The Dissertation Committee for Jean Abou Rahal
certifies that this is the approved version of the following dissertation:

Timely Information Sharing in Communication Constrained Systems

Committee:

Gustavo de Veciana, Supervisor

Constantine Caramanis

John Hasenbein

Sanjay Shakkottai

Haris Vikalo

Timely Information Sharing in Communication Constrained Systems

by

Jean Abou Rahal

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

The University of Texas at Austin

August 2023

Dedicated to my parents, Joe and Mireille and to my brother Serge.

Acknowledgments

If I was asked to write an ideal story about my Ph.D. journey, I am sure it would not have been as meaningful and impactful as it was in the past 6 years that I have spent at the University of Texas at Austin (UT).

I would like first and foremost to express my most sincere gratitude to my Ph.D. advisor, Professor Gustavo de Veciana. I joined Prof. Gustavo's team in the Spring of 2018. Since then, he has been nothing but a true Ph.D. Father to me. I am grateful for every single advice he landed, for his constant support and guidance. Professor Gustavo was always ready to answer my questions and always made time for us to meet especially when I faced research challenges, even if those meetings occurred during his break times. His leadership, selflessness and discipline are some of his extraordinary traits that any new graduate strives to acquire. He helped me acquire technical skills necessary for my professional career, pushed my knowledge boundaries to levels I never imagined I could reach by myself and created in me a critical thinker ready to take over the next life challenges. Thank you Professor Gustavo, for everything.

I would also like to thank my dissertation committee members, Prof. Constantine Caramanis, Prof. John Hasenbein, Prof. Sanjay Shakkottai and Prof. Haris Vikalo for their valuable feedback and time throughout my time as a Ph.D. candidate. I enrolled in Prof. Caramanis's course on Large-Scale Optimization and

Prof. Haris Vikalo's course on Estimation Theory, which greatly contributed to my successful internships at Qualcomm Inc. during two consecutive summers. I additionally would like to thank Prof. Brian Evans for his guidance while I was still an undergraduate student at the American University of Beirut (AUB) and getting ready to join UT.

I would not be here today if it weren't for Prof. Zaher Dawy, Prof. Rouwaida Kanj and Prof. Joseph Constantine at AUB. Their guidance and constant support have helped me secure an acceptance at UT.

I would also like to thank my lab-mates and colleagues in the Wireless, Networking and Communications Group (WNCG) including (but definitely not limited to), Dr. Jiaxiao Zheng, Dr. Pablo Caballero Garces, Dr. Saadallah Kassir, Dr. Manan Gupta, Dr. Sandy Saab, Dr. Ahmad AlAmmouri, Dr. Yi Zhang, Dr. Ashish Katiyar, Jianhan Song, Geetha Chandrasekaran, Hasan Beytur, Parikshit Hegde, Heasung Kim, Agrim Bari, Aniruddh Venkatakrishnan, Alejandro Gomez Leos, Kartik Patel, Rebecca Adaimi, Isidoros Tziotis, Shorya Consul, Ronshee Chawla, Justin Lewis, Dave Van Veen as well as all to the many many colleagues that have been nothing but helpful throughout my time in WNCG. Kartik has been my office friend since we started our graduate studies at UT, keeping me company and always ready to answer my questions and have fun and mind-tickling conversations. Manan and I developed a close friendship evolving around our love of icecream. My good friend Ashish brought nothing but laughter and happiness to the office especially in times where I was overwhelmed with work and projects. Every single mentioned colleague has immensely contributed to my Ph.D. journey and I am forever grateful to them.

My time outside of work has been filled with friends. Starting with Nisrine Sawaya who has been there for me through my ups and downs, supportive when work got overwhelming and cheerful in my successes. I can't but also mention Emile Naous, Anthony Khoury, Karl Chahine, Dany Haddad, Ghassan Fawaz, Osman Ghandour, Maria Bassil, Rayan Eter, Yorguo, Anthony and Chantale Hachem, and everyone that I met through the Lebanese Cultural Organization that I had the pleasure to lead for two consecutive years (2020 and 2021). The memories I made with them will last forever and will always remind me of my amazing time in Austin. Not to forget about my many Sundays spent at Abdallah and Lina's, this Lebanese family that have been nothing but selfless and generous. I am as well grateful for the unwavering support and friendship of my friends in Lebanon.

Most importantly, I was blessed to have a supportive family back home, whose unwavering belief in me has been the driving force behind my pursuit of knowledge. I am forever grateful to my parents' sacrifices, their unconditional love and support. I wouldn't be who I am today if it wasn't for them.

Finally, I have become a proud Longhorn for life. As I was blessed to attend one of the most prestigious schools in the world, I will spread the values that UT has taught me wherever I go. Thank you UT, and indeed, “What starts here, changes the world!”.

Timely Information Sharing in Communication Constrained Systems

Publication No. ____________________

Jean Abou Rahal, Ph.D.
The University of Texas at Austin, 2023

Supervisor: Gustavo de Veciana

In the near future it is envisaged that there will be a proliferation of disruptive applications combining sensing capabilities with cutting-edge wireless technologies. The wide deployment and availability of sensing nodes as well as the large amounts of data being collected will call for the design of “smarter” ways of gathering and processing such data. Such networks will be driven by the need to extract the most relevant data/information that is of interest to possibly multiple agents/nodes, while optimizing the allocation of the shared limited available resources to adapt to the heterogeneity of the interests of the different nodes as well as the heterogeneity in their network conditions. Indeed, the timely sharing of relevant information is shaping to be crucial in applications where real-time decisions are to be constantly made. In the automotive industry for example, it is expected that vehicles equipped with sensing nodes could collaborate by sharing sensed information which would allow for vehicles with obstructed views to make safer decisions and/or enhance the capacity of roadways. In the infotainment industry, and more specifically in

the case of Virtual Reality (VR) applications that require large amounts of data to be constantly streamed, users at proximity of each other and that are part of the same VR experience may largely benefit from sharing resources such as edge caches that could be leveraged for the timely computation and delivery of the needed data especially in settings where different users may request the same data. The optimization of information sharing in communication constrained systems will thus be a fundamental problem underlying such systems. The focus of our work is on the modeling and analysis of these classes of problems and their implications in practical sensing systems. This thesis is composed of two main parts. In the first one, we explore the optimization of applications where timely sharing of information leads to enhanced safety and more accurate real-time decisions. We investigate novel metrics and algorithms aimed at achieving a high degree of real-time situational awareness in applications with distributed sensing nodes. In the second part of this thesis, we explore the timely sharing of information in a multi-user VR setting that would provide immersive VR experiences among the users. In particular, we explore how to support 360° VR video applications where the prediction of users' future viewings orientation is a major component as well as how to leverage overlaps in users' predictions in order to relieve the load on the shared communication network resources. Such applications face major challenges mainly tied to the heterogeneity in users' devices, predictions, network conditions, etc., which we propose to tackle through the careful design of metrics and policies robust to such heterogeneity and aimed at enhancing while achieving fair VR performance among the users.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Timely sharing of information over limited network resources is a key requirement for time-sensitive applications such as autonomous cars/robots, Augmented/Virtual Reality (AR/VR) domains, etc. Wireless systems today are specifically designed to support timely sensing networks unlocking new potentials for more accurate and efficient real-time decision-making. The emergence of autonomous cars/UAVs/robots for instance have leveraged the wide deployment of collaborative sensing capabilities to improve the exchange of real-time sensed data regarding critical events which leads to enhanced safety and smoother navigation. Additionally, the timely delivery of data in multi-user AR/VR settings that has become possible using high-throughout LTE and 5G wireless technologies has played a major role in converting such applications to untethered settings largely contributing to the improvement of the users' quality of experience.

In this thesis, we study two domains that have largely benefited from timely sharing of information. The first is pertinent to distributed sensing applications where sensors exchange information about dynamic environments to achieve a high level of real-time situational awareness. The second domain pertains to multi-user VR applications sharing constrained network resources, where large amounts of data are needed to be streamed to the user's headsets to guarantee enhanced users' VR experiences.

