortunate to have Prof. Eytan Modiano as adviser. Indeed, he has been incredibly encouraging, providing mentoring in research, writing, presenting, teaching, and others. Despite his very busy schedule, he always finds the time to meet and talk about the various aspects of our research. I would like to thank Prof. Modiano for the time and effort, they are very much appreciated.

I'm thankful to Prof. Elif Uysal-Biyikoglu for her research guidance. It has been an immense pleasure working with someone so enthusiastic about research and with so many great insights. I would like to sincerely thank Prof. Moe Win for the meetings during his Statistics course and for the encouragement before the Qualifying Exam. Also, thanks to Rahul and Yu-Pin for the many technical discussions.

I would like to express my sincerest gratitude to Prof. Alessandro Anzaloni for all the support during my time in Italy, in Brazil and here at MIT. Prof. Anzaloni has been a great teacher, mentor and friend.

I am very grateful towards the people in Ashdown, LIDS, AeroAstro and MIT for fostering a fun and creative work environment. I am thankful for sharing an office with many good friends: James, Matt, Quan, Hongyi, Rajat, Anurag, Thomas, Jianan, Marzieh, Qingkai and Abhishek. The time here would not have been nearly as much fun without[1] Mario, Zied, Tianheng, Prince, Eren, Iulia, Mayara, Josimar, Karthik, Spencer, Victor, Arthur, Hendrata, Brendan, Tiffany, Casey, Hamza, Hsin-Yu, Henry, Sourav and Harry.

[1] I am sure I am missing important people, so I am apologizing in advance.

Thanks to Jean Sofronas, Joyce Light, Rachel Cohen, Lynne Dell, Brian Jones, Petra Aliberti, Jennifer Donovan and Emily Cheng, for always helping me with *everything*. I would like to especially thank Beth Marois, who helped me numerous times during my transition from Brazil to MIT. It was a particularly turbulent time and Beth was always there to help.

Last but certainly not least, I want to thank my parents, Jorge and Conceição, my sister, Paola, and my girlfriend Débora, for their support throughout the years. They have always encouraged me to pursue my dreams. Without their love and support, I would not be here, I would not be able to achieve anything and I would not be the person I am today.

Thank you!

I would like to thank the fellowship received from the Coordination for the Improvement of Higher Education Personnel (CAPES-Brazil) during the research conducted to achieve the results presented in this thesis.

Contents

List of Figures

Chapter 1

Introduction

Wireless networks provide a flexible platform to support a variety of applications such as voice, multimedia, e-mail, monitoring of sensors, control of industrial plants, et al. The increasing number of clients using those applications leads to a growing demand for various Quality of Service (QoS) requirements. For complying with those new requirements, future wireless systems will need to provide services which are tailored to the application.

According to [23], it is expected that 5G wireless systems will provide four different types of service, called operating modes. Figure 1-1 characterizes the operating modes according to the throughput requirement and typical number of interconnected users they support. Operating mode (D) refers to applications with a large number of communicating devices. A central node monitoring the information generated by numerous sensors in an industrial plant is an example. Mode (C) indicates applications that require extremely high data rates, e.g. virtual reality and immersive internet. The remaining modes, (A) and (B), point to applications that require service to be reliably guaranteed, i.e. guaranteed almost at all times ($>99\%$ availability). The traffic generated by applications in (A) requires strict delay guarantees and has low data rate, while the traffic in (B) is less time-sensitive and demands a higher data rate. Coordinating autonomous vehicles and controlling industrial plants are some examples of applications in mode (A).

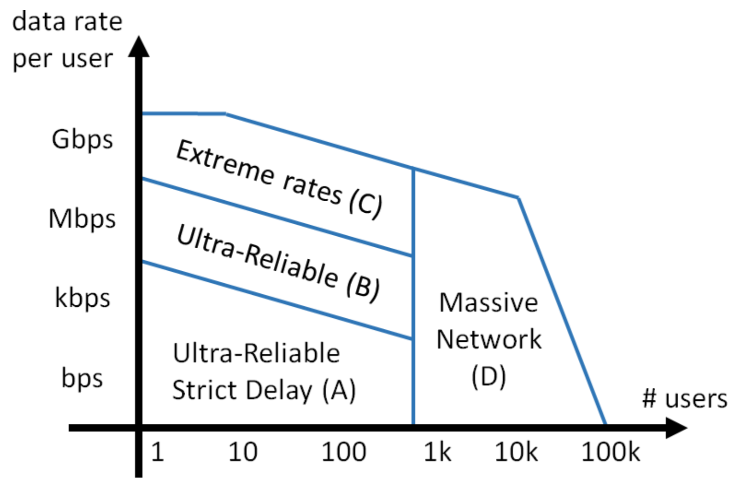

Figure 1-1: Operating Modes of wireless systems. The axes in the figure represent orders of magnitude. Figure is adapted from [23, Fig. 1].

In this thesis, we consider wireless systems that support applications with stringent delay requirements, similar to the ones in mode (A). We are particularly interested in applications that generate real-time traffic. Two important requirements associated with this type of traffic are the maximum time to deliver a packet and the minimum delivery ratio. The first imposes a deadline by which the packets must be delivered (a strict delay guarantee), while the second limits the fraction of packets that miss the deadline (a minimum service to be provided). Meeting those requirements over an unreliable wireless channel is challenging due to transmission errors that result in packet losses.

We consider a single-hop wireless network with a base station (BS) sending packets to a set of clients[1]. Packets are generated at the BS periodically and are transmitted to the clients through the shared wireless channel. At any given time, at most one packet can be scheduled for transmission and the service time of the packet is deterministic. Following every transmission, the client sends a feedback signal to the BS if the packet was received correctly. Undelivered packets are dropped from the network if and when their deadline expires. Notice that the deadline imposes a limit to the packet delay. Considering this network, our goal is to study transmission scheduling policies that guarantee a minimum delivery ratio to the clients.

[1]All results in this thesis are also valid for the case of clients sending packets to the BS. The network models are equivalent, as depicted in Figure 6-1.

It is important to emphasize that guaranteeing a minimum delivery ratio does not necessarily result in packets being delivered to the clients regularly. The delivery ratio requirement can be fulfilled even if long periods with no delivery occur, as long as those are balanced by periods of consecutive deliveries. For attending to applications that need fresh data being delivered to the clients regularly, in Chapter 5 we study scheduling policies that minimize the Age of Information. This performance metric measures the freshness of the data at the clients. The relationship between Age of Information and delivery ratio is also discussed in Chapter 5.

1.1 Related Work

The problem of optimizing scheduling decisions in networks that support traffic with deadlines has been explored in the literature in a variety of contexts and under diverse system conditions. *Wired* multi-hop networks with unicast traffic have been studied in [5, 20, 21]. In [5], a tree network with a single destination was considered. It was shown that forwarding the packet with shortest time to extinction (STE policy) maximizes the delivery ratio. In [21], a more general network with multiple source destination pairs was addressed. An online algorithm for joint admission and scheduling was developed using time slot reservations. A similar network setting, using a different analytical model, was considered in [20], where a tractable approach for scheduling in multi-hop networks based on adapting the service discipline to meet delivery ratio constraints was designed.

In other related works, researchers have studied the delivery ratio in *wireless single-hop networks* [11, 12, 13, 14, 19]. The authors in [12] proposed an analytical framework to model a network supporting unicast traffic with deadlines and instantaneous transmission feedback. Based on this framework, two debt-based policies that meet any feasible delivery ratio requirements were presented. Articles [11, 13, 14, 19] extend the model in [12] to a variety of scenarios: [13] considers variable-bit-rate traffic; [14] generalizes for heterogeneous deadlines and time-varying channel; [19] models multicast traffic with instant feedback; and [11] models broadcast traffic with no feedback.

In recent work, there has been a growing interest in the Age of Information (AoI) [1, 6, 15, 16]. This performance metric measures the freshness of the information at the destination. The authors of [6, 15, 16] adressed the problem of minimizing the AoI using Queueing Theory. In [16], the optimal server utilization for M/M/1, M/D/1 and D/M/1 systems was found. The impact of packet management in the AoI was discussed in [6] by analyzing M/M/1/1 and M/M/1/2* systems. The average peak AoI (PAoI) was minimized for the case of multi-class M/G/1 queues in [15]. The authors of [1] use Dynamic Programming to address the problem of minimizing the AoI in networks with energy replenishment constraints by optimizing the mechanism of generating new packets.

1.2 Outline and Main Contributions of the Thesis

The remainder of this thesis is organized as follows.

- **Chapter 2** introduces the background relevant to this thesis. First, the network model that is extended in later chapters is presented. Then, the problem of maximizing the delivery ratio of the clients is formulated as a Dynamic Program and, finally, the scheduling policies that satisfy any feasible set of minimum delivery ratio requirements are characterized.

- **Chapter 3** extends the network model in Chapter 2 to account for a general feedback mechanism. The problem is formulated and solved using Dynamic Programming. The impact of the feedback mechanism in the performance of the scheduling policies is discussed. Finally, a low-complexity heuristic algorithm that uses insights developed from the optimal policies is proposed and numerically assessed.

- **Chapter 4** generalizes the network model in Chapter 2 to allow for each client having multiple packet flows with heterogeneous deadlines. The delivery ratio maximization problem is addressed using DP. Under some symmetry assumptions, we use Stochastic Dominance [26] to establish that the Shortest Time to Extinction policy minimizes the expected total number of packet losses due to the deadlines. This is a well-known result for *wired* networks, that was not yet established for wireless networks.

- **Chapter 5** investigates the network model in Chapter 2 from the perspective of the Age of Information. The problem of minimizing AoI is formulated and solved using Dynamic Programming. The performances of AoI-optimal scheduling policies is numerically assessed and compared to throughput-optimal policies. Simulation results suggest that all AoI-optimal scheduling policies are also throughput-optimal.

- **Chapter 6** concludes this thesis, providing some final remarks and outlining the main contributions and insights of each chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Background

In this chapter, we introduce the background relevant to the thesis. The network model that is extended in later chapters is presented in Section 2.1 together with some useful definitions. In Section 2.2, the problem of maximizing the delivery ratio of the clients is formulated as a Dynamic Program. Finally, in Section 2.3, the scheduling policies that can satisfy any feasible set of minimum delivery ratio requirements are characterized. The results presented in this chapter were introduced in [12, 19].

2.1 Network Model

Consider a wireless system with a base station (BS) sending unicast data packets with deadlines to N clients. Time is divided into slots, and T successive slots form a frame. The BS generates one packet per client at the beginning of every frame and drops those packets that were not delivered by the end of the frame. Notice that the maximum time to deliver a packet is the frame length, T. The kth frame comprises slots $t \in \{kT, \cdots, (k+1)T-1\}$, where $t, k \in \mathbb{Z}^+$ are the slot index and the frame index, respectively. In Figure 2-1, an illustration of the system is provided.

In a slot, the BS transmits the packet of a selected client $i \in \{1, \cdots, N\}$ over the wireless channel. The packet is successfully delivered to that client with probability $p_i \in (0,1]$ and a transmission error occurs with probability $1-p_i$. In either case, the client sends an instantaneous feedback signal to the BS through an error-free control channel. The

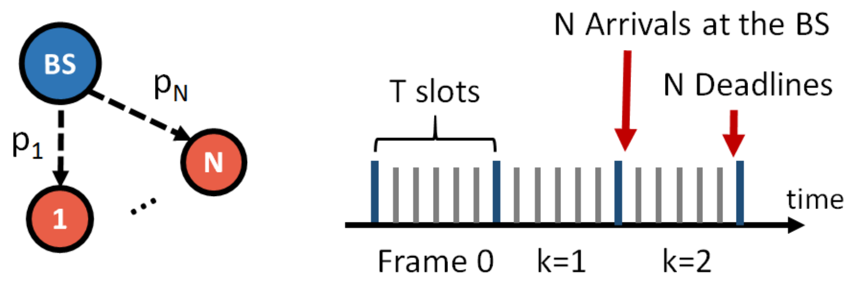

Figure 2-1: On the left, the wireless single-hop network with a BS and N clients. On the right, a timeline with the relationship between the frame structure and arrivals/deadlines.

ACK/NACK reaches the BS right after the end of the transmission, as shown in Figure 2-2.

This network model allows us to consider the packet scheduling problem over a single frame only, as in [19], due to the periodicity of the system. Packets arrive at the beginning of each frame and undelivered packets are dropped at the end of the same frame. The system is stochastically renewed [7] at the start of each frame, thus, henceforward, we regard the system over the first frame only, i.e. for $k = 0$.

The scheduling policies which are considered in this thesis are *work conserving* and *non-anticipatory*, i.e. policies which never idle when there are undelivered packets in the system, and do not use future information in making decisions. This class of policies is denoted by Π and the policies by $\eta \in \Pi$.

Since packets are dropped from the system when they miss the deadline, a performance metric of interest is the long-term delivery ratio of each client, referred to as its **throughput**. Let $D_i^\eta(k)$ be an indicator random variable that is equal to 1 if, by following policy η, the packet to client i is delivered during the kth frame, and zero otherwise. Then, the throughput of client i under policy η is given by

$$\hat{q}_i^\eta := \liminf_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} D_i^\eta(k). \tag{2.1}$$

Observe that if the same policy is used across frames, by the Law of Large Numbers $\hat{q}_i^\eta = \mathbb{E}[D_i^\eta(0)]$ with probability 1, where the RHS is the expected throughput.

Definition 1. *A given vector of required minimum delivery ratios, $(q_i)_{i=1}^N$, is fulfilled by policy η if and only if $\hat{q}_i^\eta \geq q_i$, for all i, with probability 1.*

A network-wide metric for measuring throughput is the Expected Weighted Sum Throughput (EWST). In particular, let $\vec{\alpha} = (\alpha_i)_{i=1}^{N}$ be a vector of positive client weights. Then, the EWST is expressed as:

$$\sum_{i=1}^{N} \alpha_i \mathbb{E}[D_i^{\eta}(0)] \; . \tag{2.2}$$

This metric has a central role in the problem formulation discussed in the next section.

2.2 Problem Formulation

In this section, the discrete-time system is viewed from the Dynamic Programming [2] standpoint. The objective function considered in this formulation is the EWST and the components of the cost-to-go function (state, control, transition and reward) are associated with the network model as described next.

2.2.1 State, Control, Transition and Reward

Consider the set of clients $\mathcal{N} = \{1, ..., N\}$ and the state space $S = 2^{\mathcal{N}}$, which is the collection of all subsets of clients. Let $s_t \in S$, $t \in \{0, \cdots, T-1\}$, represent the clients that have successfully received their packet by the beginning of the tth time slot, as illustrated in Figure 2-2.

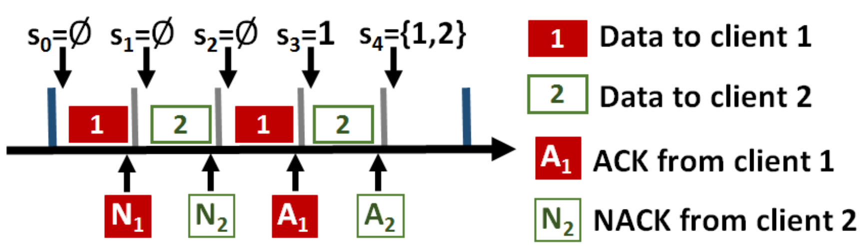

Figure 2-2: Single frame of a network with $N = 2$ and $T = 5$. At the beginning of each slot, the BS selects a packet for transmission, unless all packets were already delivered, namely $s_t = \mathcal{N}$, in which case the BS idles.

Let u_t be the scheduling decision (or control) in time slot t, i.e. the client selected by the BS for transmission in that slot. Recalling that the policies are work conserving, the set of allowed scheduling decisions can be defined as $U_t(s_t) = \mathcal{N} \backslash s_t$. Imposing $u_t \in U_t(s_t)$,

$\forall t$, guarantees that the BS will only idle when all packets have been acknowledged, i.e. $U_t(s_t = \mathcal{N}) = \emptyset$. The case in which the BS idles is illustrated in Figure 2-2.

State transitions are determined by the scheduling decision, u_t, and by the feedback signal received at the end of the slot. If the BS idles during slot t, then the state does not change

$$P(s_{t+1} = s_t | s_t, u_t = \emptyset) = 1 \; . \tag{2.3}$$

In case the BS transmits a packet to client i, then the state at the beginning of slot $t+1$ will depend upon the outcome of the transmission. The state transition from slot t to slot $t+1$ is characterized by the probabilities

$$P(s_{t+1} = s_t | s_t, u_t = i) = 1 - p_i \; ; \qquad \text{[NACK is received]} \tag{2.4}$$

$$P(s_{t+1} = \{s_t \cup i\} | s_t, u_t = i) = p_i \; . \qquad \text{[ACK is received]} \tag{2.5}$$

The last concept to be introduced prior to the actual DP formulation is the reward function. The reward is directly related to the EWST and is associated with the successful delivery of a packet, in particular with the transition (2.5). It is defined as

$$g_t(s_{t+1}, s_t) = \begin{cases} \alpha_i & \text{if } s_{t+1} = s_t \cup \{i\} \; ; \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

The system gets a reward of α_i when a packet is successfully delivered to client i and zero otherwise. With this last concept, the DP formulation is presented next.

2.2.2 Dynamic Program

Within a frame, the system evolves in steps and yields a reward which is additive over time, making it adequate for a DP formulation. For a given $\vec{\alpha}$, the problem of optimizing the EWST[1] is defined as

$$EWST(\vec{\alpha}) := \max_{\eta \in \Pi} \sum_{i=1}^{N} \alpha_i \mathbb{E}\left[D_i^{\eta}(0)\right] , \tag{2.7}$$

[1] The minimum delivery ratio requirement is reintroduced in the next section.

and solved by applying the cost-to-go function $J_t(s_t)$ iteratively, backwards in time. The initial value of the cost-to-go is $J_T(s_T) = 0, \forall s_T \in S$, and the recursion is given by

$$\begin{aligned} J_t(s_t) :&= \max_{u_t \in U_t(s_t)} \mathbb{E}_{s_{t+1}}[g_t(s_{t+1}, s_t) + J_{t+1}(s_{t+1})] \\ &= \max_{i \in U_t(s_t)} [\alpha_i p_i + p_i J_{t+1}(s_t \cup i) + (1 - p_i) J_{t+1}(s_t)]. \end{aligned} \tag{2.8}$$

At each step t and for every possible s_t, the value of $J_t(s_t)$ is attained by choosing the control u_t^* which maximizes (2.8). By keeping track of the u_t^* chosen for every (t, s_t) pair, the optimal policy η^* is obtained. The output of the recursion at $t = 0$ is the optimal performance $J_0(\emptyset) = EWST(\vec{\alpha})$ associated with η^*.