1.1 Optimizing Timely Sharing of Updates in Collaborative Sensing Systems

In our first research thrust, we investigate three different sensing scenarios. The first two consist of a fixed set of distributed sensing nodes persistently requesting timely updates whereas the third consists of a setting where requests for updates arrive at arbitrary time and the system has only a short period of time to deliver the updates regarding a time-varying environment. We investigate and propose new metrics and novel algorithms that capture and allow for the timely sharing of updates for various time-sensitive applications.

We explore the optimization of applications where a distributed set of nodes/sensors, e.g., automated vehicles, collaboratively exchange information over a network to achieve real-time situational-awareness. This involves exploring reasonable proxies for the usefulness of possibly delayed sensor updates and their sensitivity to the availability of network resources devoted to such exchanges. A key enabler to timely sharing of updates in such systems is tied to the joint optimization of (1) the application-level update rates, i.e., how often and when sensors update other nodes, and (2), the transmission resources allocated to, and resulting delays associated with, exchanging updates. We explore two different relevant scenarios tied to such applications, where the first one consists of an ad hoc wireless setting where a cluster of nodes share information by broadcasting on a single collision domain, and the second one additionally allows for such clusters to also exchange information via a base station. We provide for both settings solutions that characterize the interplay between network congestion and situational awareness amongst heterogeneous nodes.

Collaborative sensing systems have promised to offer major advantages towards enhancing real-time situational awareness in communication constrained settings such as for instance, collaborative sensing amongst vehicles to improve safety-critical decisions. In such applications, nodes may be modeled as both consumers and producers of sensed information. Consumers express an interest in information about particular locations, e.g., obstructed regions and/or road intersections, whilst producers broadcast updates on what they are currently able to see. To accommodate for such consumer-producer information-sharing platforms, we introduce and explore the optimization of trade-offs between the coverage and the space-time interest weighted average "age" of the information available to consumers. We initially propose a novel sensor selection approach that's driven towards selecting a subset of producers that maximizes the coverage of the consumers preferred regions and minimizes the time average age of these regions given that producers provide updates at a fixed rate. This approach would ensure one meets the consumers' demand for updates regarding their regions of interest, and further provides them with frequent updates about the covered regions. We further propose minimizing the interest weighted average age achieved by a fixed subset of producers with possibly overlapping coverage by optimizing their update rates. We prove that such problems have specific underlying properties and are thus amenable to efficient solutions.

Finally, we investigate in this research thrust the setting where time-sensitive applications with requests for updates arrive at arbitrary times, are short-lived and share a limited communication medium. An interesting application of such systems is one where vehicles driving at different speeds and heading towards an obstructed intersection may express interest in accurate information on the state of the intersection right before they reach it and thus generate requests for timely updates about the same intersection. Such requests have finite time windows in which updates will be relevant, reflecting the urgency at which they require updates regarding the time-varying processes they are interested in. The end of the window corresponds to the time at which a decision is to be made by the request, while the start of the window models the earliest possible time at which an update could be usefully sent. Thus, updates scheduled "just in time", i.e., as near to the end of the window as possible are deemed to be the best, since they reflect the most timely information about the environment's state. This is modelled through a reward depending on the time difference between the decision point and the last received update. The scheduling of updates in response to different requests at a time is not always supported. We hence investigate a setting where a single response to a single request can be scheduled at a time, which results in a challenging scheduling problem since not all requests' decisions can be delivered at the best possible time. With that in mind, we propose a novel update scheduling policy which maximizes the overall reward-rate in an adversarial setting and show through numerical evaluations that it outperforms natural baseline policies.

Part I of this thesis is devoted to exploring the timely sharing of updates in sensing settings. Results are included in part in [1], [2], [3], and submitted to *ACM*

Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'23).

1.2 Optimizing Timely Sharing of Data in Multi-User Virtual Reality Settings

In our second research thrust, we investigate the critical importance of timely sharing of data in time-sensitive applications where multiple users share limited resources, such as the edge server and wireless network capacity, etc., and where the main goal is to achieve an overall high but fair quality of experience among such users.

In particular, we consider a multi-user 360° Virtual Reality (VR) video setting where users equipped with VR headsets are served by a nearby edge server at proximity of a Base Station (BS). We introduce a Statistical Model Predictor (*SMP*) based on Deep Recurrent Neural Networks that estimates through recursive executions a statistical model for a user's viewing process which captures the different viewing paths that the user may follow during its VR session, along with the probabilities associated with each path. We propose a 360° VR video streaming approach aimed at achieving robustness to both view prediction error and to network capacity fluctuations which (1), prioritizes the delivery of the low-definition version of the entire video watched by a user, and (2), enhances the users' VR experience through the proactive scheduling and caching of high-definition predicted video portions through a policy π^F that leverages multicasting to a subset of users with similar predicted views. We evaluate SMP+π^F's performance against other baseline policies and find that it achieves high and fair VR performance across users. We finally explore through a preliminary analysis the advantages provided by both high predictability in a typical user's viewing process and low network capacity fluctuations on the user's VR experience.

Part II of this thesis is devoted to exploring the advantages tied to timely and proactively sharing predicted data to users in a multi-user VR setting. Results in this part are in preparation for future submission.

Part I

Optimizing Timely Sharing of Updates in Sensing Systems

Chapter 2

Optimizing Networked Situational Awareness

This chapter [1] studies the fundamental characteristics of systems aimed at achieving real-time situational-awareness based on distributed sensing resources. In particular we explore the joint selection of sensor update rates/policies and the allocation of communication resources towards optimizing Networked Situational-Awareness (NSA).

As an example of such a system, we consider automated vehicles leveraging collaborative sensing. Each car has access to on-board sensing resources, e.g., mmwave radar, LIDAR, cameras which provide a local perspective on their dynamic environment. Unfortunately such sensing modalities are tied to the availability of Line-of-Sight (LOS) views, meaning that certain key regions may be obstructed, e.g., a vehicle may not be able to see what is in front of the vehicle ahead of it, or a vehicle may wish to have redundant points of view of its environment to provide improved tracking and/or detection reliability. To overcome this challenge one might consider enabling vehicles to engage in collaborative sensing, wherein they exchange raw (or processed) sensor data with each other towards improving each vehicle's

[1] Publications based on this chapter: [1] J. Abou Rahal, G. De Veciana, T. Shimizu, and H. Lu, "Optimizing networked situational awareness," in 2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT). IEEE, 2019, pp. 1–8.

situational-awareness [4]. Such an approach would potentially involve sharing substantial amounts of information amongst nearby vehicles, possibly overloading communication resources. Network congestion or other transmission/processing delays in turn reduce the timeliness of the shared information, compromising the ability of automated vehicles to make reliable real-time decisions. Indeed the sensitivity of collaborative sensing systems to both the latency and capacity of the underlying communication network has motivated the industry to develop 5G wireless standards for Ultra-Reliable Low Latency Communications (URLLC).

The challenges of achieving real-time situational awareness through collaborative sensing in a communication constrained setting are many and involve several fundamental questions, including:

1. How often and when should sensing nodes update their neighbors regarding their respective environments?

2. What is an appropriate metric (or proxy thereof) to quantify situational-awareness and help drive the fair allocation of resources?

3. How should network resources be allocated among competing nodes' updates so as to optimize the overall nodes' situational awareness?

In order to study such systems, we require a well-defined metric. As discussed in more detail below, the Age-of-Information (AoI) has emerged as a simple intuitive metric: it measures how old relative to current time is the most recently received sensor update. This is, of course, only *loosely* tied to situational-awareness. Other more traditional metrics are tied to the achievable distortion/error, e.g. the Mean Square Error (MSE) of an estimated sensor node's "state" at a remote node. As we will see, these metrics are roughly aligned and provide the starting point of this chapter.

2.1 Related Work

There has been substantial interest in modeling systems involving the timely monitoring of remote processes over a network. The novelty of our work lies in the study of optimizing networked situational-awareness.

Age-of-Information (AoI), as discussed in [5], was introduced in the early 2010s as a measure which quantifies the freshness of the information a node has about a remote node's state. This metric became popular because it better represents the information freshness versus traditional latency/delay. Techniques to quantify and minimize AoI, or simply *Age* have been extensively studied in previous work, see examples, [6], [7], [8], [9] and [10]. In particular [7] studies how to optimally manage the freshness of information updates sent from a *single* source node to a destination, via a channel.

The recent work of [11] focuses on what is perhaps a more natural metric for tracking scenarios. The setting considered involves a single node monitoring a process (Brownian motion) and sending updates over a network (single queue) to a remote node which creates its best estimate for the process based on the received updates. The work poses and solves the problem of determining an optimal update strategy subject to a constraint on the long term rate of updates, where the cost is given by the time average MSE of the remote site's estimate for the process. Although this is an extremely simplified model, it gives a fundamental characterization of the problem at hand, and will serve to motivate our networked problem formulation.

The general approach proposed in this chapter is based on ideas underlying resource allocation in today's communication networks. Specifically work connecting the allocations achieved by transport protocols such as TCP to utility maximization, see e.g., [12], [13] and [14] for an in depth survey. However, our work differs from this body of work in that it addresses the joint optimization of sensor nodes' update policies and network resource allocation. As we shall see, the setting involves congestion constraints that are not easily decomposable but capture the underlying character of the problem at hand.

2.2 Chapter Contributions and Organization

In this chapter we propose a framework to explore the optimization of networked situational awareness. We study the joint optimization of both the application-level update rate, i.e., how often and when sensors update other nodes, and the transmission resources allocated to, and resulting delays associated with, sharing nodes' updates. We first consider a network scenario where nodes share a single resource, e.g., an ad hoc wireless setting where a cluster of nodes, e.g., platoon of vehicles, share information by broadcasting on a single collision domain, and find closed form expressions for both the update and transmission rates associated with this scenario. We then extend this to a setting where such clusters can in addition exchange information via a base station. In this setting we characterize the optimal solution and develop a natural distributed algorithm based on exchanging congestion prices associated with sensor nodes' update rates and associated network transmission rates. We conclude with a set of preliminary numerical evaluations to explore the algorithm's convergence and character of the resources' allocations.

The chapter is organized as follows. In Section 2.3 we motivate and propose an appropriate utility function for situational awareness. Section 2.4 describes our system model for a cluster of nodes broadcasting updates to each other over a shared ad-hoc wireless network. Section 2.5 expands our model to include clusters of nodes which can further communicate through network infrastructure. In Section 2.6, we design a dual decomposition algorithm used to jointly optimize sensor nodes' update rates and network transmission rates. Section 2.7 provides preliminary numerical results and analysis, and finally Section 2.8 concludes the chapter.

2.3 Modeling Networked Situational-Awareness

In this section we develop a simplified model for real-time situational awareness in a collaborative sensing system. We focus on a setting where sensing nodes are monitoring "independent" processes and updating their peers accordingly.

As a starting point we consider the AoI metric in a simple idealized setting. Suppose a sensor node periodically generates updates every $1/f$ seconds and each one is delayed by exactly d seconds before reaching the remote node. The time-varying AoI at the remote node is shown in Fig.2.1.

This model is idealized in that (1) updates are generated periodically while

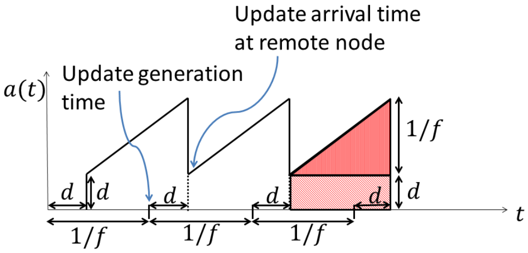


Figure 2.1: Transition from AoI to situational awareness metric

in practice they could have been generated opportunistically, e.g. based on the degree of change in the underlying process, and (2) network delays are assumed to be fixed, and (3) the focus is on AoI being the appropriate metric. The time average AoI for this idealized process is given by

$$\mathrm{AoI} = \frac{1}{2f} + d. \tag{2.1}$$

To address these limitations, let us consider the stylized result in [11]. The setting is as follows: a sensing node monitors and samples from a Brownian Motion $(W_t, t \geq 0)$ with variance σ^2. This nodes updates another remote node of the observed processes' changing state, which it does by transmitting an update over a communication link. The updates are known to have i.i.d. service times $(Y_i, i = 1, 2, \ldots)$ with the same distribution as a random variable Y. Further the sensor node is aware of the state of the link, i.e., busy or not. The key result developed in [11] is a characterization of an optimal update strategy, i.e., one that minimizes the time average MSE at the remote node subject to a constraint f on the long term frequency of updates. The optimal updating policy is parameterized by a parameter θ and can be described as follows. The optimal policy generates updates at times $(S_i,\ i = 1, 2\ldots)$, given by

$$S_{i+1} = \inf\{t \geq S_i + Y_i : |W_t - W_{S_i}| \geq \sqrt{\theta}\}, \tag{2.2}$$

i.e., the policy waits until the previous update was successfully transmitted, and then sends an update once the change in the process exceeds $\sqrt{\theta}$. The optimal θ is characterized by the following theorem.

Theorem 2.3.1 *[11] For a given constraint on the update rate f and distribution for the i.i.d. packet service times Y, the optimal θ is given by the solution to the following equation*

$$\mathbb{E}[\max(\theta, W_Y^2)] = \max\left[\frac{\sigma^2}{f}, \frac{\mathbb{E}[\max(\theta^2, W_Y^4)]}{2\theta}\right], \tag{2.3}$$

where W_Y corresponds to the distribution of a Brownian Motion $(W_t, t \geq 0)$ sampled at a random time Y. The optimal MSE is then given by

$$MSE_{opt} = \frac{\mathbb{E}[\max(\theta^2, W_Y^4)]}{6\mathbb{E}[\max(\theta, W_Y^2)]} + \sigma^2\mathbb{E}[Y]. \tag{2.4}$$

This explicit elegant characterization of an optimal updating policy captures both the role of variability of the observed process as well as the impact of packet delays. Also underlying this result is the basic observation that the sensor node should never generate an update when the channel is busy, as the update would simply wait in the queue for transmission. The limitations of this result should also be clear. In particular only a single sensor node is considered with a dedicated transmission link, along with a possibly artificial constraint on the long-term rate of updates f the node can generate.

Suppose that the update service times/delays are fixed to d, then one can show after some somewhat intricate approximations that the optimal threshold θ and associated MSE in Theorem 2.3.1 are roughly

$$\theta \approx \frac{\sigma^2}{f} \text{ and } \mathrm{MSE}_{\mathrm{opt}} \approx \sigma^2\big(\frac{1}{6f} + d\big). \tag{2.5}$$

Note that this update threshold θ matches the optimal sampling threshold derived in [15] for the case where $d = 0$. These approximate results make explicit the role played by various system parameters. As can be seen, the achievable MSE is lower bounded by $\sigma^2 d$, i.e., no matter how high the sensor update rate f is, it can not overcome the MSE arising due to the delay (service time) d to communicate with the remote node. This brings into focus the critical role that latency plays in networks supporting real-time situational awareness. Still as the allowable update rate f increases the optimal updating policy can make the MSE close to this lower bound. Indeed the reduction in MSE is inversely proportional to f.

Note that for the AoI model discussed earlier, if the observed process were a Brownian Motion, the MSE is equal to $\sigma^2 \times \mathrm{AoI}$. Following from Eq.(2.1), we have that the $\mathrm{MSE} = \sigma^2\big(\frac{1}{2f} + d\big)$. Thus the advantage of having an opportunistic update policy as exhibited in Eq.(2.2), versus a periodic updating policy with the same frequency of updates is the change in the factor multiplying $1/f$ from $\frac{1}{2}$ to $\frac{1}{6}$.