The mechanism to obtain the optimal network performance for a given $\vec{\alpha}$ was described. However, the actual constraint of the system model is not the weight vector, but a vector of minimum delivery ratios, $(q_i)_{i=1}^N$. In the next section, a scheduling policy that fulfills any feasible vector of q_i is described.

2.3 Feasible Region and Optimal Policy

Prior to characterizing the optimal policy derived in [12], we introduce the concepts of Feasible Throughput Region and Feasibility Optimal Scheduling Policy, which are also used in later chapters.

Definition 2. *The Feasible Region is the set of throughput vectors* $\left(\mathbb{E}\left[D_i^\eta(0)\right]\right)_{i=1}^N$ *that can be achieved by the network when employing any admissible policy,* $\eta \in \Pi$.

Using the method in [19], we characterize the Feasible Throughput Region by evaluating its boundaries. For a given fixed weight vector, the DP solution to (2.7) yields the maximum throughput associated with that specific $\vec{\alpha}$ and the optimal policy η^*. Therefore, by sweeping $\vec{\alpha}$ and iteratively solving the DP, the optimal throughput vectors, denoted by $(\mathbb{E}[D_i^*(0)])_{i=1}^N$, are collected. It then follows that the Feasible Region is the convex hull of those vectors. The convex hull represents randomizations between different η^*. In Figure 2-3, the Feasible Throughput Region for a given network setup (T, N, p_i) is illustrated.

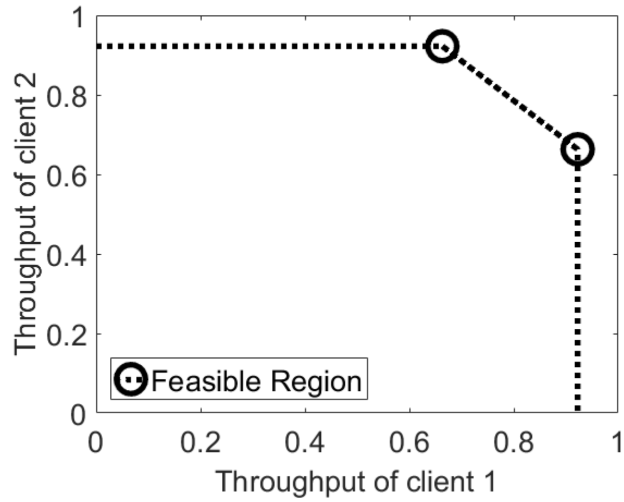

Figure 2-3: Feasible Throughput Region of a two-user wireless network with $T = 5$, $p_1 = p_2 = 0.4$. The circles represent the throughput vectors attained by the optimal policies and the dotted lines represent the convex hull.

Any vector of minimum delivery ratios, $(q_i)_{i=1}^N$, that lies inside the convex hull is called a *feasible* requirement, for it can be fulfilled by a randomization over the optimal policies. A special type of optimality, associated with feasible requirements, is defined next.

Definition 3. *a Feasibility Optimal Scheduling Policy is a transmission strategy that fulfills any feasible vector of minimum delivery ratios,* $(q_i)_{i=1}^N$.

2.3.1 Feasibility Optimal Policy

Prior to introducing the Feasibility Optimal Policy, we first need to describe the Greedy Policy. It prioritizes clients and, in each slot, schedules the client with highest priority. In the context of the network model presented in Section 2.2, the Greedy Policy is as follows: in slot t, the BS transmits $u_t = argmax_{i \in U_t(s_t)} \alpha_i p_i$ and idles only when all clients were delivered. Notice that a fixed weight vector $\vec{\alpha}$ determines a unique priority ordering for the clients and, consequently, a unique Greedy Policy.

According to [19, Theorem 1], for any fixed value of $\vec{\alpha}$, the solution to the EWST maximization problem in (2.7) is always the Greedy Policy. Thus, every vertex of the Feasible Throughput Region is associated with a Greedy Policy. This close relationship between the Greedy Policy and the Feasible Region can be seen by inspecting the optimal policies in Figure 2-3. The black circle on the top left is related to the Greedy Policy that prioritizes client 2 over client 1 and the black circle on the top right is associated with the other Greedy Policy. Since the optimal policies are all Greedy, it follows that a proper randomization among the Greedy Policies can fulfill any feasible throughput requirements.

The randomization mechanism devised in [12] is based on the concept of debt. Since the target of the scheduling policy is to fulfill the throughput requirement, $(q_i)_{i=1}^N$, it is natural to define the debt of client i, denoted $d_i(k)$, as the distance between its current throughput and its throughput requirement. The Largest Weighted-Delivery Debt First Policy randomizes between the Greedy Policies by periodically updating $d_i(k)$ and assigning it to the client weight $\alpha_i \leftarrow d_i(k)$, so that more priority is given to clients with higher debts. A complete description of the policy follows. It is shown in [12, Theorem 3] that this policy is Feasibility Optimal.

LARGEST WEIGHTED-DELIVERY DEBT FIRST POLICY

(i) At the beginning of the frame k, calculate for each client:

- the total number of packets delivered $Q_i(k)$;
- the delivery debt $d_i(k) = (kq_i - Q_i(k))/p_i$.

(ii) Assign $\alpha_i = \max\{d_i(k), 0\}$ and calculate $\alpha_i p_i$ for every client;

(iii) Employ the Greedy Policy during frame k:

- in each slot, serve user $i = argmax_{i \in U_t(s_t)} \alpha_i p_i$.

The network model and problem formulation presented in this section are generalized in different directions in Chapters 3, 4 and 5. In Chapter 3, we introduce and discuss feedback delay. In Chapter 4, we consider clients having multiple parallel packet flows with heterogeneous deadlines. And, finally, in Chapter 5, we analyze the network from the perspective of the recently proposed Age of Information metric.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

General Feedback Mechanism

Part of this chapter was published in the thesis of Kyu Kim [17, Chapter 3] and is presented here for completeness. Our main contributions in this chapter are providing the proof for Theorem 4, developing insights into the optimal policies, proposing a low-complexity heuristic algorithm and obtaining the numerical results.

3.1 Introduction

Feedback is often used to improve transmission reliability. After receiving a packet, the receiver sends an acknowledgment to the sender if the data was received correctly. The feedback mechanism can be implemented in a number of different ways, depending on the network characteristics. Latency, the ratio of the round trip delay time to the data transmission time, plays an important role in the design and performance of systems employing feedback [3].

From the sender standpoint, the feedback associated with each transmission arrives after a round trip delay. In low latency systems, such as wireless LANs with low data-rate and short round trip delay, waiting for the acknowledgment before transmitting the next packet has minimal impact on performance. On the other hand, when the round trip delay takes much longer than the data transmission, it is necessary to allow multiple data packets transmissions in a row prior to receiving the feedback. However, to the best of our knowledge, the influence of the round trip delay has not been analyzed in the context of scheduling real-time traffic, with the exception of the mentioned thesis [17] and our subsequent paper [18].

Prior literature addressed the specific cases of instantaneous transmission feedback and no feedback. As mentioned in Chapter 2, in the case of instantaneous feedback, the scheduling policy that maximizes the delivery ratio is the Greedy policy. In contrast, in the case of no feedback, the scheduler does not know which packets were delivered or not, and the optimal policy allocates slots to clients a priori, according to the algorithm in [11].

The network model considered in this chapter extends the model in Chapter 2 to account for a general feedback mechanism. This new network model accommodates instantaneous feedback, delayed feedback and no feedback. As discussed later in this chapter, the feedback mechanism has a great influence in the Feasible Region and in the Optimal Scheduling Policy. The throughput-optimal policy is not always Greedy, in fact, for some network setups, prioritizing clients harms the performance and the optimal policy is found to be Round Robin.

The network model with a general feedback mechanism is presented in Section 3.2. The problem is formulated and solved using Dynamic Programming in Section 3.3. In Section 3.4, the Feasible Throughput Region and the Feasibility Optimal Algorithm are described. In Section 3.5, we propose a Heuristic Algorithm based on insights into the optimal policies and also present some numerical results.

3.2 Network Model

Consider the network presented in Chapter 2, with a BS sending unicast packets with hard-deadlines to N clients. Time is divided into slots, and T successive slots form a frame. The BS generates one packet per client at the beginning of every frame and drops those packets that were not delivered by the end of the frame. The kth frame comprises slots $t \in \{kT, \cdots, (k+1)T-1\}$, where $t, k \in \mathbb{Z}^+$ are the slot index and the frame index, respectively.

In a slot, the BS transmits a selected packet over the wireless channel. The packet is successfully delivered to client i with probability $p_i \in (0,1]$ and a transmission error occurs with probability $1-p_i$. In either case, the feedback signal is sent by the client through a

delayed control channel. The ACK/NACK reaches the BS, without errors, d slots after the transmission. Therefore, when a packet is transmitted in slot t, the BS receives the feedback at the end of slot $t+d$, implying that the feedback information is only available for the scheduling decision of slot $t+d+1$ onward. Figure 3-1 illustrates this events.

Similarly to Chapter 2, the network is stochastically renewed at the beginning of every frame, ergo, it is sufficient to consider the problem over a single frame. Hereafter, we regard the system over frame $k=0$ only. Recall that admissible policies are *work conserving* and *non-anticipatory*. This class of policies is represented by Π.

The performance metric of interest for measuring the throughput of the network is the Expected Weighted Sum Throughput defined as

$$\sum_{i=1}^{N} \alpha_i \mathbb{E}[D_i^{\eta}(0)]. \tag{3.1}$$

where $\vec{\alpha} = (\alpha_i)_{i=1}^{N}$ is the vector of positive client weights and $D_i^{\eta}(k)$ is the indicator random variable that is equal to 1 if, by following policy η, the packet is delivered to client i during frame k, and zero otherwise.

3.3 Problem Formulation

In this section, the problem is formulated as a DP, based on the EWST performance metric. Next, the four components of the cost-to-go function (state, control, transition and reward) are presented and the finite-horizon program is solved.

3.3.1 Augmented State and Control

Consider the set of clients $\mathcal{N} = \{1,...,N\}$ and the state space $S = 2^{\mathcal{N}}$. Let $s_t \in S$, $t \in \{0,\cdots,T-1\}$, represent the clients that have received their packet and *whose ACK signal have reached the BS* by the beginning of the tth time slot. The evolution of s_t is illustrated in Figure 3-1. One implication of this definition is that $s_0 = \cdots = s_d = \emptyset$, because the first time a feedback is received in any frame is at the end of slot d.

From the point of view of the BS, clients can be divided into three groups. i) clients

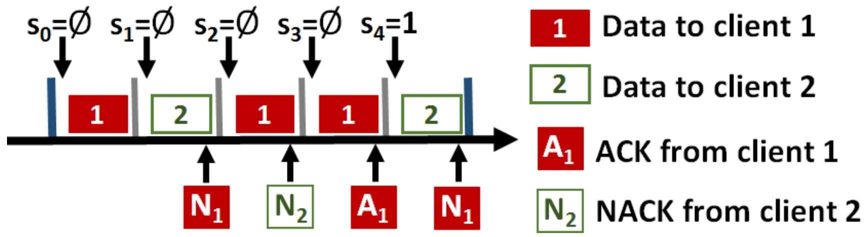

Figure 3-1: Timeline of a frame for $N = 2$, $T = 5$ and $d = 1$. The set s_t represents the clients that have received their packet and whose ACK signal have reached the BS by the beginning of the time slot t.

with confirmed delivered packets, ii) clients that have been served but the feedback signal has not arrived yet and iii) clients that were not served or have confirmed a transmission error (NACK). As each client belongs to one group, a complete representation of the system state can be achieved by characterizing two out of the three groups.

Let u_t be the scheduling decision in time slot t, i.e. the client selected by the BS for transmission in that slot. The set $(u_{t-1},\cdots,u_{t-d})$ depicts the transmissions which have not been acknowledged by the beginning of slot t. Hence, combining s_t with $(u_{t-1},\cdots,u_{t-d})$ gives the augmented state $\tilde{s}_t = (s_t;u_{t-1},\cdots,u_{t-d})$, which fully represents the system.

Since the policies are work conserving, the set of allowed scheduling decisions can be defined as $U_t(s_t) = \mathcal{N}\backslash s_t$. Imposing $u_t \in U_t(s_t)$, $\forall t$, guarantees that the BS will only idle when all packets have been acknowledged, i.e. $U_t(s_t = \mathcal{N}) = \emptyset$.

3.3.2 State Transition

As mentioned, the first feedback arrives to the BS at the end of slot d. This means that the (augmented) state[1] for $t \in \{0,1,\cdots,d-1\}$ changes according to the transition probability

$$P\{\tilde{s}_{t+1} = (s_{t+1} = \emptyset;u_t,u_{t-1},\cdots,u_0)|\tilde{s}_t,u_t\} = 1. \tag{3.2}$$

On the other hand, when $t \in \{d,\cdots,T-1\}$, the transition depends on the feedback to be received at the end of slot t. There are three possible events: the feedback is a NACK, an ACK or it is associated to the transmission of a previously confirmed delivery. These

[1] Henceforth in this chapter, we use the terms augmented state and state interchangeably to denote $\tilde{s}_t$.

events are illustrated in Figure 3-1 at $t = 2$, 3 and 4, respectively. In the case of a NACK, the associated transition probability is

$$P\{\tilde{s}_{t+1} = (s_t; u_t, \cdots, u_{t-d+1}) | \tilde{s}_t, u_t\} = 1 - p_{u_{t-d}}, \tag{3.3}$$

and in the case of an ACK,

$$P\{\tilde{s}_{t+1} = (s_t \cup u_{t-d}; u_t, \cdots, u_{t-d+1}) | \tilde{s}_t, u_t\} = p_{u_{t-d}}. \tag{3.4}$$

Notice that these transitions can only take place at slot t if $u_{t-d} \in U_t(s_t)$. In contrast, $u_{t-d} \notin U_t(s_t)$ yields the deterministic transition

$$P\{\tilde{s}_{t+1} = (s_t; u_t, \cdots, u_{t-d+1}) | \tilde{s}_t, u_t\} = 1. \tag{3.5}$$

With all state transition probabilities defined, in the next section we analyze the reward function associated with each state transition.

3.3.3 Reward Function

The last concept to be introduced prior to the actual DP formulation is the reward function. The reward is directly related to the EWST. From the expression of this performance metric, a straightforward definition for the reward at time t is that the system gets a reward of $\alpha_{u_{t-d}}$ when an ACK is received from client u_{t-d} and zero otherwise.

For describing the reward function, three periods are distinguished. The first, $t \in \{0, \cdots, d-1\}$, in which there is no feedback:

$$g_t(\tilde{s}_{t+1}, \tilde{s}_t) = 0. \tag{3.6}$$

The second, $t \in \{d, \cdots, T-1\}$, with delayed feedback:

$$g_t(\tilde{s}_{t+1}, \tilde{s}_t) = \begin{cases} \alpha_{u_{t-d}} & \text{if } s_{t+1} = s_t \cup \{u_{t-d}\} \text{ and } u_{t-d} \in U_t(s_t), \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

The third, $t = T$, to account for the expected reward of packets that are successfully received by the end of the frame but whose ACK is received after the end of the frame:

$$g_T(\tilde{s}_T) = \sum_{i \in U_t(s_t)} \alpha_i (1 - (1 - p_i)^{n_i}), \tag{3.8}$$

where n_i is the cardinality of i in the set $\{u_{T-1}, \cdots, u_{T-d}\}$ and $U_t(s_t) = \mathcal{N} \setminus s_t$.

3.3.4 Dynamic Program

Within a frame, the system evolves in steps and yields a reward which is additive over time, making it adequate for a DP formulation. For a given $\vec{\alpha}$, the problem of optimizing the EWST is defined as

$$EWST(\vec{\alpha}) := \max_{\eta \in \Pi} \sum_{i=1}^{N} \alpha_i \mathbb{E}\left[D_i^{\eta}(0)\right], \tag{3.9}$$

and solved by applying the cost-to-go function $J_t(\tilde{s}_t)$ recursively. Working backwards in time, we have

- for $t = T$:

$$J_T(\tilde{s}_T) = g_T(\tilde{s}_T = (s_T; u_{T-1}, \cdots, u_{T-d}));$$

- for $t \in \{d, \cdots, T-1\}$, the general equation:

$$J_t(\tilde{s}_t) = \max_{u_t \in U_t(s_t)} \mathbb{E}_{\tilde{s}_{t+1}}[g_t(\tilde{s}_{t+1}, \tilde{s}_t) + J_{t+1}(\tilde{s}_{t+1})],$$

  which, in the case $u_{t-d} \notin U_t(s_t)$, takes the form:

$$J_t(\tilde{s}_t) = \max_{u_t \in U_t(s_t)} [J_{t+1}(s_t; u_t, \cdots, u_{t-d+1})],$$

  and with $u_{t-d} \in U_t(s_t)$, the form:

$$J_t(\tilde{s}_t) = \max_{u_t \in U_t(s_t)} [\alpha_{u_{t-d}} p_{u_{t-d}} + p_{u_{t-d}} J_{t+1}(s_t \cup u_{t-d}; u_t, \cdots, u_{t-d+1}) + \\ +(1 - p_{u_{t-d}}) J_{t+1}(s_t; u_t, \cdots, u_{t-d+1})];$$

- lastly, for $t \in \{0, \cdots, d-1\}$:

$$J_t(\tilde{s}_t) = \max_{u_t \in U_t(s_t)} [J_{t+1}(s_{t+1} = \emptyset; u_t, \cdots, u_0)].$$

At each step t and for every possible $\tilde{s}_t$, the value of $J_t(\tilde{s}_t)$ is attained by choosing the optimal u_t^*. By keeping track of those choices, the optimal policy η^* is obtained. The output of the recursion at $t = 0$ is the optimal performance $J_0(\emptyset) = EWST(\vec{\alpha})$ associated with η^*.

So far, the mechanism for obtaining a policy that maximizes the network throughput for a given $\vec{\alpha}$ was derived. Nonetheless, as noted in Chapter 2, it is of particular interest to consider clients with minimum throughput requirements, q_i. In the next section, we describe an algorithm that satisfies any *feasible* requirement vector $(q_i)_{i=1}^N$.

3.4 Feasible Region and Optimal Algorithm

In this section, we discuss the Feasible Throughput Region and describe the Feasibility Optimal Algorithm called Frame-based Max-weight Policy. The concepts of Feasible Region and Optimal Policy follow definitions 2 and 3, respectively.

3.4.1 Feasible Throughput Region

For characterizing the Feasible Region, we sweep $\vec{\alpha}$ and iteratively solve the DP, collecting the optimal throughput vectors. It follows that the boundary of the Feasible Region is the convex hull of those vectors. In Figure 3-2 the Feasible Region for a fixed (T, N, p_i) is illustrated. Delays range from instantaneous feedback, $d = 0$, to no feedback, $d \geq T - 1$. The convex hull is depicted by the lines and the optimal throughput vectors by the circles, some of which are numbered for future reference.

Figure 3-2 shows a comparison between the Feasible Throughput Region of networks with different feedback delays. From this comparison, it is evident that the feedback mechanism influences the throughput performance of the network. Moreover, as discussed later in Section 3.5, the feedback mechanism also affects the optimal scheduling policy.

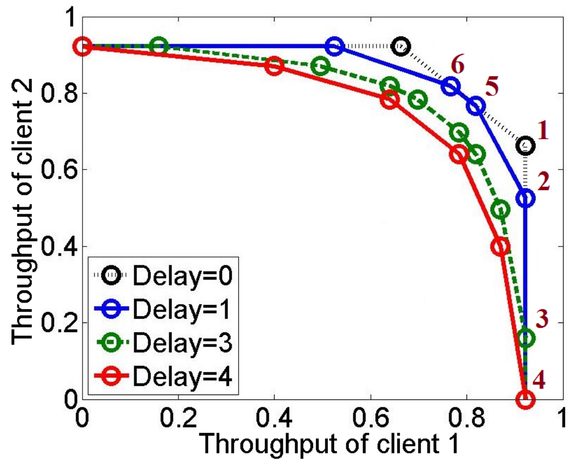

Figure 3-2: Feasible Throughput Region of a two-user wireless network with $T = 5$, $p_1 = p_2 = 0.4$ and different values of delay. The circles represent the throughput vectors attained by the optimal policies.

3.4.2 Optimal Algorithm

The Optimal Algorithm is a dynamic schedule that drives the throughput vector of the network to a point above the feasible requirement. To achieve this goal, we use the delivery debt, defined as the difference between the number of required deliveries and the current number of deliveries, as the weight vector ($\vec{\alpha}$) in the Dynamic Program. The output is the optimal policy η_k^* to be applied in the current frame. This describes the outline of one iteration of the Optimal Algorithm. The precise description follows.

FRAME-BASED MAX-WEIGHT ALGORITHM:

(i) At the beginning of the frame k, calculate for each client:

- the total number of packets delivered $Q_i(k)$;
- the delivery debt $d_i(k) = kq_i - Q_i(k)$.

(ii) Solve the Dynamic Program with $\alpha_i = \max\{d_i(k), 0\}$;

(iii) Employ η_k^* during frame k.

In each frame, the algorithm employs the optimal policy that gives higher weights to clients with higher debts. Simulation results are provided in Section 3.5.

Theorem 4. *The Frame-Based Max-Weight Algorithm is feasibility optimal.*

Proof. The proof is provided in Appendix A. □

A negative aspect of the Optimal Algorithm is that it can be computationally demanding, for it involves solving the Dynamic Program at every frame. One iteration of the DP entails maximizing over u_t, at each step t, for every possible $\tilde{s}_t = (s_t; u_{t-1}, \cdots, u_{t-d})$. Recalling that there are T slots in a frame, 2^N subsets s_t and N^d different sets $(u_{t-1}, \cdots, u_{t-d})$, this amounts to $\mathcal{O}(2^N N^d T)$ maximizations in a single iteration. A heuristic algorithm which is *suited for platforms with low computational power* and has a performance comparable to the optimal is presented next.

3.5 Insight into the Policies and Heuristic Algorithm

By solving the DP, the Optimal Algorithm randomizes over all possible η^*, covering the entire Feasible Region. The Heuristic Algorithm uses a predefined subset of policies and a randomization based on vector projections to cover a considerable portion of the Feasible Region. The first part of this section discusses the choice of the subset of policies and the second part describes the Heuristic Algorithm.

3.5.1 Subset of Policies

It was shown in Chapter 2, that the Greedy Policy is optimal for the case of instantaneous feedback, namely $d = 0$. The outcome of the DP extends this result, showing that, for some ranges of $\vec{\alpha}$, the Greedy Policy is also optimal for delayed feedback. In Figure 3-2, points 1 through 4 and their symmetric along the diagonal correspond to Greedy Policies.

For a given network setting (T, N, p_i, d), consider the subset containing $N!$ Greedy Policies, one for each possible priority sequence. By randomizing over those policies, it is possible to obtain the convex hull of the corresponding limiting points. Figure 3-3 shows the simulated throughput vectors of the Greedy Policies and the resulting Greedy Convex

Hull. As can be seen, the Greedy Convex Hull is significantly reduced in comparison to the Feasible Throughput Region.

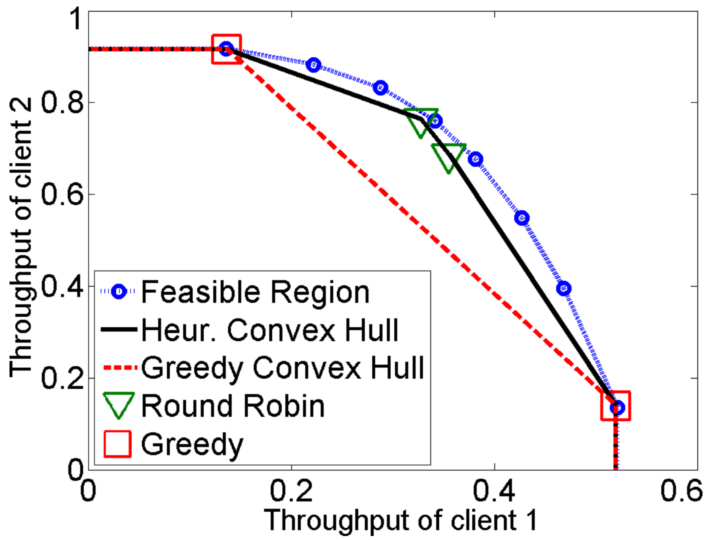

Figure 3-3: Comparison of the different convex hulls for $T=7$, $N=2$, $d=3$, $p_1=0.1$ and $p_2=0.3$. The throughput vectors of the Greedy and Round Robin Policies are obtained by simulating the network for 5×10^4 frames.

To expand the Greedy Convex Hull, we add to the subset a group of policies denoted here by Round Robin. Points 5 and 6 in Figure 3-2 are examples of such policies. Round Robin policies choose the next client to be scheduled (from the pool of allowed u_t) in a cyclic order. Considering $T=5$ and $N=3$, a sample Round Robin schedule would be $(1,2,3,1,2)$. Once the cyclic order is defined for the first frame, it is repeated in the following frames. Intuitively, the objective is to postpone the retransmission of packets with pending feedback, in order to avoid the state transition represented by (3.5), which is associated to zero reward. As illustrated in Figure 3-3, the simulated throughput vectors of the Round Robin policies expand the Greedy Convex Hull into the Heuristic Convex Hull. Notice that there are $N!$ Round Robin polices in the subset, one for each permutation of clients.

The Round Robin policy is not optimal for every network setting. However, as shown in Figure 3-3, even when it is not optimal, this policy can be used to expand the Greedy Convex Hull. Naturally, the shorter the distance between the throughput vector of the Round Robin policy and the border of the Feasible Region, the larger is the Heuristic Convex Hull.

To examine this distance in different network settings, we present next some simulation results.

Let the performance metric be $mean_i(\hat{q}_i^{\eta})$, which is the average throughput per client of policy η. Consider two policies: i) Round Robin with clients in order of increasing index; ii) Optimal Policy obtained from the DP for $\vec{\alpha} = (1)_{i=1}^{N}$. Figures 3-4 and 3-5 compare both policies in a variety of settings. Simulations run for 10^6 frames. The results show that the average throughput per client of the Round Robin policy is comparable to the Optimal in every simulation, what suggests that the performance of the Round Robin policy is in fact close to the border of the Feasible Region, specifically of the point which represents the performance of the Optimal Policy for $\vec{\alpha} = (1)_{i=1}^{N}$.

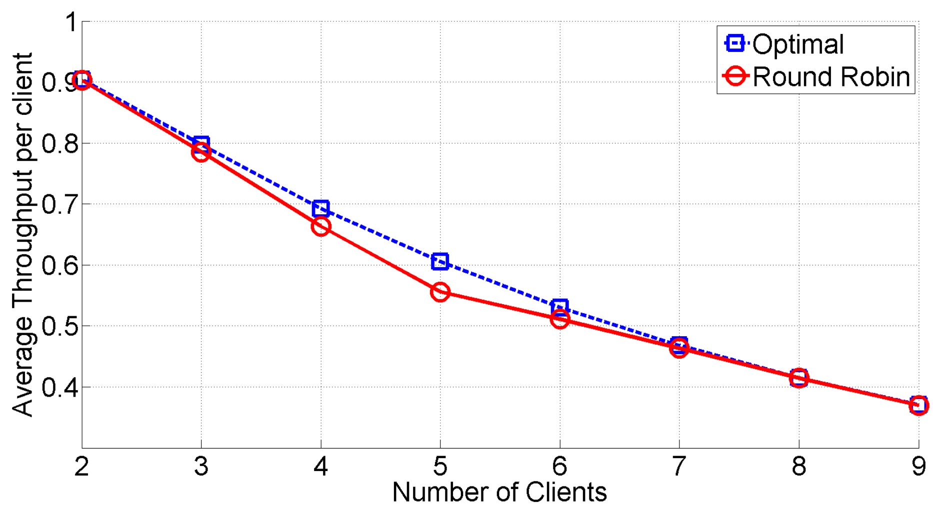

Figure 3-4: Simulation results of the Average Throughput per client in networks with $T = 10$, $d = 5$ and $p_i = 1/3, \forall i$. For each value of N, simulations run for 10^6 frames.

3.5.2 Heuristic Algorithm and Simulation Results

The goal of the Heuristic Algorithm is to fulfill the minimum delivery ratio requirements by selecting the best policy, within the subset, to be employed in each frame. At the beginning of the kth frame, let the average debt of client i be defined as $a_i(k) := q_i - Q_i(k)/k$, where $(q_i)_{i=1}^{N}$ is the requirement vector and $Q_i(k)$ is the total number of delivered packets. The vector of average debts $(a_i(k))_{i=1}^{N}$ represents the best return a policy could provide. From the same perspective, the throughput vectors, $(\hat{q}_i^{\eta})_{i=1}^{N}$, of each Greedy and Round Robin

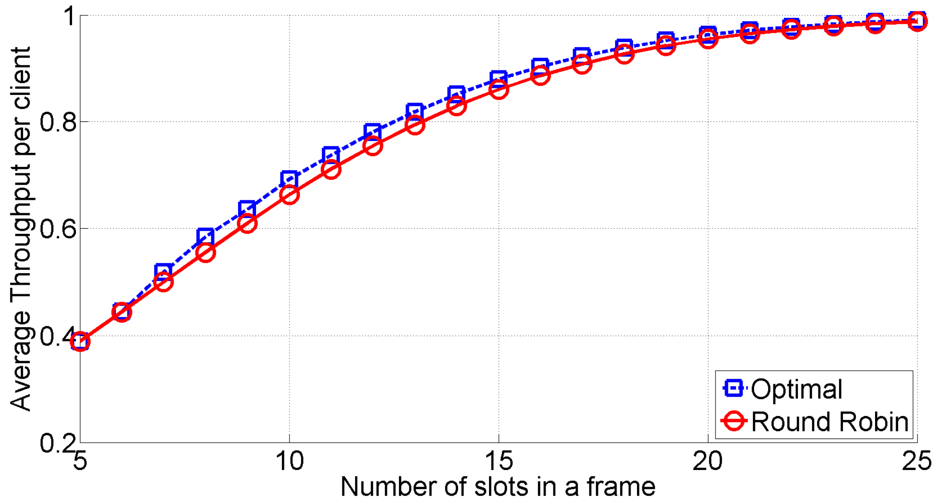

Figure 3-5: Simulation results of the Average Throughput per client in networks with $N = 4$, $d = 5$ and $p_i = 1/3, \forall i$. For each value of T, simulations run for 10^6 frames.

policies represent their expected returns. By projecting each $(\hat{q}_i^\eta)_{i=1}^N$ on $(a_i(k))_{i=1}^N$ and comparing the projection lengths, the algorithm can evaluate which policy is most suited for the kth frame. Figure 3-6 illustrates the projection of one throughput vector. A complete description of the Heuristic Algorithm follows.

HEURISTIC ALGORITHM

(i) At the beginning of the frame k, compute for each client:

- the total number of packets delivered $Q_i(k)$,
- the average debt $a_i(k) = q_i - Q_i(k)/k$;

(ii) For each policy η in the subset, project the vector $(\hat{q}_i^\eta)_{i=1}^N$ on $(a_i(k))_{i=1}^N$;

(iii) During the kth frame, employ the policy with longer projection length.

The Optimal and Heuristic Algorithms are studied via simulations. To examine their performances, the metrics of interest are the achieved throughput vector and the Deadlines

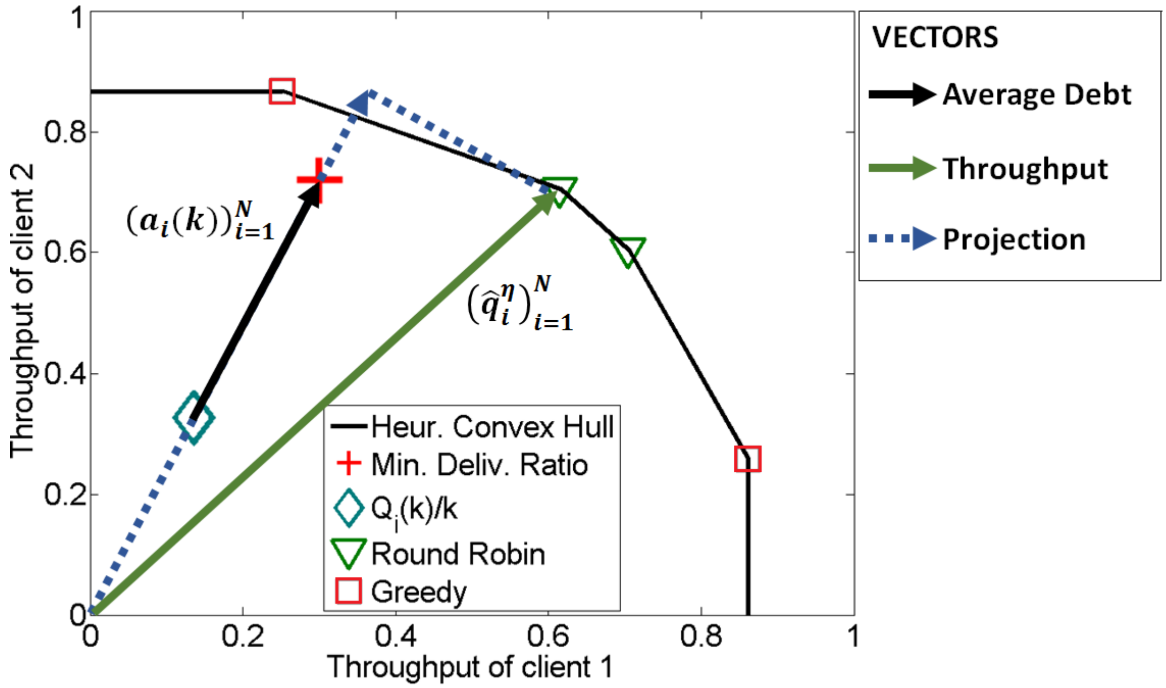

Figure 3-6: Projection of the throughput vector of one of the Round Robin policies on the vector of average debts.

Miss Ratio [12]. The DMR is defined as

$$\mathrm{DMR}(k) := \frac{1}{N}\sum_{i=1}^{N}\max\{d_i(k),0\}, \tag{3.10}$$

and represents the dynamic behavior of the algorithm regarding the delivery debt.

Simulation runs were provided for 5×10^4 frames in two distinct settings. Figures 3-7 and 3-8 consider the network with a requirement vector placed inside the Heuristic Convex Hull, while Figures 3-9 and 3-10 consider a requirement vector placed between the Heuristic Convex Hull and the Feasible Region. The simulated throughput vectors in Figure 3-7 shows that both algorithms can fulfill any constraints that lie inside the Heuristic Convex Hull. Moreover, the DMR in Figure 3-8 shows that the Optimal and Heuristic Algorithms drive the throughput vector to the minimum delivery ratio within 200 frames even when this requirement is close to the edge of the Heuristic Convex Hull. In contrast, when the requirement vector is outside the Heuristic Convex Hull but inside the Feasible Region, we see from Figures 3-9 and 3-10 that the Heuristic Algorithm fails to fulfill the delivery requirement but the Optimal Algorithm obtains the desired delivery ratio within a few hundred frames.