Motivated by the above results, we propose the following parametric model for situational-awareness. Suppose that a sensor update has a size ν bits, and suppose that the sensor has a dedicated link with capacity r bps. Then the delay d to transmit an update is given by $d = \frac{\nu}{r}$. Also note that if the update rate is f

(Hz), then the associated average bit rate ρ (bps) generated by the sensor node is given by $\rho = f\nu$. With these new variables the approximate optimal MSE given in Eq.(2.5) is

$$\text{MSE}_{\text{opt}} = \left(\frac{\sigma^2\nu}{6}\right)\frac{1}{\rho} + (\sigma^2\nu)\frac{1}{r},$$

with a similar functional result for AoI. This captures the impact of allowing an increased update rate ρ and/or provisioning a link with an increased capacity r on the MSE/AoI. Note that $\rho \leq r$.

Based on these observations we propose the following proxy metric that captures the situational-awareness cost.

Definition 2.3.1 *Given an update data rate ρ (bps) and link transmission rate r (bps), we model the Situational Awareness Error (SAE) of a node, $s(.,.)$ as follows:*

$$s(\rho, r) = \frac{a}{\rho} + \frac{b}{r}, \tag{2.6}$$

where $b \geq a > 0$ are constants.

Remark 2.3.1 *Through the parameters a and b, this model can capture the salient characteristics of the underlying system. For example, the variability of the underlying process (captured by σ^2) that a sensor node is monitoring would scale a, b. Also, different types of sensor nodes, e.g. video/imaging, LIDAR, might generate updates of different sizes, which would also scale a, b. Finally, the relative values of a, b model the nature of the update policy being used, e.g., deterministic, opportunistic, or other.*

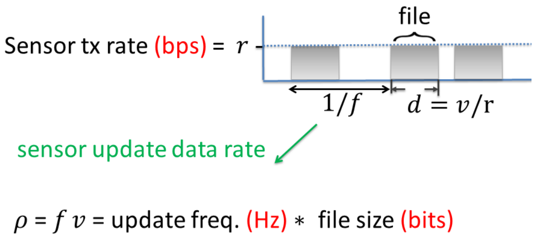

file
Sensor tx rate (bps) = r
$1/f$
$d = v/\mathrm{r}$
sensor update data rate
$\rho = f\ v$ = update freq. (Hz) ∗ file size (bits)

Figure 2.2: System model: sensor update process and SAE model

The figure below exhibits the "on/off" dynamics of the dedicated transmission link. A simple Corollary to the result developed to prove Theorem 2.3.1 gives the characteristics of this process.

Corollary 2.3.1 *Under the optimal policy given in both Theorem 2.3.1 and Eq.(2.2) for an update rate constraint f and deterministic packet service times $d = \nu/r$, the stationary dynamics of the communication link correspond to an "on/off" alternating renewal process which has an average "on/off" cycle time of $1/f$ and "on" period of length d during which the link transmits at rate r. Whence the fraction of time the link is busy is $\rho/r < 1$.*

In the next section, we shall leverage this basic result to study a more general setting.

2.4 NSA Optimization on a Shared Broadcast Network

As a first step, we consider a cluster of sensor nodes $\mathcal{N}$ sharing a single broadcast resource (single collision domain). Each node broadcasts updates to all the other nodes. Since different nodes within the cluster are located in different positions, the broadcast rate, μ_n of each node $n \in \mathcal{N}$ may be different, e.g., a node more centrally located within the cluster might have a higher broadcast rate .

Below we consider the problem of jointly optimizing the sensor nodes' update rates $\boldsymbol{\rho} = (\rho_n : n \in \mathcal{N})$ where $\rho_n = f_n \nu_n$ and transmission rates $\boldsymbol{r} = (r_n : n \in \mathcal{N})$. To that end, we define an appropriate cost function along with appropriate capacity constraints.

Objective function. The situational awareness error SAE of node n is modeled as

$$s_n(\rho_n, r_n) = \frac{a_n}{\rho_n} + \frac{b_n}{r_n},$$

where $b_n \geq a_n > 0, \ \forall n \in \mathcal{N}$. The network's overall SAE is then given by

$$g(\boldsymbol{\rho}, \boldsymbol{r}) = \sum_{n \in \mathcal{N}} s_n(\rho_n, r_n) = \sum_{n \in \mathcal{N}} \frac{a_n}{\rho_n} + \frac{b_n}{r_n}.$$

As should be clear, increasing a sensor node's update rate ρ_n and/or transmission rate r_n decreases the SAE as seen at the nodes that it is updating, leading to an improved network situational awareness. Further, note that the overall SAE is convex encoding a degree of fairness across the SAE's of the cluster's nodes.

Congestion constraints. As seen in Corollary 2.3.1, if nodes operate on dedicated links, each will act as a stationary alternating renewal process. At any random time, sensor node n could be "on" with probability ρ_n / r_n and transmitting at rate r_n. Otherwise it is "off". We model the state of a sensor node n at a typical time via a Bernoulli random variable $X_n \sim$ Bernoulli (ρ_n / r_n), where ρ_n / r_n represents the fraction of time the link is busy sending node n's update, as described in Corollary 2.3.1. We shall further make the following assumption.

Assumption 2.4.1 *(Independence of sensor nodes' processes). We shall assume that the sensor nodes' states are independent, e.g., the underlying processes they observe are independent.*

In our model, if at some point in time a sensor node is transmitting at rate r_n, it will require a fraction of the shared broadcast resource, r_n/μ_n, and if all the nodes were active, to ensure all the nodes' transmissions can be supported, we require that

$$\sum_{n \in \mathcal{N}} \frac{r_n}{\mu_n} \leq 1.$$

However since not all the nodes are active all the time, we will impose a relaxed chance constraint [16] which ensures that with high probability, $1 - \epsilon$ (for ϵ very small), the sensor nodes' update transmissions can be supported. In particular, we require

$$P\bigg(\sum_{n \in \mathcal{N}} X_n \frac{r_n}{\mu_n} > 1 \bigg) < \epsilon. \tag{2.7}$$

Using the Hoeffding bound, one can show (see Appendix A.1) that a sufficient condition for the above constraint to be satisfied, is given in the following Lemma.

Lemma 2.4.1 *Under Assumption 2.4.1, the network congestion constraint Eq.(2.7) is satisfied if*

$$\sum_{n \in \mathcal{N}} \frac{\rho_n}{\mu_n} + \omega \parallel \mathbf{r} \parallel_{\boldsymbol{\mu},2} \leq 1, \tag{2.8}$$

where $\omega = \sqrt{-\frac{1}{2}\ln(\epsilon)}$ *and* $\parallel \mathbf{x} \parallel_{\boldsymbol{\mu},q} := \bigg(\sum_{n \in \mathcal{N}} \big(\frac{x_n}{\mu_n}\big)^q \bigg)^{\frac{1}{q}}$ *which for* $q \geq 1$ *is a weighted, by positive reciprocals of* $\boldsymbol{\mu}^{-1}$*, norm.*

Given Lemma 2.4.1, one can formulate the following optimization problem of NSA, and solve for both the update and transmission rates.

Problem 2.4.1 *(***NSA optimization - single shared resource***)*

$$\min_{\boldsymbol{\rho},\boldsymbol{r}}\{g(\boldsymbol{\rho},\boldsymbol{r}) \mid \sum_{n\in\mathcal{N}}\frac{\rho_n}{\mu_n}+\omega \parallel \mathbf{r} \parallel_{\boldsymbol{\mu},2}\leq 1, \quad \boldsymbol{\rho}\leq\boldsymbol{r}\leq\boldsymbol{\mu}\}. \tag{2.9}$$

Proposition 2.4.1 *The NSA optimization Problem 2.4.1 is convex with a unique solution, which for ϵ small enough, e.g., $e^{-72}\leq\epsilon\leq e^{-2}$, is given by:*

$$\rho_n^* = \frac{\alpha_n}{\parallel \boldsymbol{\alpha} \parallel_1 + \sqrt{\omega} \parallel \boldsymbol{\beta} \parallel_{\frac{4}{3}}}\mu_n, \quad r_n^* = \frac{1}{\sqrt{\omega}}\frac{\beta_n}{\alpha_n}\left(\frac{\parallel \boldsymbol{\beta} \parallel_{\frac{4}{3}}}{\beta_n}\right)^{\frac{1}{3}}\rho_n^*,$$

for all $n\in\mathcal{N}$, and where $\boldsymbol{\alpha}=(\alpha_n=\sqrt{\frac{a_n}{\mu_n}}:n\in\mathcal{N})$ and $\boldsymbol{\beta}=(\beta_n=\sqrt{\frac{b_n}{\mu_n}}:n\in\mathcal{N})$.