3.6 Further Work

This chapter studied the problem of scheduling packets with hard deadlines and delivery ratio constraints in a wireless network with delayed feedback. We extended the analytical model of [12] to account for delayed feedback and used Dynamic Programming to solve for the optimal policy. This solution was the groundwork for the Optimal and Heuristic Algorithms. The Optimal Algorithm was shown to be feasibility optimal and both were evaluated through simulations. Results suggested that the low-complexity Heuristic Algorithm can fulfill any delivery ratio requirements that lie inside the convex hull formed by its subset of policies.

Some interesting extensions of this work include consideration of multicast traffic and time-varying channels. Multicast traffic could be approached with the simplifications proposed in [19], in which the system model can be adapted, without loss of generality, to consider each client subscribing to a single flow. Time-varying channels could be considered using the framework proposed in [14].

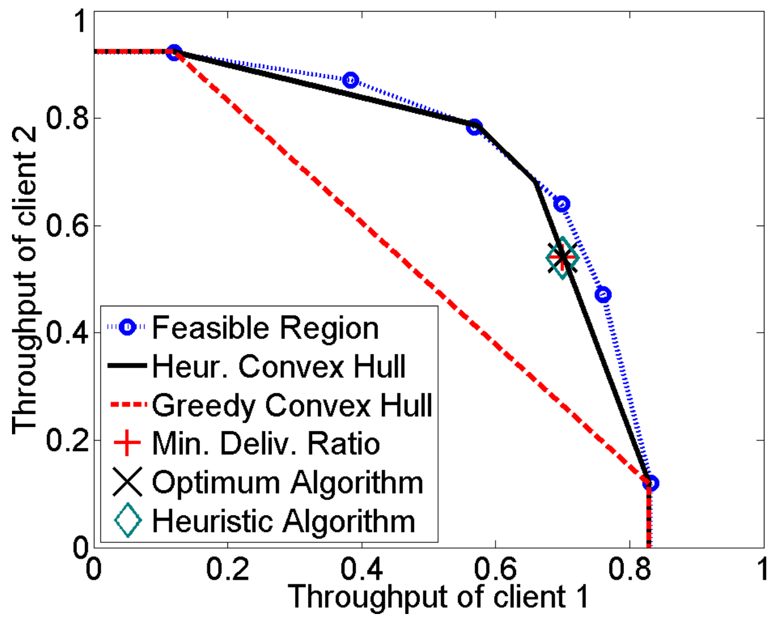

Figure 3-7: Comparison of the throughput vectors attained by the Optimal and Heuristic Algorithms in a two-user network with $T = 5$, $d = 3$, $p_1 = 0.3$, $p_2 = 0.4$, $q_1 = 0.7$ and $q_2 = 0.54$. Both algorithms fulfill the minimum delivery ratio requirement.

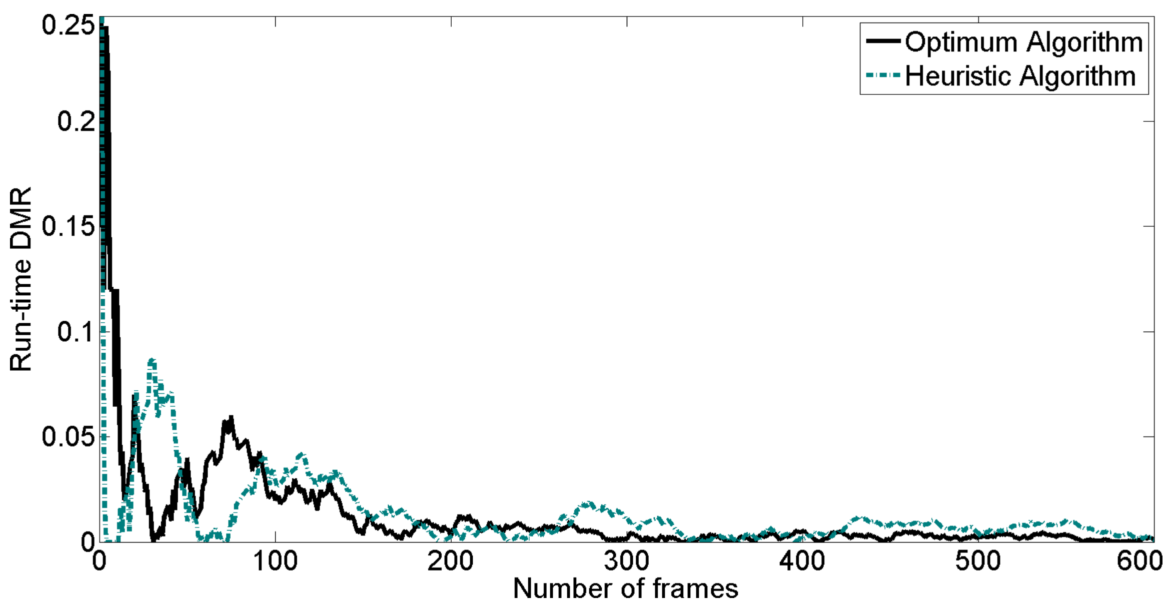

Figure 3-8: Dynamic behavior of the simulation in Figure 3-7 for the first 600 frames.

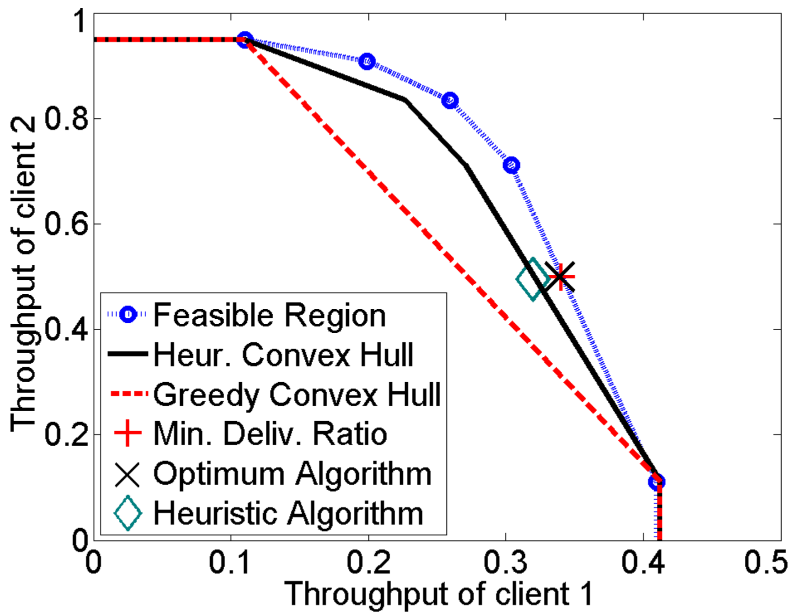

Figure 3-9: Comparison of the throughput vectors attained by the Optimal and Heuristic Algorithms in a two-user network with $T = 5$, $d = 2$, $p_1 = 0.1$, $p_2 = 0.45$, $q_1 = 0.34$ and $q_2 = 0.5$. The Optimal Algorithm fulfills the minimum requirement, while the Heuristic Algorithm fails.

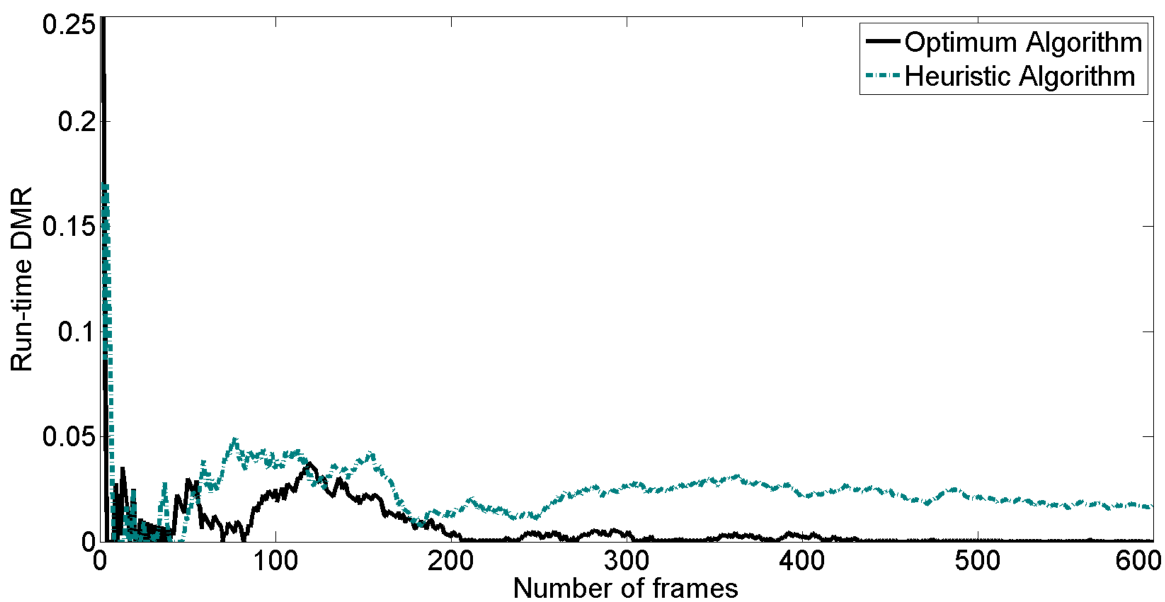

Figure 3-10: Dynamic behavior of the simulation in Figure 3-9 for the first 600 frames.

Chapter 4

Heterogeneous Deadlines

Until now, we have considered networks in which all packets have the same deadline T. The problem of scheduling packets with heterogeneous deadlines in **wired** networks has been extensively studied in the literature. A well-known scheduling strategy is the Shortest Time to Extinction (STE), which is also called Earliest Deadline First or Earliest Due Date. The STE policy is as follows: in each slot, forward the packet that is closest to its deadline. This policy is known for maximizing the throughput of networks with arbitrary packet arrivals and arbitrary deadlines (announced upon arrival). The optimality of the STE policy is derived in [4] for a single-link network and in [5, 21] for a tree network with a single destination.

Wireless single-hop networks with heterogeneous deadlines are analyzed in [14, 25]. In [25], the authors point out that, in general, the STE policy yields suboptimal throughput performance in wireless networks with arbitrary arrivals and arbitrary deadlines, even when the BS has perfect knowledge of the state of the channels and packets are transmitted only in channels which are *ON*. Assuming perfect knowledge of the channel, [25] establishes that the STE policy is optimal when deadlines are homogeneous. In [14], a throughput-optimal policy is derived for a wireless network employing rate adaptation, in which case the BS transmits error-free packets with higher service times. For the case of networks not employing rate adaptation, [14] proposes an interesting heuristic called Adaptive-Allocation Policy.

In this chapter, we assess a wireless single-hop network with heterogeneous deadlines and multiple packet flows per client. It is assumed that the BS has *no knowledge* of the channel state prior to transmitting, other than the value of p_i. Using a stochastic dominance argument [26], the STE policy is shown to be throughput-optimal under the assumption of symmetric channels, i.e. all clients having the same channel reliability $p_i = p$.

The remainder of the chapter is outlined as follows. In Section 4.1, the network model and traffic assumptions are introduced. Section 4.2 formulates the throughput optimization problem using Dynamic Programming. The optimal scheduling policies, including STE and Greedy, are discussed in Section 4.3. Section 4.4 concludes this chapter with some final comments.

4.1 Network Model

Consider a wireless single-hop network with a BS sending unicast packets to N^* clients. Let $i \in \{1, \cdots, N^*\}$ be the client index and F_i be the total number of flows to client i. Flows are uniquely identified by the tuple (i, f), where $f \in \{1, 2, \cdots, F_i\}$ is the flow index. Time is slotted and $T_{i,f}$ consecutive slots form a *frame of flow* (i, f). Figure 4-1 illustrates the frames of two different flows to client 1.

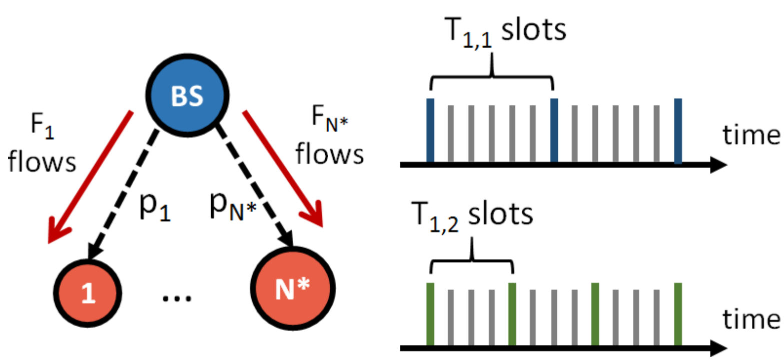

Figure 4-1: Network model and frame structure. On the right, we have the frame structure of two flows to client 1, one with $T_{1,1} = 6$ slots and the other with $T_{1,2} = 4$ slots.

At the beginning of every frame of flow (i, f), a single packet is generated at the BS. Packets of flow (i, f) which are not delivered to client i by the end of the frame, i.e. after $T_{i,f}$ slots, are discarded. Notice that, at any time slot, each flow has at most one packet available for transmission. The kth frame of flow (i,f) comprises slots $t \in \{kT_{i,f}, \cdots, (k+1)T_{i,f}-1\}$, where $t,k \in \mathbb{Z}^+$ are the slot index and the frame index, respectively.

In a slot, the BS selects a flow and transmits its packet over the wireless channel. The outcome of the transmission of packet (i,f) can be either a successful delivery, with probability $p_i \in (0,1]$, or a transmission error, with probability $1-p_i$. Naturally, the channel reliability depends only on the client i and not on the specific flow f. After each transmission, the client sends an instantaneous feedback, ACK or NACK, to the BS through an error-free control channel.

For any given set of finite frame lengths, $T_{i,f}$, the network model is stochastically renewed in intervals of T_{lcm} slots, where T_{lcm} is the least common multiple of $T_{i,f}, \forall(i,f)$. Denote the renewal intervals of T_{lcm} consecutive time slots as *super-frames*. Analogously to Chapter 2, it is sufficient to regard the network over the first super-frame only for the purpose of analyzing throughput.

Without loss of generality, this network model can be further simplified to the case of one flow per client. This is done by substituting each client i by F_i *pseudo-clients* with a single flow. Pseudo-clients associated with client i have channel reliability p_i and flows preserve their original frame length $T_{i,f}$, as illustrated in Figure 4-2. When transitioning to the Single Flow Model, the pseudo-client index is (re)defined as $i \in \{1, \cdots, N\}$, where $N = \sum_{i=1}^{N^*} F_i$, and the frame lengths are renamed T_i. Notice that for the new model, the index i unambiguously identify the flows. Henceforth in this chapter, the *Single Flow Model* is assumed and the terms pseudo-client and client are used interchangeably.

The performance metric of interest for this network is the Expected Weighted Sum Throughput (EWST). This metric is optimized over the class of non-anticipative and work conserving policies, denoted Π. Let $\vec{\alpha} = (\alpha_i)_{i=1}^N$ be the positive vector of client weights and $D_i^\eta \in \{0,1,2,\cdots,T_{lcm}/T_i\}$ be the random variable that represents the number of packets delivered to client i during the *first super-frame* when the BS employs policy $\eta \in \Pi$. Then, for a given vector $\vec{\alpha}$, the optimization of the EWST over the first super-frame is expressed as

$$EWST(\vec{\alpha}) := \max_{\eta \in \Pi} \sum_{i=1}^{N} \alpha_i \mathbb{E}\left[D_i^\eta\right]. \tag{4.1}$$

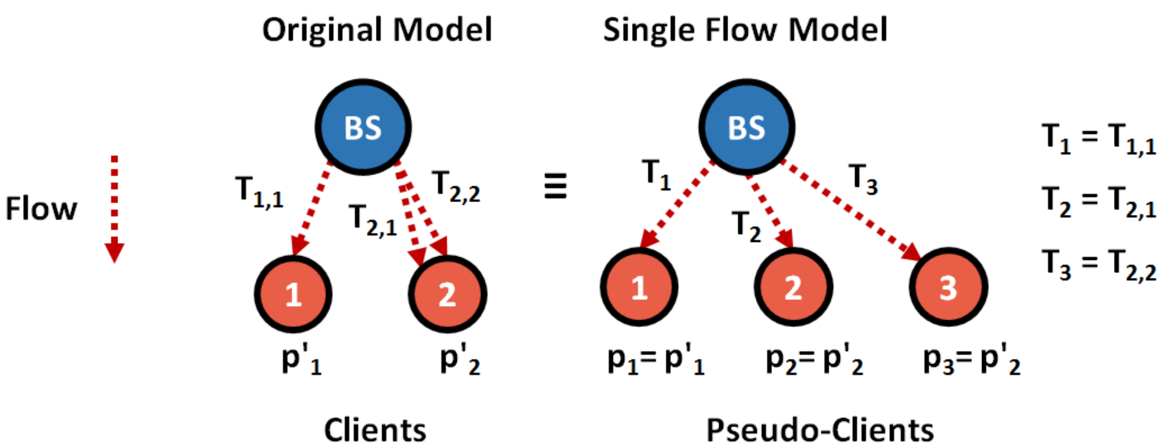

Figure 4-2: Transition from the Original Model to the Single Flow Model. The original model has $N^* = 2$ clients. One client has $F_1 = 1$ flow and the other has $F_2 = 2$ flows. The equivalent simplified model has $N = 3$ clients with a single flow each. Notice that the channel reliabilities and the values of the frame lengths are preserved.

In previous chapters, the throughput and the delivery-ratio were identical metrics, what is not the case for networks with heterogeneous deadlines. The relationship between the expected delivery ratio of client i, $\hat{q}_i^\eta$, and the expected throughput of client i is

$$\hat{q}_i^\eta = \frac{T_i \, \mathbb{E}[D_i^\eta]}{T_{lcm}} \, , \forall i.$$

With the network model described and the problem formulated, in the next section we address the problem using Dynamic Programming.

4.2 Problem Formulation

In this section, the network model is formulated as a Dynamic Program. The objective function considered is the EWST in (4.1). The components of the cost-to-go function used for solving the DP are described next.

4.2.1 State, Control, Transition and Reward

The frame structure of each flow determines the times in which undelivered packets are dropped and new packets arrive. For representing those times in a systematic way, suitable for the DP formulation, the sequence a_t is defined for the first super-frame. Let a_t represent the set of flows that have an arrival at the beginning of slot t. It follows that $i \in a_t, \forall t \in \{0, T_i, 2T_i, \cdots, T_{lcm} - T_i\}$ for every client. Notice that the sequence a_t completely characterizes the frame structure of all flows in the network, and vice versa.

Consider the set of clients $\mathscr{N} = \{1, \cdots, N\}$ and the state space $S = 2^{\mathscr{N}}$. Let the set $s_t \in S, t \in \{0, \cdots, T_{lcm} - 1\}$, represent the clients that have successfully received their packet by the beginning of slot t. Recall that, at any time slot, each client has at most one packet awaiting transmission. The set s_t is the state of the network, since it provides a complete characterization of the system.

Let u_t be the scheduling decision in time slot t. The set of clients (or flows) which are available for transmission during slot t is denoted $U_t(s_t) = \mathscr{N} \backslash s_t$. Therefore, naturally, $u_t \in U_t(s_t)$, $\forall t$. For guaranteeing work conservation, it is imposed that the BS only idles when all packets have been acknowledged, i.e. $U_t(s_t = \mathscr{N}) = \emptyset$.

State transitions are determined by the scheduling decision, u_t, by the feedback signal received at the end of the slot and by the arrivals at the next slot, a_{t+1}. If the BS idles during slot t, then the state in the next slot is $s_{t+1} = \{s_t \backslash a_{t+1}\}$.

$$P(s_{t+1} = \{s_t \backslash a_{t+1}\} | s_t, u_t = \emptyset) = 1 \; . \tag{4.2}$$

On the other hand, if the BS transmits the packet from flow i, then the state at the beginning of slot $t+1$ will depend upon the outcome of the transmission and the arrivals at slot $t+1$, as illustrated in Figure 4-3. The transition from slot t to slot $t+1$ is characterized by the probabilities

$$P(s_{t+1} = \{s_t \backslash a_{t+1}\} | s_t, u_t = i) = 1 - p_i \; ; \qquad \text{[NACK is received]} \tag{4.3}$$

$$P(s_{t+1} = \{[s_t \cup i] \backslash a_{t+1}\} | s_t, u_t = i) = p_i \; . \qquad \text{[ACK is received]} \tag{4.4}$$

The last concept to be introduced prior to the actual DP formulation is the reward function. The reward is directly related to the EWST and is associated with the successful delivery of a packet, i.e. with transition (4.4). When an ACK from client i is received, the

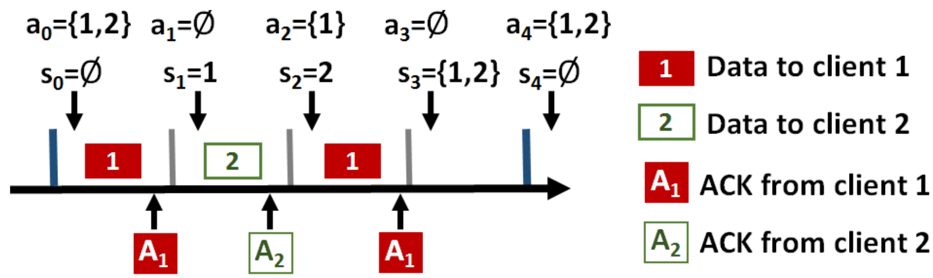

Figure 4-3: Super-frame of a network with $N = 2$, $T_1 = 2$, $T_2 = 4$. It follows from the values of the frame lengths that $a_0 = \{1,2\}$, $a_2 = \{1\}$ and $a_1 = a_3 = \emptyset$. The scheduling decisions and transmission feedback determine the sample state transitions displayed.

network collects a reward of $alpha_i$. Thus, the reward function is defined as

$$g_t(s_{t+1}, s_t) = \begin{cases} \alpha_i & \text{if an ACK is received and } s_{t+1} = [s_t \cup i] \backslash a_{t+1} \text{ ;} \\ 0 & \text{otherwise .} \end{cases} \tag{4.5}$$

The system gets a reward of α_i when a packet is successfully delivered to client i and zero otherwise. Recalling that D_i^η represents the number of packets delivered to client i in the first super-frame, the following equivalence holds

$$\mathbb{E}\left[\sum_{i=1}^{N} \alpha_i D_i^\eta\right] = \mathbb{E}\left[\sum_{t=0}^{T_{lcm}-1} g_t(s_{t+1}, s_t)\right]. \tag{4.6}$$

4.2.2 Dynamic Program

Applying (4.6) to the definition of EWST in (4.1) yields

$$EWST(\vec{\alpha}) := \max_{\eta \in \Pi} \sum_{i=1}^{N} \mathbb{E}\left[\alpha_i D_i^\eta\right] = \max_{\eta \in \Pi} \sum_{t=0}^{T_{lcm}-1} \mathbb{E}\left[g_t(s_{t+1}, s_t)\right]. \tag{4.7}$$

Within a super-frame, the system evolves in steps and gives a reward which is additive over time, making it adequate for a DP formulation. For a given $\vec{\alpha}$, the problem of optimizing the EWST is defined in (4.7) and solved by applying the cost-to-go function $J_t(s_t)$ iteratively, backwards in time. The initial value of the cost-to-go, specifically for $t = T_{lcm}$, is given by

$J_{T_{lcm}}(s_{T_{lcm}}) = 0, \forall s_{T_{lcm}} \in S$, and the recursion is expressed as

$$\begin{aligned} J_t(s_t) &:= \max_{u_t \in U_t(s_t)} \mathbb{E}_{s_{t+1}}[g_t(s_{t+1}, s_t) + J_{t+1}(s_{t+1})] \\ &= \max_{i \in U_t(s_t)} [\alpha_i p_i + p_i J_{t+1}([s_t \cup i] \backslash a_{t+1}) + (1-p_i) J_{t+1}(s_t \backslash a_{t+1})]. \end{aligned} \tag{4.8}$$

At each step t and for every possible s_t, the value of $J_t(s_t)$ is attained by choosing the control u_t^* which maximizes (4.8). By keeping track of the u_t^* chosen for every (t, s_t) pair, the optimal policy η^* is obtained. The output of the recursion at $t = 0$ is the optimal performance $J_0(\emptyset) = EWST(\vec{\alpha})$ associated with η^*.

With this, the description of the Dynamic Programming formulation is complete. In the next section, we provide some insight into the optimal policy η^*.

4.3 Feasible Region and Optimal Scheduling Policies

In this section, we characterize the Feasible Region of the network and discuss the Optimal Scheduling Policies, including the Greedy and the Shortest Time to Extinction. The concept of Feasible Region follows from definition 2. In Theorem 5, it is shown that under the assumption of clients having the same weight, $\alpha_i = \alpha$, and channel reliability, $p_i = p$, the STE policy minimizes the expected number of packet losses due to the deadlines.

4.3.1 Feasible Throughput Region

For a fixed $\vec{\alpha}$, the DP solution to (4.7) yields an optimal policy η^* and the associated maximum throughput vector, $(\mathbb{E}[D_i^*])_{i=1}^N$. Therefore, by sweeping $\vec{\alpha}$ and iteratively solving the DP, the optimal throughput vectors are collected. The Feasible Region is the convex hull of those vectors and it represents the maximum throughput that can be achieved by any admissible policy. In Figure 4-4, the Feasible Region for a given network setup $(T_i; N; p_i)$ is illustrated. The convex hull is depicted by the dotted lines and the optimal throughput vectors by the blue circles. The throughput vectors are numbered for future reference.

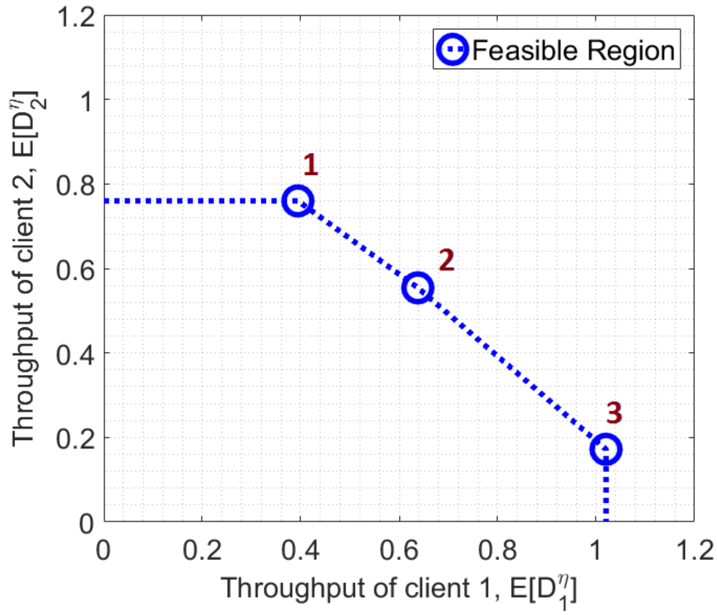

Figure 4-4: Feasible Throughput Region of a wireless network with $N = 2$, $T_1 = 2$, $T_2 = 4$, $p_1 = p_2 = 0.3$. The circles represent the throughput vectors attained by the optimal policies.

4.3.2 Optimal Scheduling Policies

It was shown in Chapter 2, that the Greedy Policy is optimal for the case of homogeneous deadlines, $T_i = T, \forall i \in \mathcal{N}$. The outcome of the DP extends this result, showing that, for some ranges of $\vec{\alpha}$, the Greedy Policy is also optimal for heterogeneous deadlines. In Figure 4-4, points 1 and 3 correspond to Greedy Policies. Point 1 is associated to the policy that prioritizes client 2 over client 1 and point 3 has the opposite priority order.

Point 2 in Figure 4-4 is associated with the Shortest Time to Extinction policy. In every slot, the STE policy selects, from the pool of available flows, the packet which is closest to its deadline. For illustrating the scheduling decisions of the BS when employing the STE policy, consider a network with $N = 2$, $T_1 = 2$ and $T_2 = 3$, and assume that both flows are available for transmission *in every slot*, namely $s_t = \emptyset, \forall t$. Then, the STE policy is as illustrated in Figure 4-5[1]. Notice that if $s_t \neq \emptyset$ for any particular slot t, then the choice of the BS in that slot is trivial: transmit the sole available client or idle if there is no available

[1] In Figure 4-5, the STE policy could have scheduled either flow in the last couple of slots, since the deadline of both flows coincide. In those cases, it can be imposed that ties are broken in favor to the client with highest (or lowest) index i.

client. Intuitively, the STE policy maximizes the throughput of the network by prioritizing the transmission of the packet which is closest to being dropped.

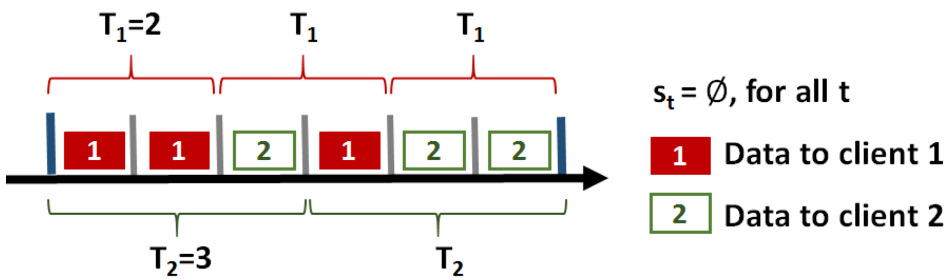

Figure 4-5: Super-frame of a network with $N = 2$, $T_1 = 2$ and $T_2 = 3$. It is assumed that $s_t = \emptyset, \forall t$. These scheduling decisions are repeated in every successive super-frame.

The STE policy is not optimal in every network setting. Despite the fact that the time-to-extinction plays an important role in maximizing the throughput of the network, other factors such as channel reliability and client weight also influence scheduling decisions. In Theorem 5, the STE policy is shown to be throughput-optimal for the class of symmetric wireless networks.

Theorem 5. *Consider the network model in Section 4.1 with arbitrary frame lengths T_i. Assume that all clients have the same channel reliability $p_i = p \in (0,1]$ and weight $\alpha_i = \alpha$. Then, the STE policy optimizes the objective function, EWST, in (4.7).*

Proof. Theorem 5 assumes that $p_i = p$ and $\alpha_i = \alpha$ for all clients. Thus, the objective function in (4.7) can be simplified

$$EWST(\vec{\alpha}) = \alpha \max_{\eta \in \Pi} \mathbb{E}\left[\sum_{i=1}^{N} D_i^{\eta}\right]. \tag{4.9}$$

Under this symmetry assumptions, maximizing the EWST is equivalent to minimizing the expected number of packets discarded by the BS due to the deadlines. Let $L^{\eta}(t)$ be the total number of packet losses until (but not including) time slot t when policy η is employed. Then, it follows that:

$$\eta^* = \arg\max_{\eta \in \Pi} \mathbb{E}\left[\sum_{i=1}^{N} D_i^{\eta}\right] = \arg\min_{\eta \in \Pi} \mathbb{E}\left[L^{\eta}(T_{lcm})\right]. \tag{4.10}$$

And for showing that the STE policy satisfies (4.10), it is sufficient to verify that

$$\mathbb{E}[L^{STE}(T_{lcm})] \leq \mathbb{E}[L^{\eta}(T_{lcm})], \forall \eta \in \Pi \,. \tag{4.11}$$

Prior to verifying condition (4.11), we introduce the concept of stochastic dominance [26]. Denote the stochastic process associated with the sequence $\{L^{\eta}(t)\}_{t=0}^{T_{lcm}}$ as L^{η} and its sample path as l^{η}. Let $\mathbb{D}$ be the space of all sample paths l^{η}. Define $\mathscr{F}$ as the set of measurable functions $f : \mathbb{D} \to \mathbb{R}^{+}$ such that $f(l^{STE}) \leq f(l^{\eta})$ for every $l^{STE}, l^{\eta} \in \mathbb{D}$ which satisfy $l^{STE}(t) \leq l^{\eta}(t), \forall t$. Then, we can define stochastic dominance.

Definition 6. *Stochastic Dominance. We say that L^{STE} is stochastically smaller than L^{η} and write $L^{STE} \leq_{st} L^{\eta}$ if $P\{f(L^{STE}) > z\} \leq P\{f(L^{\eta}) > z\}, \forall z \in \mathbb{R}, \forall f \in \mathscr{F}$.*

Since $f(L^{\eta})$ is positive valued, $L^{STE} \leq_{st} L^{\eta}$ implies[2] $\mathbb{E}[f(L^{STE})] \leq \mathbb{E}[f(L^{\eta})], \forall f \in \mathscr{F}$. Knowing that one function that satisfies the conditions in $\mathscr{F}$ is $f(L^{\eta}) = L^{\eta}(T_{lcm})$, it follows that if $L^{STE} \leq_{st} L^{\eta}, \forall \eta \in \Pi$, then $\mathbb{E}[L^{STE}(T_{lcm})] \leq \mathbb{E}[L^{\eta}(T_{lcm})], \forall \eta \in \Pi$, which is our target condition in (4.11). Therefore, it follows that for establishing the optimality of the STE policy, it is sufficient to confirm that L^{STE} is stochastically smaller than $L^{\eta}, \forall \eta$.

Stochastic dominance can be demonstrated using the definition. Nonetheless, in this chapter, and in works such as [4, 8, 24, 26], an alternative and often simpler method known as stochastic coupling is utilized. Let $\eta \in \Pi$ be an *arbitrary* admissible policy. Then, for verifying that $L^{STE} \leq_{st} L^{\eta}$, we show that there exist two stochastic processes $\widetilde{L}^{STE}$ and $\widetilde{L}^{\eta}$ such that:

(i) L^{STE} and $\widetilde{L}^{STE}$ have the same probability distribution;

(ii) L^{η} and $\widetilde{L}^{\eta}$ have the same probability distribution;

(iii) $\widetilde{L}^{STE}$ and $\widetilde{L}^{\eta}$ are on a common probability space;

(iv) $\widetilde{L}^{STE}(t) \leq \widetilde{L}^{\eta}(t)$, with probability 1, $\forall t$.

Prior to discussing these four conditions, we formally introduce the channel state and the stochastic coupling method. Let $E_i(t) \sim Ber(p)$ be the random variable that represents

[2] Recall that for any positive valued X, it follows that $\mathbb{E}[X] = \int_{x=0}^{\infty}(1 - P\{X \leq x\})dx = \int_{x=0}^{\infty} P\{X > x\}dx$.

the channel state of client i during slot t

$$E_i(t) = \begin{cases} 1, & \text{with probability } p \quad \text{[Channel is } ON\text{]} \, ; \\ 0, & \text{with probability } 1-p \quad \text{[Channel is } OFF\text{]} \, . \end{cases} \tag{4.12}$$

The channel state of each client is independent of the channel state of other clients and of scheduling decisions. Recall that the BS has no knowledge of the channel state before transmitting the packet.

Stochastic coupling is a method utilized for comparing stochastic processes by imposing a common underlying probability space. Let the process $\widetilde{L}^{\eta}$ be identical to L^{η}, then, the probability space of $\widetilde{L}^{\eta}$ is associated to the outcome of the transmission in each slot. Thus, it only depends on scheduling decisions and channel states. Now, let us construct $\widetilde{L}^{STE}$ on the same probability space as the process $\widetilde{L}^{\eta}$. For coupling these two stochastic processes, we dynamically connect the channel states of the STE policy to the channel states of η as follows. Suppose that in slot t, the STE policy schedules client i while policy η schedules client j, then, for the duration of that slot, we assign $E_i(t) \leftarrow E_j(t)$ under STE. This connection imposes that, at every slot, the outcome of both transmissions are identical, regardless of client selected by the STE policy. This is only possible because $E_i(t)$ is identically distributed for all clients. Then, the processes $\widetilde{L}^{STE}$ and $\widetilde{L}^{\eta}$ are said to be stochastically coupled.