Proposition 2.4.1 is proven in Appendix A.2.

The relatively simple closed form given above, is obtained by relaxing the constraint $\boldsymbol{\rho}\leq\boldsymbol{r}\leq\boldsymbol{\mu}$ and verifying that under the congestion constraint and the assumption that both $b_n\geq a_n>0$, $\forall n\in\mathcal{N}$, and $0<\omega\leq 1$, it will be satisfied.

Remark 2.4.1 *To get further insight on the NSA problem, consider the homogeneous case, where all the nodes share the same a_n, b_n parameters and broadcast capacity μ. In this case the optimal sensor node update and transmission rates are given by*

$$\rho^* = \left(\frac{1}{1+\sqrt{\omega}\sqrt{\frac{b}{a}}N^{-\frac{1}{4}}}\right)\frac{\mu}{N} \quad \textit{and} \quad r^*=\sqrt{\frac{b}{a}}\frac{1}{\sqrt{\omega}}N^{\frac{1}{4}}\rho^*,$$

where $N=|\mathcal{N}|$.

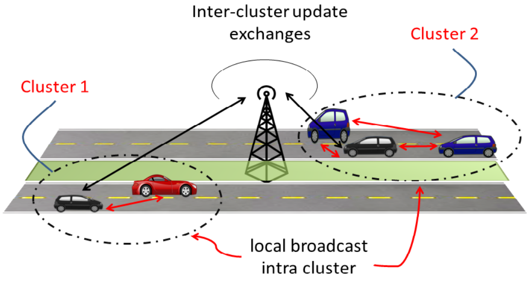


Figure 2.3: Infrastructure assisted intra-cluster update exchanges.

As can be seen, for fixed μ, as N grows, the update rate ρ^* behaves as $\frac{\mu}{N}$ while the optimal transmission rate r^* behaves as $\frac{1}{N^{\frac{3}{4}}}$. Intuitively, we might argue that as the number of sensing nodes in the network increases, optimizing NSA requires that the probability of each node staying "on" decreases as $\frac{1}{N^{\frac{1}{4}}}$, while the transmission rate allocated to each user experiences a less stringent decrease, i.e., as $\frac{1}{N^{\frac{3}{4}}}$, i.e., each user still transmits at a high rate to reduce update delays. Another interesting setting is one where the broadcast capacity scales in N, i.e., $\mu = \kappa N$, where $\kappa > 0$ is a constant. In that case, ρ^* converges to κ, while r^* increases on the order of $N^{\frac{1}{4}}$. Intuitively, when the broadcast capacity scales linearly, each node can increase its transmission rate, which reduces both its probability of being 'on' and the update delay.

2.5 NSA Optimization for Infrastructure Assisted Inter-Cluster Communication

In this section, we extend our previous model to include multiple clusters which can further exchange updates via a base station. The setting is exhibited in Fig. 2.3.

In particular the sensor nodes distribute their updates as follows: (1) *Intra-Cluster broadcast*: Sensor nodes within the same cluster communicate among themselves via a local broadcast. (2) *Inter-Cluster broadcast*: Clusters can share updates with each other by transmitting up to the BS which in turn can broadcast them down to other clusters. We will assume that inter-cluster and intra-cluster communications do not interfere with each other, i.e., operate on orthogonal frequency bands, and that clustering is such that inter-cluster broadcasts do not interfere with each other.

Let $\mathcal{C}$ denote a set of clusters, and $\mathcal{N}_c$ be the set of nodes in cluster $c \in \mathcal{C}$. We provide further discussion regarding the network under consideration.

- Broadcasting is inherently unreliable, thus the intra-cluster and base station inter-cluster broadcast could provide an additional level of reliability for intra-cluster update sharing.

- The uplink transmissions to the base station could be performed in various ways. One possibility is that each sensing node directly sends the update to the base station. Another one is that each cluster selects a *cluster head* which is in charge of forwarding the updates generated by any node in the cluster to the base station. The cluster head might be selected to have either a good connectivity to the nodes in the cluster and/or a good uplink capacity to the BS, thus reducing congestion on the BS uplink.

- For base station downlink transmissions, one could consider either an omni-directional broadcast, where all the nodes in the network receive the update forwarded from the BS instantaneously, or, one could assume that the BS uses a directional type of broadcasting, where it directs its broadcast to a single cluster, one cluster at a time. This might be necessary to ensure better reliability and higher transmission rates from the BS to the nodes. Doing so would require the base station to use several transmissions on the downlink (in fact, one for each cluster).

While the above options can be handled in our framework, below we proceed with a simple and straightforward version: each node $n \in \mathcal{N}_c$ shares its update to all the other nodes in the network. The nodes within the same cluster receive the update via both intra-cluster and base station level broadcast. The nodes in other clusters receive the update via the BS broadcast alone. Each node sends its updates to the BS which then broadcasts them to all the nodes in $\mathcal{N}$. We also make the following assumption:

Assumption 2.5.1 *(Cut-through assumption). We assume that BS uplink/downlink relaying of an update incurs no relaying delay.*

Below we consider the problem of jointly optimizing the sensor nodes' update rates $\boldsymbol{\rho} = (\rho_n : n \in \mathcal{N})$ and transmission rates $\boldsymbol{r} = (r_n : n \in \mathcal{N})$. We shall also define $\boldsymbol{\rho}^c = (\rho_n : n \in \mathcal{N}_c)$ and $\boldsymbol{r}^c = (r_n : n \in \mathcal{N}_c)$.

The overall cost function SAE, $g(\boldsymbol{\rho}, \boldsymbol{r})$, is the same as that defined earlier. We denote the intra-cluster broadcast rate of a sensor node n to the nodes in its cluster by μ_n^a, and the uplink/dowlink peak rates from/to particular nodes by μ_n^u and μ_n^d, respectively. We let $\boldsymbol{\mu^a} = (\mu_n^a : n \in \mathcal{N})$. Similarly $\boldsymbol{\mu^u} = (\mu_n^u : n \in \mathcal{N})$, and

$\boldsymbol{\mu^d} = (\mu_n^d : n \in \mathcal{N})$. We assume that μ_n^d is the same for all the nodes and hence equal to μ^d.

Congestion constraints. As seen in Corollary 2.3.1, if nodes operate in isolation, each will act as a stationary alternating renewal process.

In our infrastructure assisted network model, if at some point in time, node $n \in \mathcal{N}_c$ is transmitting an update at rate r_n, three main constraints need to be satisfied. The first one is dictated by each cluster's resources: each node will require a fraction of its cluster's resources, given by r_n/μ_n^a, and to ensure all the nodes' transmissions can be supported, we require that

$$\sum_{n \in \mathcal{N}_c} \frac{r_n}{\mu_n^a} \leq 1.$$

The second and third constraints are set to avoid congestion at the base station, i.e. the activity at the base station (which is receiving/broadcasting updates) must be supported for all nodes $n \in \mathcal{N}$ on both the uplink and the downlink, and must satisfy

$$\sum_{n \in \mathcal{N}} \frac{r_n}{\mu_n^u} \leq 1 \text{ and } \sum_{n \in \mathcal{N}} \frac{r_n}{\mu^d} \leq 1.$$