Returning to our four conditions, it follows from the coupling procedure described above that conditions (i), (ii) and (iii) are satisfied by the method of constructing $\widetilde{L}^{\eta}$ and $\widetilde{L}^{STE}$. Therefore, the only condition that remains to be shown is

$$\widetilde{L}^{STE}(t) \leq \widetilde{L}^{\eta}(t), \text{with probability } 1, \forall t. \tag{4.13}$$

Since the random variables $\widetilde{L}^{STE}(t)$ and $\widetilde{L}^{\eta}(t)$ are measures of the total number of packet losses until time slot t, our network model can be viewed as in Figure 4-6. The network of N single flow clients is equivalent to the network of a single client and N parallel flows, because $\widetilde{L}^{STE}(t)$ and $\widetilde{L}^{\eta}(t)$ are not affected by the destination of the packets.

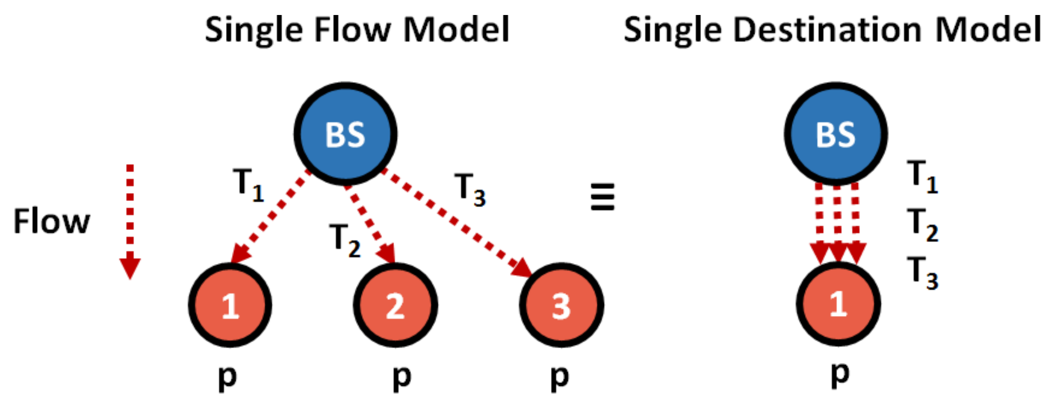

Figure 4-6: Equivalence between the Single Flow Model and the Single Destination Model. The three flows from the Single Flow Model run in parallel in the Single Destination Model with their original frame lengths.

It was established in [5, Theorem 1] that for a **wired** tree network with a single destination and arbitrary arrival and deadline processes, the STE policy minimizes the total number of packet losses due to the deadlines. The single destination model in Figure 4-6 is a particular case of a tree network, thus, the result in [5, Theorem 1] applies. Accordingly, in the context of our model, when the channel reliability is $p = 1$, it follows that the condition $\widetilde{L}^{STE}(t) \leq \widetilde{L}^{\eta}(t), \forall t$ is fulfilled. Notice that for a channel with $p = 1$, the quantities $\widetilde{L}^{STE}(t)$ and $\widetilde{L}^{\eta}(t)$ are deterministic.

For showing that the condition $\widetilde{L}^{STE}(t) \leq \widetilde{L}^{\eta}(t)$, with probability 1, $\forall t$, holds for any $p \in (0, 1]$, consider the system evolving under policies STE and η. Couple the realization of the channel states as described, giving to the STE policy the same transmission outcome of the packet selected by η. Now, suppose that the STE policy *knows the sequence of channel states a priori*. From the perspective of this anticipatory policy, the system is deterministic. It follows from [5, Theorem 1] that employing the anticipatory STE policy to the slots in which the channel is *ON* minimizes the number of packet losses for any sample sequence of channel states. Since scheduling decisions in slots which are *OFF* do not affect the system, we can employ the anticipatory STE policy to all slots at no cost.

The knowledge of the channel states does not provide the STE policy with an incentive for changing the client selection even when it is known that the transmission will fail. This is because the channel state of the STE policy is coupled to the channel state of η.

Thus, changing the client selected by the STE policy would not alter the outcome of the transmission.

Moreover, the knowledge of the channel states does not affect the sequence of packets selected by the STE policy. In slot t, the STE policy selects, from the pool of available flows, the packet which is closest to its deadline. The non-anticipatory STE policy is identical to the anticipatory STE policy. Therefore, the non-anticipatory STE policy also minimizes the total number of packet losses for any sample sequence of channel states. Finally, we conclude that $\widetilde{L}^{STE}(t) \leq \widetilde{L}^{\eta}(t)$, with probability 1, $\forall t$, holds for $p \in (0,1]$, and the proof is complete.

□

4.4 Remarks

This chapter studied the problem of scheduling multiple flows of packets with heterogeneous deadlines in a wireless single-hop network. We formulated the problem using Dynamic Programming and obtained the scheduling policy that maximizes the Expected Weighted Sum Throughput. Moreover, the STE policy was shown to be throughput-optimal under the symmetry assumptions in Theorem 5. This optimality result showed that the three aspects that affect the optimal scheduling decisions are the time-to-extinction, the client weights and the channel reliability.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Age of Information

The Age of Information (AoI) is a recently proposed metric that has been receiving a lot of attention [1, 6, 10, 15, 16]. It represents the freshness of the information at the destination, thus, being suited for applications with time-sensitive data. An interesting feature of the AoI is that it takes the perspective of the client in measuring how fresh is the information, as opposed to the well-known packet delay, that is simply the time elapsed from the packet‘s generation to its delivery.

For illustrating the Age of Information, consider the communication system in Figure 5-1, with a BS sending packets to a single client. In this system, packets generated at the BS wait in line before being served. The server transmits packets in order of arrival, i.e. using a FCFS discipline. The evolution of the AoI is as follows. It increases linearly in time while there is no packet delivery and, when a delivery occurs, the AoI takes the value of the packet delay. In Figure 5-1, the evolution of the AoI associated with a sample sequence of arrivals and deliveries is shown.

As can be seen from Figure 5-1, the AoI is influenced by the packet delay and also by the packet delivery process. Naturally, in general, controlling only the packet delay is insufficient for achieving a good AoI performance. Suppose that the system in Figure 5-1 has a server with constant service time and a BS that rarely generates new packets. In this case, the queue would be often empty, leading to low packet delay. Even so, the AoI performance would be poor, as a consequence of the lack of packet deliveries. For obtaining fresh data,

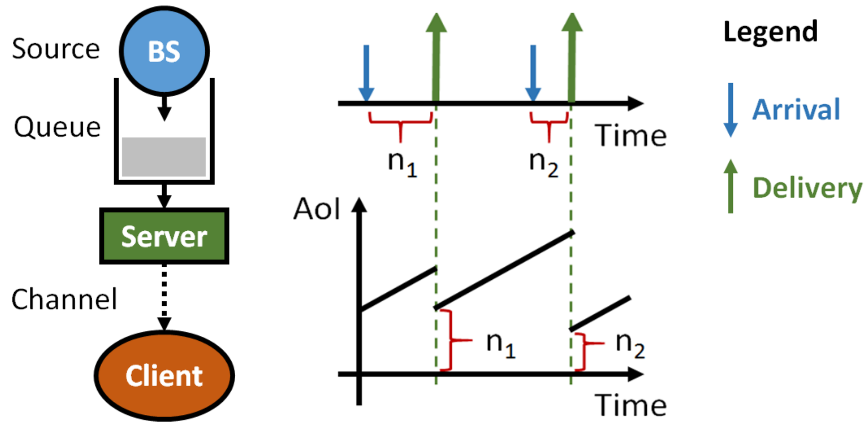

Figure 5-1: On the left, we illustrate the communication system and, on the right, the evolution of the AoI. The variable n_k is the delay of the kth packet, i.e. the time elapsed from its generation to its delivery.

the communication system needs *packets with low delay being regularly delivered to the clients*.

In this chapter, our goal is to find a transmission scheduling policy that minimizes the Age of Information of the wireless single-hop network with N clients introduced in Chapter 2. This network model impacts the AoI formulation in two distinct ways. The hard-deadline constraint imposes an upper-bound on the packet delays, thus reducing its influence on AoI. In addition, the periodic generation of fresh packets at the BS allows the scheduling policy to deliver packets to the clients regularly. It is important to emphasize the difference between service regularity and minimum throughput. A minimum throughput requirement can be fulfilled even if long periods of starvation occur, as long as those are balanced by periods of consecutive deliveries.

The remainder of this chapter is outlined as follows. In the next section, the system model and the Age of Information formulation are introduced. In Section 5.2, the problem of minimizing the AoI is solved using Dynamic Programming. In Section 5.3, we compare the performances of AoI-optimal scheduling policies with the throughput-optimal policies discussed in Chapter 2. Our main conclusion is that AoI-optimal policies are also throughput-optimal, while the converse is not true. The chapter is concluded and further work is discussed in Section 5.4.

5.1 Network Model

Consider the wireless network presented in Chapter 2, in which a BS is sending packets with hard-deadlines to N clients. Time is slotted and frames are composed of T successive time slots. The BS generates one packet per client at the beginning of every frame and drops those packets that were not delivered by the end of the frame. This implies that packet delay can be at most T slots. The kth frame comprises slots $t \in \{kT, \cdots, (k+1)T-1\}$, where $t, k \in \mathbb{Z}^+$ are the slot index and the frame index, respectively. This periodical frame structure and the underlying network are illustrated in Figure 5-2.

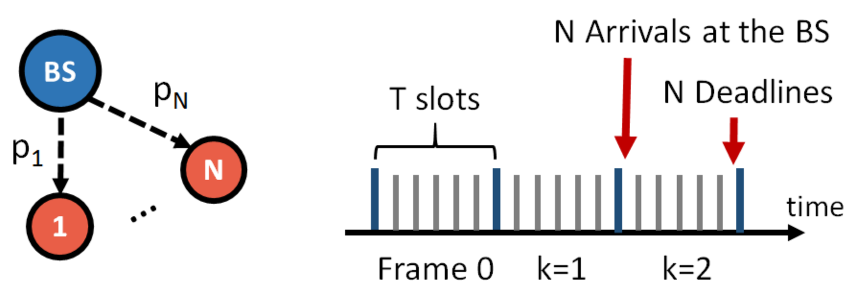

Figure 5-2: On the left, the single-hop wireless network with a BS and N clients. On the right, a timeline with the relationship between the frame structure and arrivals/deadlines. (same as Figure 2-1)

In a slot, the BS transmits a selected packet over the wireless channel. The packet is successfully delivered to client i with probability $p_i \in (0,1]$ and a transmission error occurs with probability $1-p_i$. The client sends a feedback signal to the BS after every transmission. This ACK/NACK reaches the BS instantaneously and without errors.

Admissible scheduling policies are *work conserving* and *non-anticipatory*. This class of policies is denoted by Π. Our goal in this chapter is to characterize the scheduling policy $\pi \in \Pi$ that minimizes the weighted sum of the expected AoI of the clients. This metric is thoroughly discussed in the next section.

Recall from previous chapters that, for assessing throughput, we considered the system over a single frame only. As we will see next, the AoI is different in this regard for it is *not stochastically renewed* at the beginning of every frame. An immediate consequence is that characterizing the AoI-optimal scheduling policy is more computationally demanding.

5.1.1 Age of Information

Prior to introducing the DP formulation, we discuss the impact of the frame structure on the AoI. Since packets arrive at the beginning of each frame and undelivered packets are dropped at the end of the same frame, every packet can be unequivocally identified by a tuple (k,i). Let $n_{k,i}$ be the delay of packet (k,i), i.e. the number of slots elapsed from its generation to its delivery, and let AoI_i be the positive real number that represents the Age of Information associated with client i at time slot t. The AoI_i increases linearly in time while there is no packet delivery to client i and, when an ACK is received, AoI_i is updated to $n_{k,i}$. In Figure 5-3, the evolution of AoI_i is illustrated for a given sample sequence of deliveries to client i. Notice that for every (k,i), the delay is $n_{k,i} \in \{1,2,\cdots,T\}$.

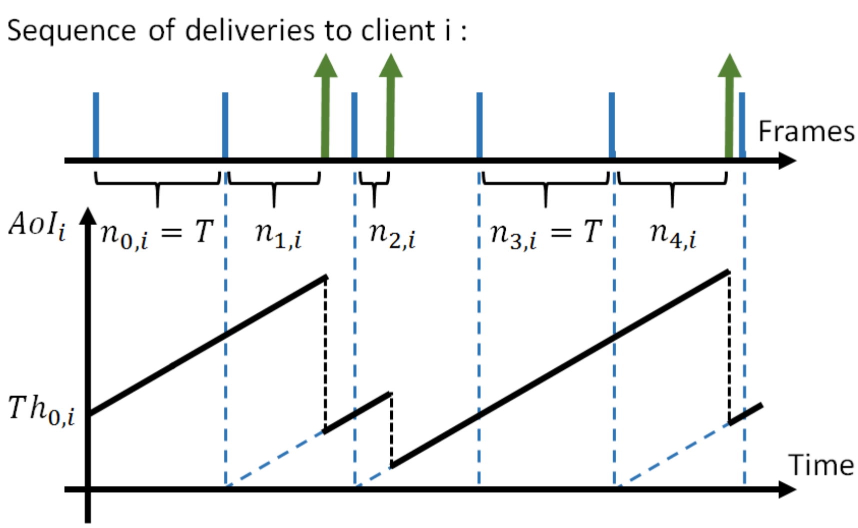

Figure 5-3: On the top, five frames of a given sample sequence of deliveries to client i. Packet deliveries are represented by the arrows. On the bottom, the evolution of the AoI_i.

Figure 5-4 depicts AoI_i in more detail and introduces the variable $h_{k,i}$. This positive integer represents the number of frames since the last delivery from client i. At the beginning of frame $k+1$, it is updated as follows

$$h_{k+1,i} = \begin{cases} h_{k,i}+1 & \text{, if packet } (k,i) \text{ was } \textit{not} \text{ delivered;} \\ 1 & \text{, otherwise.} \end{cases} \tag{5.1}$$

This parameter is initialized by the vector $\vec{h}_0 = [h_{0,1} \cdots h_{0,N}]^T$.

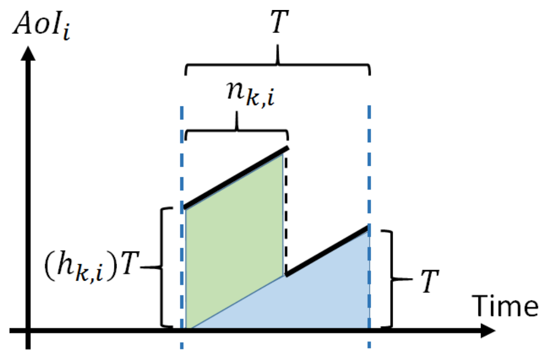

Figure 5-4: Representation of the area under AoI_i during frame k in terms of $n_{k,i}$ and $h_{k,i}$. The area of the triangle is $T^2/2$, independently of the scheduling policy.

For measuring the overall freshness of the information at the clients, we propose the metric called Frame-Average Expected Weighted Sum of the AoI, represented by EWSAoI_K. The frame-average AoI_i is associated with the area under the curve in Figure 5-4, which depends only on two parameters: $n_{k,i}$ and $h_{k,i}$. The delay, $n_{k,i}$, characterizes the performance of the scheduling policy inside the frame, while $h_{k,i}$ characterizes the performance in-between frames. Notice in Figure 5-4 that, for any frame k and client i, the area under AoI_i can be divided into a triangle of *fixed* area $T^2/2$ and a parallelogram of area $h_{k,i}Tn_{k,i}$. Therefore, for a network that runs for a total of K frames, EWSAoI_K is defined as

$$\begin{aligned} EWSAoI_K &= \frac{1}{K}\mathbb{E}\left\{\sum_{k=0}^{K-1}\sum_{i=1}^{N}\alpha_i\left(\frac{T^2}{2}+h_{k,i}Tn_{k,i}\right)\middle|\,\vec{h}_0\right\} = \\ &= \frac{\sum_{i=1}^{N}\alpha_i T^2}{2}+\frac{1}{K}\mathbb{E}\left\{\sum_{k=0}^{K-1}\sum_{i=1}^{N}\alpha_i\, h_{k,i}Tn_{k,i}\middle|\,\vec{h}_0\right\}, \end{aligned} \tag{5.2}$$

where α_i is the positive real value that represents the client's weight. The simplest form of the objective function, denoted J_K^π, is derived from (5.2) as

$$\begin{aligned} \frac{K}{T}\left[EWSAoI_K-\frac{\sum_{i=1}^{N}\alpha_i T^2}{2}\right] &= \mathbb{E}\left\{\sum_{k=0}^{K-1}\sum_{i=1}^{N}\alpha_i\, h_{k,i}\, n_{k,i}\middle|\,\vec{h}_0\right\} \\ J_K^\pi = \min_{\pi\in\Pi}\mathbb{E}&\left\{\sum_{k=0}^{K-1}\sum_{i=1}^{N}\alpha_i\, h_{k,i}\, n_{k,i}\middle|\,\vec{h}_0\right\}. \end{aligned} \tag{5.3}$$

In the following section, we formulate the DP based on the objective function in (5.3).

5.2 Problem Formulation

In this section, the discrete-time system is formulated and solved using DP. The objective function considered in this formulation is the J_K^π in (5.3) and the components of the cost-to-go function (augmented state, control, transition and reward) are associated with the network model, as described next.

5.2.1 Augmented State, Control, Transition and Reward

Consider the set of clients $\mathcal{N} = \{1,...,N\}$ and the state space $S = 2^{\mathcal{N}}$, which is the collection of all subsets of clients. Let $s_t \in S$, $t \in \{0,1,\cdots,KT-1\}$, represent the clients that have successfully received their packet by the beginning of the tth time slot, as illustrated in Figure 5-5. The tuple $(s_t, \vec{h}_k)$ is the augmented[1] state of the network, for it provides a complete characterization of the system at slot t of frame k.