However, since not all nodes are active at the same time, we shall, once again, impose a chance constraint [16] which ensures with high probability that the sensor node's update transmissions can be supported. In particular, we require that,

$$P\bigg(\sum_{n \in \mathcal{N}_c} X_n \frac{r_n}{\mu_n^a} > 1 \bigg) < \epsilon, \ \forall c \in \mathcal{C},$$

$$P\Big(\sum_{n\in\mathcal{N}} X_n \frac{r_n}{\mu_n^u} > 1\Big) < \epsilon \text{ and } P\Big(\sum_{n\in\mathcal{N}} X_n \frac{r_n}{\mu^d} > 1\Big) < \epsilon,$$

where $X_n \sim$ Bernoulli (ρ_n/r_n) for all $n \in \mathcal{N}$. As shown in Lemma 2.4.1, sufficient constraints for these to be satisfied can be found based on the following norms:

$$\sqrt{\sum_{n\in\mathcal{N}_c} \left(\frac{r_n}{\mu_n^a}\right)^2} = \parallel \mathbf{r}^c \parallel_{\boldsymbol{\mu^a},2},$$

$$\sqrt{\sum_{n\in\mathcal{N}} \left(\frac{r_n}{\mu_n^u}\right)^2} = \parallel \mathbf{r} \parallel_{\boldsymbol{\mu^u},2} \quad \text{and} \quad \sqrt{\sum_{n\in\mathcal{N}} \left(\frac{r_n}{\mu^d}\right)^2} = \parallel \mathbf{r} \parallel_{\boldsymbol{\mu^d},2} .$$

The joint NSA optimization problem with infrastructure assistance can be formulated as follows.

Problem 2.5.1 *(***NSA optimization for infrastructure assisted setting***).*

$$\begin{aligned}\min_{\boldsymbol{\rho},\boldsymbol{r}}\{g(\boldsymbol{\rho},\boldsymbol{r}) \quad | \quad & \sum_{n\in\mathcal{N}_c} \frac{\rho_n}{\mu_n} + \omega \parallel \mathbf{r}^c \parallel_{\boldsymbol{\mu^a},2} \leq 1, \forall c \in \mathcal{C},\\ & \sum_{n\in\mathcal{N}} \frac{\rho_n}{\mu_n^u} + \omega \parallel \mathbf{r} \parallel_{\boldsymbol{\mu^u},2} \leq 1,\\ & \sum_{n\in\mathcal{N}} \frac{\rho_n}{\mu^d} + \omega \parallel \mathbf{r} \parallel_{\boldsymbol{\mu^d},2} \leq 1\}.\end{aligned}$$

Proposition 2.5.1 *Under Assumption 2.4.1, Problem 2.5.1 is convex with a unique solution characterized by first order optimality conditions which gives the following solution: For all $n \in \mathcal{N}$,*

$$\rho_n^* = \sqrt{\frac{a_n}{\frac{\lambda_c^a}{\mu_n^a} + \frac{\lambda_b^u}{\mu_n^u} + \frac{\lambda_b^d}{\mu^d}}}, \tag{2.10}$$

$$r_n^* = \sqrt[3]{\frac{b_n}{\omega\big(\frac{\lambda_c^a}{(\mu_n^a)^2\|\mathbf{r}^c\|_{\boldsymbol{\mu^a},2}} + \frac{\lambda_b^u}{(\mu_n^u)^2\|\mathbf{r}\|_{\boldsymbol{\mu^u},2}} + \frac{\lambda_b^d}{(\mu^d)^2\|\mathbf{r}\|_{\boldsymbol{\mu^d},2}}\big)}}, \tag{2.11}$$

where $\lambda_c^a,\ c \in \mathcal{C}$, are Lagrange multipliers associated with the intra-cluster congestion constraint and λ_b^u and λ_b^d are associated with BS uplink and downlink congestion constraints, respectively.

Remark 2.5.1 *An interesting observation is that while ρ_n^* depends only on congestion prices, r_n^* depends also on other nodes' transmission rates.*

In the next section, we propose a distributed algorithm to determine the optimal joint sensor node update and transmission rates' allocation across the sensor nodes.

2.6 NSA Algorithm

The algorithm works as follows. Each cluster of nodes $c \in \mathcal{C}$ updates its Lagrange multiplier λ_c^a which we refer to as *cluster price*, while the base station updates the Lagrange multipliers, λ_b^u and λ_b^d corresponding to the *uplink/downlink prices*, respectively. Meanwhile each sensor node $n \in \mathcal{N}$ responds by updating its sensor update and transmission rates ρ_n and r_n, respectively.

Suppose that each cluster elects a single node to serve as the cluster head. Its main role will be to compute the *cluster price* and establish a direct connection to exchange price information with the base station. λ_c^a is updated at the cluster head, while λ_b^u and λ_b^d are updated at the base station. λ_b^u and λ_b^d are shared with the cluster heads, who forward λ_c^a, λ_b^u and λ_b^d to the corresponding clusters' nodes. The optimal form for the sensor node update and transmission rates given in Eq.

(2.10) and (2.11) can be re-written as

$$\rho_n = \sqrt{\frac{a_n}{p_n}} \quad \text{and} \quad r_n = \sqrt[3]{\frac{b_n}{\omega q_n}}, \tag{2.12}$$

where p_n and q_n can be interpreted as *nodal update rate price* and *nodal transmission rate price* given respectively by

$$p_n = \frac{\lambda_c^a}{\mu_n^a} + \frac{\lambda_b^u}{\mu_n^u} + \frac{\lambda_b^d}{\mu^d},$$

$$q_n = \frac{1}{(\mu_n^a)^2}\frac{\lambda_c^a}{\gamma_c^a} + \frac{1}{(\mu_n^u)^2}\frac{\lambda_b^u}{\gamma_b^u} + \frac{1}{(\mu^d)^2}\frac{\lambda_b^d}{\gamma_b^d},$$

and are computed once the Lagrange multipliers are known.

Remark 2.6.1 *We shall assume that the cluster head knows the broadcast rates μ_n^a of all $n \in \mathcal{N}_c$, while the base station knows the uplink/downlink peak rates, μ_n^u and μ^b respectively, of all $n \in \mathcal{N}$.*

We summarize the algorithm as follows. Each node computes p_n and sends $\frac{a_n}{p_n}$ to the cluster head which forwards it to the base station. The cluster head computes a cluster quantity γ_c^a that we refer to as the "spare capacity" and which is defined as follows,

$$\gamma_c^a = \frac{1}{\omega} \max\left[1 - \sum_{n \in \mathcal{N}_c} \frac{1}{\mu_n^a}\sqrt{\frac{a_n}{p_n}},\ \delta\right],$$

for some small $\delta > 0$. The base station uses the quantity $\frac{a_n}{p_n}$ to compute uplink/downlink quantities, γ_b^u and γ_b^d respectively, also referred to as BS's uplink and downlink "spare capacity", and defined as,

$$\gamma_b^u = \frac{1}{\omega} \max\left[1 - \sum_{n \in \mathcal{N}} \frac{1}{\mu_n^u}\sqrt{\frac{a_n}{p_n}}, \delta\right],$$

$$\gamma_b^d = \frac{1}{\omega} \max \left[1 - \sum_{n \in \mathcal{N}} \frac{1}{\mu^d} \sqrt{\frac{a_n}{p_n}}, \delta \right].$$

Note that γ_c^a depends on the update rate prices of all the nodes in cluster c, while γ_b^u and γ_b^d depend on the update rate prices of all the nodes in sharing the BS. Given the nodal update prices and spare capacities, each cluster head node determines a cluster price on transmission rate given by λ_c^a/γ_c^a, while the base station determines the uplink/downlink rate transmission prices given by λ_b^u/γ_b^u and λ_b^d/γ_b^d, and then shares them with the corresponding clusters' heads. Note that the higher the spare capacity the lower the price of adopting a higher transmission rate for sensor nodes updates at node n. The transmissions' prices are then distributed from the cluster head amongst the cluster nodes. Each node in the network can now compute their own q_n. At this point, given that each node have their p_n and q_n, they update their ρ_n and r_n according to Eq.(2.12), then send them to the corresponding cluster heads who share them with the BS. Cluster heads update their respective prices according to

$$\lambda_c^a(t+1) = [\lambda_c^a(t) - \kappa(1 - \sum_{n \in \mathcal{N}_c} \frac{\rho_n(t)}{\mu_n^a} - \omega \parallel \mathbf{r}^c(t) \parallel_{\boldsymbol{\mu^a},2})]^+,$$

where $[x]^+ = \max[x, 0]$. Similarly, base station prices are updated at each time step (at the base station itself), as follows

$$\lambda_b^u(t+1) = [\lambda_b^u(t) - \kappa(1 - \sum_{n \in \mathcal{N}} \frac{\rho_n(t)}{\mu_n^u} - \omega \parallel \mathbf{r}(t) \parallel_{\boldsymbol{\mu^u},2})]^+,$$

$$\lambda_b^d(t+1) = [\lambda_b^d(t) - \kappa(1 - \sum_{n \in \mathcal{N}} \frac{\rho_n(t)}{\mu^d} - \omega \parallel \mathbf{r}(t) \parallel_{\boldsymbol{\mu^d},2})]^+.$$

The proposed algorithm is based on the natural dual decomposition approach [12] and [14], but accounts for the non-linear coupling on congested network resources. A such algorithm will naturally converge to the appropriate Lagrange multipliers, i.e., prices associated with the problem at hand.

2.7 Numerical Results

We conducted various preliminary numerical evaluations to explore the algorithm's convergence and character of the resources' allocations. We considered a network with three clusters of sensing nodes sharing a single base station.

2.7.1 Convergence of the NSA Algorithm

We first show that the NSA algorithm we designed in Section 2.5 converges fairly quickly. The representative results shown in Fig.2.4 correspond to the case where there are 3 clusters, each having 5 nodes. The intra-cluster broadcast capacity of each node is 100 Mbps. The uplink capacity from each node to the base station is also 100 Mbps, while the downlink capacity is 100 Mbps. Under homogeneous assumptions (i.e. model parameters a_n and b_n are the same across all clusters in the network, where $a_n = 1$ and $b_n = 6$), we exhibit the convergence of the resource allocations for a single node. We note that this algorithm can in principle adapt to changing network capacities and topologies.

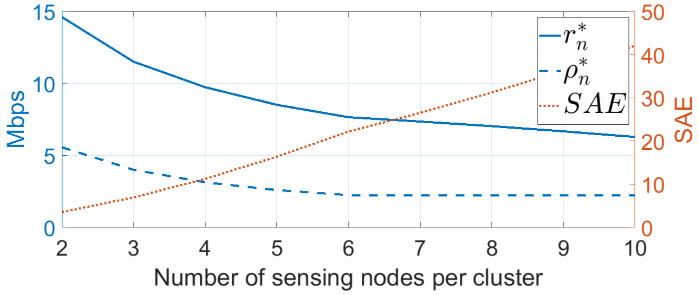


Figure 2.5: Update and transmission rates vs cluster size.

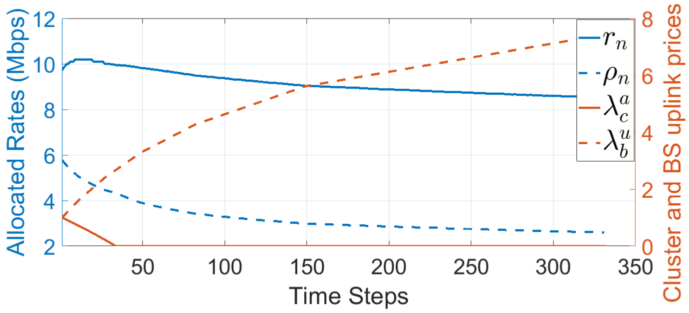


Figure 2.4: Convergence of our designed NSA algorithm.

2.7.2 Increasing the cluster size

Next, we studied how the nodes' update and transmission rates vary as a function of the number of sensing nodes (assuming again homogeneous conditions). For this purpose, we increase the number of nodes per cluster, N, from 2 to 10. The intra-cluster broadcast rate of each node is 100 Mbps, while the uplink and downlink capacities (from each node to BS and vice-versa) are both 50 Mbps. We plot below the optimal nodal update and transmission rates, as well as the overall network SAE. As expected, the transmission and propagation rates decrease as N increases.

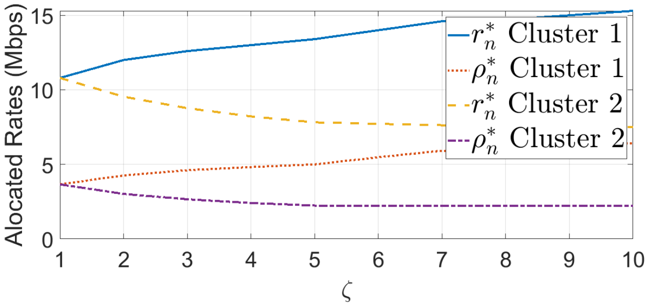


Figure 2.6: Heterogeneous Setting: Leading cluster has higher SAE sensitivity.

2.7.3 Cluster heterogeneity in nodal SAE

Another feature we explored is the impact of giving a higher priority (or SAE sensitivity) to all the nodes in a particular cluster. This time we considered a network with two clusters, each having five nodes. We assign a higher sensitivity (higher weight) to all the nodes of one cluster (say Cluster 1), while keeping the same weight to all the nodes of Cluster 2. An example of this scenario is when a cluster of cars is moving faster than another one. We implement this scenario as follows: Starting with $a_n = 1$ and $b_n = 6$ for all nodes in Cluster 1, we multiply them by some constant, say ζ, which we keep increasing. We then plot the rates for both clusters in function of the constant. All capacities are 100 Mbps. As expected, more rate is allocated to Cluster 1, showing that NSA optimization requires more updates and faster transmission rates to these nodes.

2.7.4 Increasing the imbalance of nodes across clusters

Finally, we are interested in understanding the effect of increasing the number of nodes in one of the clusters. For this purpose, we considered a network with

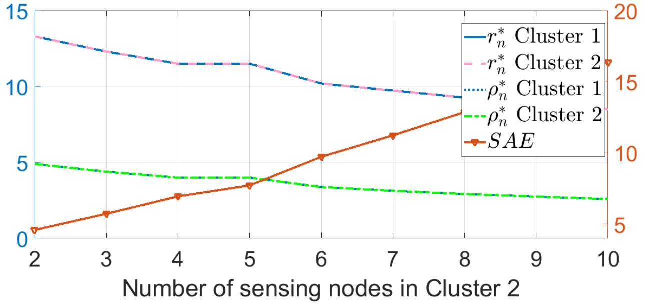


Figure 2.7: Clusters with imbalanced number of nodes.

two clusters, 1 and 2, and a single base station. The number of nodes in Cluster 1 is kept fixed at 5, while we vary the number of nodes in Cluster 2 from 2 to 10. Below, we plot the resulting update and transmission rates, as well as the overall network SAE. We observe that our algorithm achieves fairness, which results in an equal allocation of resources among all nodes.

2.8 Chapter conclusion

We proposed in this chapter a framework to study distributed collaborative sensing of a dynamic environment based on sharing information over limited network resources. A proposed model for situational awareness, SAE, was introduced. It is dependent on both the environment variability and the sensor heterogeneity. The main goal was to establish key trade-offs among sensors' update and transmission rates.

We considered first a simple setting where a cluster of nodes are sharing updates over a single communication resource, which we referred to as intra-cluster broadcast, then we extended the model to include multiple clusters sharing updates via a single base station, referred to as inter-cluster broadcast. We also developed a new algorithm, NSA algorithm, geared at jointly minimizing the SAE, which could optimize the allocation of resources to heterogeneous sensor nodes and time varying network capacity (and topology).

In our future work, we would like to explore more in depth a more general network setting where path routing and resource allocation decisions need to be made. This would follow from the extension of the current model to include the impact of relaying delays at the base station on SAE, along with the impact of the geographical positioning of the sensor nodes relatively to the cluster head, which will directly affect the SAE model.