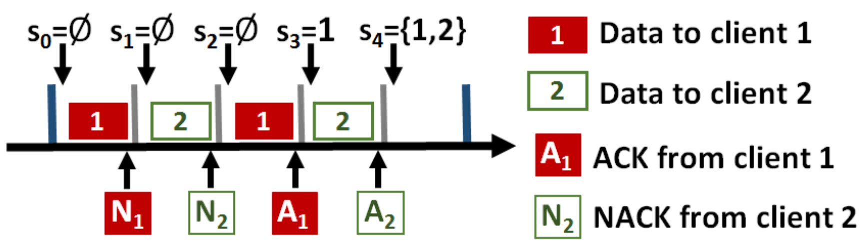

Figure 5-5: Single frame of a network with $N = 2$ and $T = 5$. At the beginning of each slot, the BS selects a packet for transmission, unless all packets were already delivered, namely $s_t = \mathcal{N}$, in which case the BS idles. (same as Figure 2-2)

Let u_t be the scheduling decision (or control) in time slot t, i.e. the client selected by the base station for transmission in that slot. Since the admissible policies are work conserving, the set of scheduling decisions which are allowed in slot t can be defined as $U_t(s_t) = \mathcal{N} \setminus s_t$. Imposing $u_t \in U_t(s_t)$, $\forall t$, guarantees that the BS will only idle when all packets have been acknowledged, namely $U_t(s_t = \mathcal{N}) = \emptyset$. The case in which the base station idles is illustrated in Figure 5-5.

[1]The set s_t is defined as the system state in previous chapters, therefore, for consistency, we emphasize here that $(s_t, \vec{h}_k)$ is the augmented state. Nonetheless, henceforth in this chapter, we will use the terms augmented state and state interchangeably to denote $(s_t, \vec{h}_k)$.

State transitions are different at frame boundaries and within frames. At the first slot of the kth frame, i.e. slots $t = kT$, $\forall k \in \{1,2,\cdots,K-1\}$, the newly generated packets impose that $s_{kT} = \emptyset$. Moreover, at the beginning of the same slot, the vector $\vec{h}_{k-1}$ evolves to $\vec{h}_k$ according to (5.1). Therefore, at the boundary of frames $k-1$ and k, each component of the state space $(s_{kT}, \vec{h}_k)$ evolves in a distinct way. The set s_{kT} is always empty, regardless of the set s_{kT-1}. The evolution of the *element* $h_{k,i}$ is divided into two cases: i) when the scheduling policy transmits packet i during slot $kT-1$, namely $u_{kT-1} = i$, then the state transition from slot $kT-1$ to slot kT depends on the feedback signal as follows

$$P(h_{k,i} = h_{k-1,i} + 1 | h_{k-1,i}) = 1 - p_i \; ; \qquad \text{[NACK is received]} \tag{5.4}$$

$$P(h_{k,i} = 1 | h_{k-1,i}) = p_i \; ; \qquad \text{[ACK is received]} \tag{5.5}$$

and ii) when the policy does not select client i, then the transition is deterministic

$$P(h_{k,i} = 1 | h_{k-1,i}) = 1 \; , \qquad \text{if } i \in s_{kT-1} \; ; \tag{5.6}$$

$$P(h_{k,i} = h_{k-1,i} + 1 | h_{k-1,i}) = 1 \; , \qquad \text{if } i \notin s_{kT-1} \; . \tag{5.7}$$

For state transitions that do not occur at frame boundaries, the vector $\vec{h}_k$ remains fixed and the evolution of the set s_t is determined by the scheduling decision, u_{t-1}, and by the feedback signal received at the end of the slot $t-1$. If the BS idles during slot $t-1$, then the set s_{t-1} does not change

$$P(s_t = s_{t-1} | s_{t-1}, u_{t-1} = \emptyset) = 1 \; . \tag{5.8}$$

In case the BS transmits a packet to client i, then s_t will depend upon the outcome of the transmission. The evolution from slot $t-1$ to slot t is characterized by the probabilities

$$P(s_t = s_{t-1} | s_{t-1}, u_{t-1} = i) = 1 - p_i \; ; \qquad \text{[NACK is received]} \tag{5.9}$$

$$P(s_t = \{s_{t-1} \cup i\} | s_{t-1}, u_{t-1} = i) = p_i \; . \qquad \text{[ACK is received]} \tag{5.10}$$

The last concept to be introduced prior to the DP formulation is the reward function.

The reward is associated to the objective function, J_K^π, rewritten to account for slot time increments, rather than frames. The new expression for (5.3) is given by

$$\begin{aligned} J_K^\pi &= \min_{\pi\in\Pi} \mathbb{E}\left\{ \sum_{k=0}^{K-1} \sum_{i=1}^{N} \alpha_i\, h_{k,i}\, n_{k,i} \,\middle|\, \vec{h}_0 \right\} = \\ &= \min_{\pi\in\Pi} \mathbb{E}\left\{ \sum_{k=0}^{K-1} \sum_{t=kT}^{(k+1)T-1} \sum_{i=1}^{N} \alpha_i\, h_{k,i}\, \mathbb{I}_{\{i\notin s_t\}} \,\middle|\, \vec{h}_0 \right\} , \end{aligned} \tag{5.11}$$

where $\mathbb{I}_{\{i\notin s_t\}}$ is an indicator function that takes the value 1 if $\{i \notin s_t\}$ is true and takes the value 0, otherwise. An interpretation of (5.11), is that the system should be charged a penalty of $\alpha_i h_{k,i}$ for every slot in which the packet from client i remains undelivered. Thus, it is straightforward that an appropriate reward function is given by

$$g_t(s_t,\vec{h}_k) = \sum_{i=1}^{N} \alpha_i\, h_{k,i}\, \mathbb{I}_{\{i\notin s_t\}} \tag{5.12}$$

With this last concept introduced, the DP formulation is presented next.

5.2.2 Dynamic Program

Substituting the reward $g_t(s_t,\vec{h}_k)$ on the objective function in (5.11) yields

$$J_K^\pi(\vec{\alpha}) = \min_{\pi\in\Pi} \sum_{k=0}^{K-1} \sum_{t=kT}^{(k+1)T-1} \mathbb{E}\left[g_t(s_t,\vec{h}_k) \,\middle|\, \vec{h}_0 \right] , \tag{5.13}$$

This system evolves in discrete steps and has a additive reward, making it suitable for a DP formulation. For a given $\vec{\alpha}$, the optimization problem in (5.13) is solved by applying the cost-to-go function $J_t(s_t,\vec{h}_k)$ iteratively, backwards in time. The initial value of the cost-to-go is $J_{KT}(s_{KT},\vec{h}_K) = 0$, for all $(s_{KT},\vec{h}_K)$, and the recursion for slot t in frame k is divided in two cases. If slot t is the *last slot* of frame k, namely $t = (k+1)T-1$, then

$$\begin{aligned} J_t(s_t,\vec{h}_k) &= \min_{u_t\in U_t(s_t)} \mathbb{E}[g_t(s_t,\vec{h}_k) + J_{t+1}(s_{t+1},\vec{h}_{k+1})] \,; \\ &= g_t(s_t,\vec{h}_k) + \min_{u_t\in U_t(s_t)} \mathbb{E}[J_{t+1}(\emptyset,\vec{h}_{k+1})] \,. \end{aligned} \tag{5.14}$$

On the other hand, if slot t is *not the last slot* of frame k, then

$$\begin{aligned} J_t(s_t,\vec{h}_k) &= \min_{u_t \in U_t(s_t)} \mathbb{E}[g_t(s_t,\vec{h}_k) + J_{t+1}(s_{t+1},\vec{h}_k)] \; ; \\ &= g_t(s_t,\vec{h}_k) + \min_{u_t \in U_t(s_t)} \mathbb{E}[J_{t+1}(s_{t+1},\vec{h}_k)] \; . \end{aligned} \tag{5.15}$$

At each step t and for every possible $(s_t,\vec{h}_k)$, the value of $J_t(s_t,\vec{h}_k)$ is attained by choosing the optimal u_t^* which minimizes the cost-to-go. By keeping track of those choices for every possible tuple $(t,s_t,\vec{h}_k)$, the optimal policy π^* is obtained. The output of the recursion at $t=0$ and for the initial vector $\vec{h}_0$ is $J_0(\emptyset,\vec{h}_0)$. This quantity is the optimal objective function, $J_K^\pi(\vec{\alpha})$, associated with π^*.

The mechanism to obtain the AoI-optimal policy for a given $\vec{\alpha}$ was described. In the next section, we compare the performances of the AoI-optimal policies with the throughput-optimal policies discussed in Chapter 2.

5.3 Numerical Results

For comparing the throughput-optimal policies described in Chapter 2 with the AoI-optimal policies described in this chapter, we consider a common underlying wireless network for both metrics. In particular, we consider the network presented in Section 5.1. Notice that the network models in Sections 2.1 and 5.1 are identical. The throughput optimization problem was defined in (2.7) and is repeated here for convenience

$$EWST(\vec{\alpha}) = \max_{\eta \in \Pi} \sum_{i=1}^{N} \alpha_i \mathbb{E}\left[D_i^\eta(0)\right] \; ,$$

where $D_i^\eta(k)$ is an indicator random variable that is equal to 1 if, by following policy η, the packet is delivered to client i during the kth frame, and zero otherwise. The AoI optimization problem is defined in (5.13) and is also repeated here

$$J_K^\pi(\vec{\alpha}) = \min_{\pi \in \Pi} \sum_{k=0}^{K-1} \sum_{t=kT}^{(k+1)T-1} \mathbb{E}\left[g_t(s_t,\vec{h}_k) \,\middle|\, \vec{h}_0 \right] \; .$$

Recall that $J_K^\pi(\vec{\alpha})$ refers to a simplified version of the actual objective function, which is given in its complete form by the relation

$$EWSAoI_K(\vec{\alpha}) = \frac{\sum_{i=1}^N \alpha_i T^2}{2} + \frac{T}{K} J_K^\pi(\vec{\alpha}) \, .$$

In Sections 2.2 and 5.2, we described DP formulations that solve the throughput and the AoI problems, providing the optimal objective functions, $\mathrm{EWST}(\vec{\alpha})$ and $\mathrm{EWSAoI}_K(\vec{\alpha})$, and the associated optimal policies. For a given vector of client weights, $\vec{\alpha}$, denote the throughput-optimal policy η^* and the AoI-optimal policy π^*. By using η^* it is possible to compute the expected throughput of each client, denoted $\mathrm{Thr}_i^{\eta^*}$. Moreover, it is also possible to compute the expected AoI associated with the throughput-optimal policy. This expected AoI for client i is denoted $\mathrm{AoI}_i^{\eta^*}$. An analogous procedure can be carried out for the AoI-optimal policy π^*, facilitating the comparison illustrated in Figure 5-6. Notice that $\mathrm{AoI}_i^{\eta^*}$ and $\mathrm{Thr}_i^{\pi^*}$ are likely to be sub-optimal.

Figure 5-6: Diagram that represents the comparison between AoI-optimal policies and throughput-optimal policies. For each fixed $\vec{\alpha}$, four vectors are obtained, one for each combination of metric and policy. One example of such vector is $(\mathrm{AoI}_i^{\eta^*})_{i=1}^N$.

By sweeping $\vec{\alpha}$ and iteratively solving the DPs for a common underlying network, the performance metrics in Figure 5-6 are collected. Those results are plotted in Figures 5-7 and 5-8. In both figures, the red crosses are computations associated with the AoI-optimal policies π^*, the blue circles are computations associated with the throughput-optimal policies η^*, and the black circles are *simulations* associated with randomizations of the throughput-optimal policies. The randomizations are described next.

Recall from Chapter 2 that, for any vector $\vec{\alpha}$, the solution to the throughput DP is a Greedy Policy. In particular, for a two-user network, only two throughput-optimal policies are found: a policy that prioritizes client 1, denoted $G1$, and a policy that prioritizes client 2, denoted $G2$. Randomizations between $G1$ and $G2$ across frames are simulated for supplementing the results, by adding more throughput-optimal policies. Each randomized policy in Figures 5-7 and 5-8 is characterized by the frequency in which policies $G1$ and $G2$ are employed throughout the frames. Eleven different frequencies were simulated.

Figure 5-7 shows the Age of Information of the clients when the network employs the AoI-optimal policies (cross) and the throughput-optimal policies (circles). Unsurprisingly, the AoI-optimal policies attain better Age of Information result, except for the two extreme points, in which all policies achieve the same performance. Those two points are associated with policies $G1$ and $G2$.

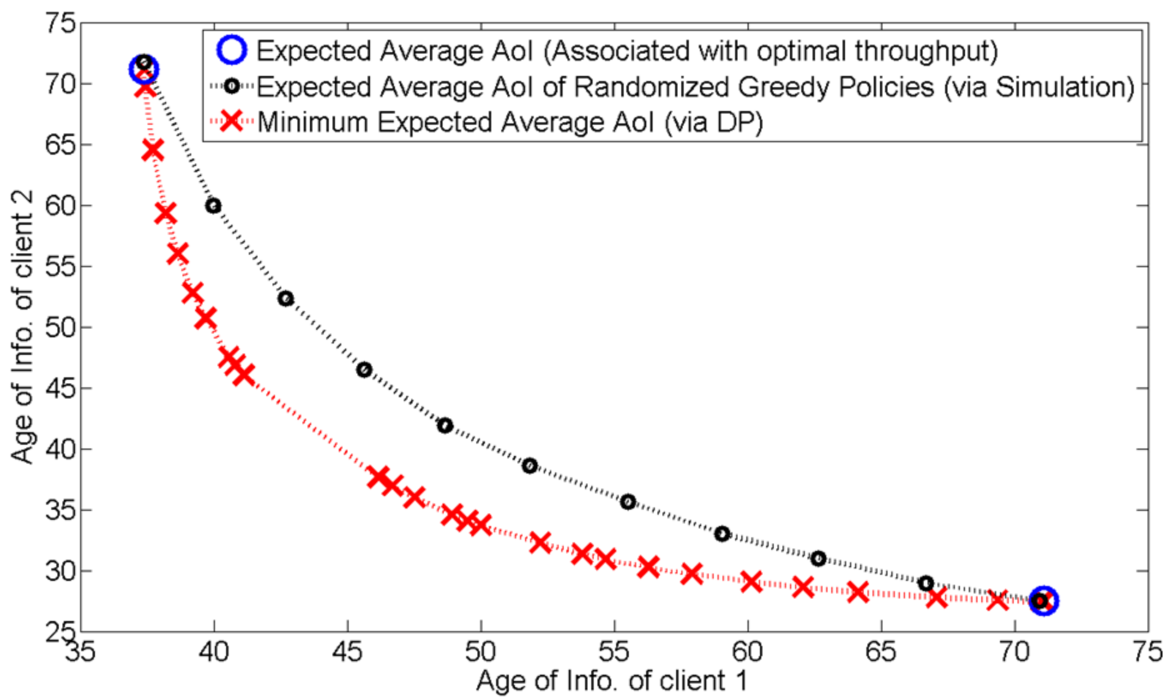

Figure 5-7: Comparison of the AoI attained by the Throughput-optimal and AoI-optimal policies in a two-user network with $T = 5$, $K = 120$, $p_1 = 1/5$, $p_2 = 1/3$ and variable vector $\vec{\alpha}$.

Figure 5-8 displays the Throughput of the clients when the network employs the AoI-optimal policies (cross) and the throughput-optimal policies (circles). The result suggests that **AoI-optimal policies are throughput optimal in the Pareto sense**[2], while the converse is not true. In the context of our network, Pareto optimality means that the throughput vector $(\text{Thr}_i^{\pi^*})_{i=1}^N$ lies on the border of the Feasible Throughput Region.

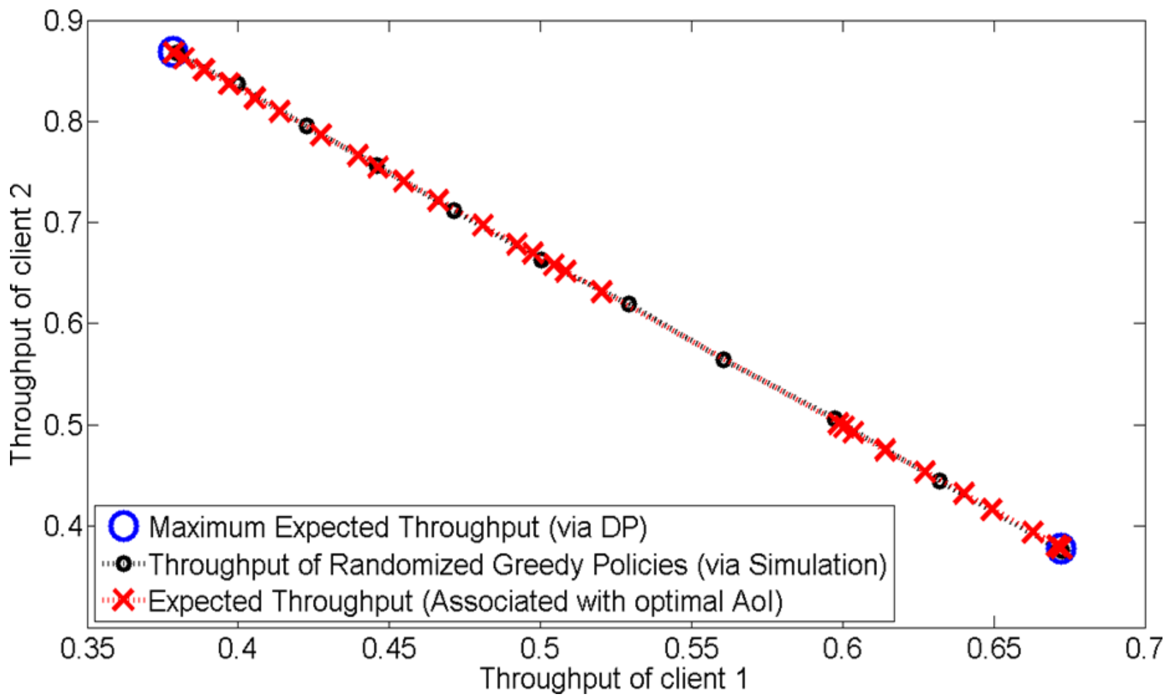

Figure 5-8: Comparison of the Throughput attained by the Throughput-optimal and AoI-optimal policies in a two-user network with $T = 5$, $K = 120$, $p_1 = 1/5$, $p_2 = 1/3$ and variable vector $\vec{\alpha}$.

If this result was extrapolated to other values of (K, T, p_i), we could conclude that AoI-optimal policies are stronger than Throughput-optimal policies. Intuitively, this makes sense, since AoI is a more stringent metric. As mentioned, for attaining a good AoI performance, the communication system needs packets with low delay being delivered regularly. Thus, a poor throughput would necessarily harm the AoI performance.

We conjecture that all work conserving policies are throughput-optimal in the Pareto sense. Meaning that the only way of wasting throughput is by idling when packets are still available for transmission. Furthermore, we also conjecture that all AoI-optimal policies are work-conserving, implying that all AoI-optimal policies are throughput-optimal.

[2]Pareto optimality means that there is no admissible policy that can increase the throughput of one of the clients without decreasing the throughput of at least one other.

A negative aspect of the AoI-optimal policies is that they can be computationally demanding. Solving the DP entails minimizing over u_t, at each slot t, for every possible $(s_t, \vec{h}_k)$. Knowing that there is a total of KT slots, 2^N different subsets s_t and the quantity $h_{k,i}$ can take any value in the set $\{1,2,\cdots,K\}$, this amounts to $\mathcal{O}(2^N K^N T)$ minimizations. Therefore, especially for networks with a high number of clients, the DP approach presented in this chapter may not be tractable.

5.4 Future Work

This chapter addressed the problem of minimizing the Age of Information in a wireless single-hop network with deadline constrained traffic. The problem was formulated and solved using Dynamic Programming. We compared the performances of AoI-optimal scheduling policies and throughput-optimal policies. Simulation suggests that AoI-optimal policies are throughput-optimal in the Pareto sense, while the opposite statement is not necessarily true.

Interesting extensions of this work are in reducing the computational complexity of the AoI problem. One possibility is to use Approximate Dynamic Programming [2] for simplifying the state space. Alternatively, the Restless MultiArmed Bandit framework [9, 27] can be applied to the problem in order to (possibly) obtain the Whittle's Index and a tractable policy. A third possibility is to characterize the AoI-optimal policy for some classes of networks, such as the class of symmetric networks analyzed in Chapter 4. Finding optimal policies for particular classes of networks might be useful for understanding the behavior of AoI-optimal policies in general.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Conclusion

In this thesis, we characterized the optimal scheduling policies for single-hop wireless networks supporting periodic real-time traffic. The *basic network* considered throughout this thesis was composed of a BS transmitting deadline constrained packets to N clients. Notice that inverting the direction of the packet transmissions, as in Figure 6-1, has no impact on the mathematical model. Therefore, all results in this thesis are also valid for networks with clients sending packets to the BS.

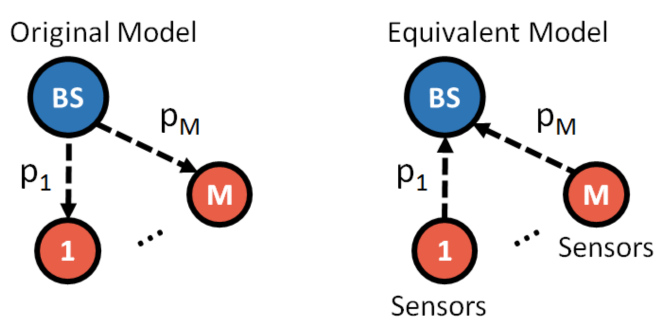

Figure 6-1: Equivalent Network Models.

In Chapter 2, the mathematical model for the basic network and the throughput optimization problem were described in detail. In addition, important definitions, such as Feasible Throughput Region and Feasibility Optimal Policy, were introduced.

In Chapter 3, the basic network model was extended to account for a general feedback mechanism. A feasibility optimal algorithm was described and a low-complexity heuristic algorithm was proposed. Both algorithms were simulated and their throughput performances compared.

In Chapter 4, the basic network model was generalized to allow for each client having multiple packet flows with heterogeneous deadlines. The throughput maximization problem was solved using DP. A class of wireless networks for which the Shortest Time to Extinction policy is throughput-optimal was defined.

In Chapter 5, the basic network was modeled using the Age of Information metric. The problem of minimizing the AoI was formulated and solved using DP. Moreover, the performance of AoI-optimal scheduling policies was numerically compared to throughput-optimal policies. Simulation results suggest that all AoI-optimal scheduling policies are throughput optimal in the Pareto sense.

The results obtained in this thesis provide useful insights for wireless networks that support real-time traffic. Chapter 3 showed how the feedback mechanism impacts the throughput-optimal policy. Chapter 4 showed that the Shortest Time to Extinction policy is not only throughput-optimal for wired networks, but also for a class of wireless networks. Chapter 5 pointed to the AoI as a metric of interest for real-time traffic and a direction for further work.

Appendix A

Proof of Theorem 4

At the beginning of every frame, the Frame-Based Max-Weight Algorithm measures the delivery debt $d_i(k)$, assigns $\alpha_i \leftarrow d_i^+(k) := \max\{d_i(k), 0\}$ and solves the maximization problem in (3.9). For analyzing the impact of this Algorithm on $EWST(\vec{\alpha})$, throughout this proof, let us redefine $d_i(k)$ to be the random variable representing the delivery debt at the beginning of frame k. Then, (3.9) is as follows

$$EWST(\vec{\alpha}) = \max_{\eta \in \Pi} \sum_{i=1}^{N} d_i^+(k) E\left[D_i^\eta(0)\right]. \tag{A.1}$$

Recalling that $D_i^\eta(k)$ is the random variable that indicates if the packet is delivered during frame k, the delivery debt can be expressed as

$$d_i(k) = \sum_{j=0}^{k-1} [q_i - D_i^\eta(j)] \tag{A.2}$$

This alternative representation allow us to treat the delivery debt as a virtual queue and apply concepts from Lyapunov Optimization [22].

First, we express the evolution of the queue length based on (A.2)

$$\begin{aligned} d_i^+(k+1) &= \max\{d_i(k+1), 0\} \\ &= \max\{d_i(k) + q_i - D_i^\eta(k), 0\} \\ &\leq \max\{d_i^+(k) + q_i - D_i^\eta(k), 0\} \\ &= \max\{d_i^+(k) - D_i^\eta(k), -q_i\} + q_i \\ &\leq \max\{d_i^+(k) - D_i^\eta(k), 0\} + q_i. \end{aligned} \tag{A.3}$$

Then, we define the Lyapunov function for frame k, which represents a measure of queue congestion:

$$L(k) := \frac{1}{2}\sum_{i=1}^{N}(d_i(k)^+)^2, \tag{A.4}$$

and simplify the following difference using (A.3)

$$\begin{aligned} L(k+1) - L(k) &= \frac{1}{2}\sum_{i=1}^{N}[(d_i^+(k+1))^2 - (d_i^+(k))^2] \\ &\leq \frac{1}{2}\sum_{i=1}^{N}[(\max\{d_i^+(k) - D_i^\eta(k), 0\} + q_i)^2 - (d_i^+(k))^2] \\ &\leq \frac{1}{2}\sum_{i=1}^{N}[(d_i^+(k) - D_i^\eta(k))^2 + q_i^2 + 2q_i \max\{d_i^+(k) - D_i^\eta(k), 0\} - (d_i^+(k))^2] \\ &= \frac{1}{2}\sum_{i=1}^{N}[-2d_i^+(k)D_i^\eta(k) + [D_i^\eta(k)]^2 + q_i^2 + 2q_i \max\{d_i^+(k) - D_i^\eta(k), 0\}] \\ &\leq \frac{1}{2}\sum_{i=1}^{N}[-2d_i^+(k)D_i^\eta(k) + [D_i^\eta(k)]^2 + q_i^2 + 2q_i d_i^+(k)] \\ &= \frac{1}{2}\sum_{i=1}^{N} 2d_i^+(k)(q_i - D_i^\eta(k)) + \frac{1}{2}\sum_{i=1}^{N}[[D_i^\eta(k)]^2 + q_i^2] \\ &\leq \sum_{i=1}^{N} d_i^+(k)(q_i - D_i^\eta(k)) + \frac{1}{2}\sum_{i=1}^{N} 2 \qquad \text{(both terms are lower than 1)} \\ &= \sum_{i=1}^{N} d_i^+(k)(q_i - D_i^\eta(k)) + N \end{aligned}$$

$$L(k+1) - L(k) \leq \sum_{i=1}^{N} d_i^+(k)(q_i - D_i^\eta(k)) + N. \tag{A.5}$$

Define the one-frame conditional Lyapunov drift

$$\Delta(k) := E[L(k+1) - L(k)|\mathscr{H}(k)], \tag{A.6}$$

where $\mathscr{H}(k)$ is the history of the vector of delivery debts $(d_i(k))_{i=1}^N$ until frame k. Applying (A.5) in the definition of $\Delta(k)$ yields:

$$\Delta(k) \leq N + \sum_{i=1}^{N} d_i^+(k)q_i - \sum_{i=1}^{N} d_i^+(k)E[D_i^\eta(k)|\mathscr{H}(k)]. \tag{A.7}$$

Let $(q_i)_{i=1}^N$ be any *feasible* vector of minimum delivery ratios. By the definition of feasibility, there exists a (history independent) randomized stationary policy $\eta \in \Pi$ that fulfills the delivery ratio requirement, i.e.

$$E[D_i^\eta(k)|\mathscr{H}(k)] = E[D_i^\eta(k)] \geq q_i, \forall i,k \quad \rightarrow \quad q_i - E[D_i^\eta(k)] \leq 0 \; .$$

Then, for the stationary policy, the following relation holds:

$$\Delta(k) \leq N + \sum_{i=1}^{N} d_i^+(k)q_i - \sum_{i=1}^{N} d_i^+(k)E[D_i^\eta(k)] \leq N. \tag{A.8}$$

The Frame-based Max-weight Algorithm, denoted FM, is designed to maximize (A.1). Consequently, it also minimizes the RHS of (A.7). Let $\Delta(k)$ be the drift associated with FM, then we have

$$\begin{aligned}\Delta(k) &\leq N + \sum_{i=1}^{N} d_i^+(k)q_i - \sum_{i=1}^{N} d_i^+(k)E[D_i^{FM}(k)|\mathscr{H}(k)] \leq \\ &\leq N + \sum_{i=1}^{N} d_i^+(k)q_i - \sum_{i=1}^{N} d_i^+(k)E[D_i^\eta(k)] \leq N. \end{aligned} \tag{A.9}$$

Taking the expectation of (A.9) over $\mathscr{H}(k)$:

$$E[\Delta(k)] \leq N + \sum_{i=1}^{N} E[d_i^+(k)]q_i - \sum_{i=1}^{N} E[d_i^+(k)]E[D_i^\eta(k)] \leq N$$

$$E[L(k+1)] - E[L(k)] \leq N - (E[D_i^{\eta}(k)] - q_i) \sum_{i=1}^{N} E[d_i^+(k)] \leq N. \tag{A.10}$$

Summing (A.10) over $k = \{0, \cdots, K-1\}$ for some positive integer K, yields:

$$E[L(K)] - E[L(0)] \leq KN - \sum_{k=0}^{K-1} (E[D_i^{\eta}(k)] - q_i) \sum_{i=1}^{N} E[d_i^+(k)] \leq KN$$

Assuming that $d_i(0) = 0, \forall i$, implies $E[L(0)] = 0$, and by the definition of $L(k)$ in (A.4)

$$E[L(K)] \leq KN \quad \rightarrow \quad E[L(K)] = \frac{1}{2} \sum_{i=1}^{N} E[(d_i(K)^+)^2] \leq KN$$

Finally, we obtain

$$E[(d_i(K)^+)^2] \leq 2KN, \forall i. \tag{A.11}$$

From the inequality $E^2[(d_i(K)^+)] \leq E[(d_i(K)^+)^2]$, it follows that

$$0 \leq \frac{E[(d_i(K)^+)]}{K} \leq \sqrt{\frac{E[(d_i(K)^+)^2]}{K^2}} \leq \sqrt{\frac{2N}{K}}\,. \tag{A.12}$$

And knowing that $d_i(K) \leq d_i^+(K)$ and $E[(d_i(K)^+)]/K \rightarrow 0$ when $K \rightarrow \infty$, gives

$$\limsup_{K \to \infty} \frac{E[(d_i(K))]}{K} \leq \lim_{K \to \infty} \frac{E[(d_i^+(K))]}{K} = 0,$$

or equivalently, using the expression of delivery debt

$$\liminf_{K \to \infty} \frac{-E[d_i(K)]}{K} = \liminf_{K \to \infty} \frac{1}{K} \sum_{j=0}^{K-1} [E[D_i(j)] - q_i] \geq 0, \forall i.$$

Therefore:

$$\liminf_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} E[D_i(k)] \geq q_i, \forall i, \tag{A.13}$$

and the Frame-based Max-weight Algorithm achieves the minimum requirement vector. The proof is complete.

□

Bibliography

[1] Baran Tan Bacinoglu, Elif Tugce Ceran, and Elif Uysal-Biyikoglu. Age of information under energy replenishment constraints. In *Information Theory and Applications Workshop*, 2015.

[2] Dimitri Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3 edition, 2005.

[3] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, 2 edition, 1992.

[4] Partha P. Bhattacharya and Anthony Ephremides. Optimal scheduling with strict deadlines. *IEEE Transactions on Automatic Control*, 34:721–728, July 1989.

[5] Partha P. Bhattacharya, Leandros Tassiulas, and Anthony Ephremides. Optimal scheduling with deadline constraints in tree networks. *IEEE Transactions on Automatic Control*, 42:1703–1705, Dec. 1997.

[6] Maice Costa, Marian Codreanu, and Anthony Ephremides. On the age of information in status update systems with packet management. *IEEE Transactions on Information Theory*, 62(4), April 2016.

[7] Robert G. Gallager. *Stochastic Processes: Theory for Applications*. Cambridge University Press, 2013.

[8] Anand Ganti, Eytan Modiano, and John N. Tsitsiklis. Optimal transmission scheduling in symmetric communication models with intermittent connectivity. *IEEE Transactions on Information Theory*, 53(3), Mar 2007.

[9] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed Bandit Allocation Indices*. Wiley, 2 edition, Mar 2011.

[10] Qing He, Di Yuan, and Anthony Ephremides. Optimizing freshness of information: On minimum age link scheduling in wireless systems. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 1–8, May 2016.

[11] I-Hong Hou. Broadcasting delay-constrained traffic over unreliable wireless links with network coding. *IEEE/ACM Transactions on Networking*, 23:728–740, Feb. 2014.

[12] I-Hong Hou, V. Borkar, and P. R. Kumar. A theory of qos for wireless. In *Proceedings of IEEE INFOCOM*, pages 486–494, Apr. 2009.

[13] I-Hong Hou and P. R. Kumar. Admission control and scheduling for qos guarantees for variable-bit-rate applications on wireless channels. In *Proceedings of ACM MOBIHOC*, pages 175–184, May 2009.

[14] I-Hong Hou and P. R. Kumar. Scheduling heterogeneous real-time traffic over fading wireless channels. In *Proceedings of IEEE INFOCOM*, Mar. 2010.

[15] Longbo Huang and Eytan Modiano. Optimizing age-of-information in a multi-class queueing system. In *IEEE International Symposium on Information Theory (ISIT)*, 2015.

[16] Sanjit Kaul, Roy Yates, and Marco Gruteser. Real-time status: How often should one update? In *Proceedings of IEEE INFOCOM mini-conference*, 2012.

[17] Kyu Seob Kim. Transmission scheduling with deadline and throughput constraints. Master's thesis, Massachusetts Institute of Technology, 2013.

[18] Kyu Seob Kim, Chih-Ping Li, Igor Kadota, and Eytan Modiano. Optimal scheduling of real-time traffic in wireless networks with delayed feedback. In *Proceedings of IEEE Allerton Conference on Communication, Control and Computing*, pages 1143–1149, Oct 2015.

[19] Kyu Seob Kim, Chih-Ping Li, and Eytan Modiano. Scheduling multicast traffic with deadlines in wireless networks. In *Proceedings of IEEE INFOCOM*, pages 2193–2201, May 2014.

[20] Ruogu Li and Atilla Eryilmaz. Scheduling for end-to-end deadline-constrained traffic with reliability requirements in multihop networks. *IEEE/ACM Transactions on Networking*, 20:1649–1662, Feb. 2012.

[21] Zhoujia Mao, Can Emre Koksal, and Ness B. Shroff. Optimal online scheduling with arbitrary hard deadlines in multihop communication networks. *IEEE/ACM Transactions on Networking*, Nov. 2014.

[22] Michael J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[23] Petar Popovski. Ultra-reliable communication in 5g wireless systems. In *Proceedings of IEEE International Conference on 5G for Ubiquitous Connectivity*, pages 146–151, Nov 2014.

[24] Vivek Raghunathan, Vivek Borkar, Min Cao, and P. R. Kumar. Index policies for real-time multicast scheduling for wireless broadcast systems. In *Proceedings of IEEE INFOCOM*, Apr 2008.

[25] Sanjay Shakkottai and R. Srikant. Scheduling real-time traffic with deadlines over a wireless channel. *Wireless Networks*, 8(1):13–26, Jan 2002.

[26] Dietrich Stoyan. *Comparison Methods for Queues and Other Stochastic Models*. Wiley Series in Probability and Statistics. Wiley, 1983.

[27] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988.