Chapter 3

Optimizing Timely Coverage in Communication Constrained Collaborative Sensing Systems

In the near future it is envisaged that there will be several disruptions and challenges to the automotive and wireless industries. Amongst these, an intriguing and challenging one will be the emergence of automated cars (also UAVs, robots, etc.) with the ability to collaboratively navigate through complex environments. In order to enable such functionality, it is expected that nodes will collaborate by sharing sensed information, e.g., cars share their views of obstructed locations in their environment. The aim is to achieve a high degree of "real-time situational-awareness," i.e., to detect/recognize and then effectively track dynamic objects in their vicinity so as to improve safety-critical decisions. To that end, it is expected that vehicles will not only rely on on-board multimodal sensing, but also share (raw or fused) sensing information with each other, with the goal of facilitating coordination. This may require the transport of substantial volumes of data among cars, as well as to/from the network edge and/or cloud. The optimization of information sharing in a communication constrained system will thus be a fundamental problem underlying such systems. The focus of this chapter [1] is on the modeling and

[1] Publications based on this chapter [2] J. Abou Rahal, G. de Veciana, T. Shimizu, and H. Lu, "Optimizing timely coverage in communication constrained collaborative sensing systems," in 2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless

analysis of this problem and its implications for collaborative sensing systems.

A key step in this direction is to identify appropriate/usable metrics to assess how well an information sharing policy is performing. This involves at least two concerns. On the one hand, one is interested in *coverage*, i.e., the fraction of the relevant region that a set of collaborating nodes (producers) is able to view. On the other hand, for dynamic environments, one is interested in the timeliness of the available information across space, e.g., the Age of Information (AoI), when sensors periodically share what they see. Intuitively high coverage is achieved by ensuring that all sensors disseminate their (possibly redundant) information to all the other relevant nodes while the minimization of age may involve giving some well positioned sensors/nodes a higher update rate or leveraging overlaps among sensors' coverage sets. Additionally it is of interest to incorporate contextual information in that nodes may want to have a higher awareness of the on-goings in close proximity or uncertain regions on their path, e.g., it is more critical for a car to have fresh information of on-goings in its neighborhood or of obstructed regions at intersections they are about to enter rather than receiving frequent updates about distant locations. Roughly speaking optimizing the "timely coverage" for a collaborative sensing system requires modeling the relative value/usefulness of each sensor's updates, e.g., in terms of the overall coverage, importance, and timeliness.

We focus on four major intertwined questions:

Networks (WiOPT). IEEE, 2020, pp. 1–8.; and [3] J. Abou Rahal, G. de Veciana, T. Shimizu, and H. Lu, "Optimizing timely coverage in communication constrained collaborative sensing systems," IEEE Transactions on Control of Network Systems, 2022.

1. *Can one provide a model and metrics to evaluate the coverage/timeliness trade-offs achieved by a given information sharing policy for a set of collaborative sensors?*

2. *Assuming that a fixed number of sensing nodes broadcast information at a fixed rate regarding their overlapping coverage, can one determine the best subset to participate in information exchanges so as to meet the overall nodes' demand for timely updates about the regions of most interest?*

3. *Assuming a subset of sensors is chosen, can one jointly optimize their update rates to minimize the interest weighted (space-time) average age over their coverage set?*

4. *How do optimized information sharing policies compare to simple policies as a function of the sensor/node density, i.e., inherent overlap, and system communication capacity?*

3.1 Related Work

A key motivating application for this work is collaborative sensing in support of automated vehicles. The basic idea is to facilitate real-time exchanges of sensor information among vehicles and/or road side units to enhance their 'situational awareness' in obstructed and dynamic environments, see e.g., [17–21] and recent work in [22, 23] which leverage stochastic geometry to model and analyze collaborative sensing coverage in obstructed environments as a function of the penetration of vehicles with sensing capabilities.

When addressing real-time situational awareness, it is key that the decision-making nodes have access to timely information. The modeling and delivery of timely information has recently received substantial attention, see e.g., [24, 25]. The newly proposed metric, Age of Information (AoI) became popular since it better captures information freshness as compared to the traditional delay metric. AoI has been extensively studied in the literature, see e.g., [7, 26–28]. A close work to ours is that of [29] where the goal is to design an update policy that allows a single cache to update stored files one at a time by accessing the server. A policy geared at minimizing the average AoI of all the files stored in the cache is derived, given that the update duration time depends on the file size. The work in [30] is perhaps the closest to that in this chapter in that it addresses the issue of optimizing the overall AoI by carefully choosing sensors' update rates and allocating network resources. However, by contrast with these works, in this chapter, we model and explore the impact that updates from multiple sensors with possibly overlapping fields of view will have on the AoI, as well as trade-offs between coverage and timeliness.

Many instances of coverage and sensor selection problems, see e.g., [31] are known to be submodular which in turn are amenable to greedy approximations, see e.g., [32]. To our knowledge, this chapter is distinct from previous work in that it introduces and explores a new fundamental trade-off between coverage and AoI in a collaborative sensing system.

3.2 Chapter Contributions and Organization

Given a set of sensors periodically broadcasting updates (at possibly different rates) regarding their coverage sets we define and characterize the interest weighted (space-time) average age for the information exchanged. To the best of our knowledge this is the first work addressing the "timely coverage" for a set of collaborative sensors. We model our network of sensing nodes as a set of information producers and consumers and explore the resource allocation and performance trade-offs in such systems. In particular we formulate and study two possible settings.

The first captures a trade-off between maximizing the interest weighted coverage and minimizing the interest weighted average age of the spatial information requested by consumers about regions where they lack timely updates, e.g., obstructed regions or regions with high uncertainty, when all the sensors have the same update rate. We show that this weighted coverage-age trade-off optimization problem has a submodular structure which leads to efficient greedy optimization algorithms. In the second setting we fix the subset of sensors, e.g., all that are available or those selected in the first setting, which now act as producers of information, and explore the benefits of jointly optimizing their periodic update rates towards minimizing the interest weighted average age. When producers have non-overlapping coverage sets, we show that their optimal periodic update rates are proportional to the square root of their coverage's weights. However, more generally, the interest weighted average age minimization problem has a non-convex/non-concave structure. We explore the use of the Frank-Wolfe gradient method to show the potential benefits of update rate optimization for collaborative sensing.

A numerical evaluation of the benefits of these approaches from the point of view of coverage and interest weighted average age achieved by consumers is conducted. It exhibits the possible advantages that constraining both the number of active producers as well as the resource allocation amongst these active nodes in a collaborative sensing setting should play, particularly in congested environments with limited communication resources.

The chapter is organized as follows. In Section 3.3 we introduce our system model. In Section 3.4, we develop metrics for collaborative sensing based on the AoI. In Section 3.5 we introduce and study the problem of selecting a subset of sensors that achieves maximal coverage of the consumers' preferred regions, as well as minimal weighted age of the covered regions, while in Section 3.6 we consider the minimization of the weighted age function by jointly optimizing the update rates for a fixed set of sensors. Section 3.7 presents our numerical results and analysis of the underlying characteristics of collaborative sensing, and Section 3.8 concludes the chapter.

3.3 System Model

We shall begin by formally describing our model for a collaborative sensing system and the associated notation.

3.3.1 Sensor coverage sets, consumers, producers and interest weighted measures

Without loss of generality we consider a set of sensors V in a given overall region $R \subset \mathbb{R}^2$. Sensors are indexed by their locations $v \in V$ and the coverage (field-of-view) of sensor v in a given environment is denoted by a subset $C_v \subseteq R$. Given a subset of sensors $X \subseteq V$, we denote X's overall coverage by $C(X) := \bigcup_{v \in X} C_v$. In the remainder of the chapter, we assume that the sensors' coverage regions are fixed, e.g., sensors installed on fixed road-side units or base stations, and that the underlying setting of interest is one where they observe a dynamic environment.

The coverage sets for a subset of sensors X, i.e., $(C_v,\ v \in X)$ induce a partition of the overall coverage set $C(X)$ which we denote by $\mathcal{P}^X = \{P_i^X,\ i = 1, ..., |\mathcal{P}^X|\}$. Each subset of the partition P_i^X is such that each location $x \in P_i^X$ can be seen by the *same* subset of sensors $V_i^X \subseteq V$, i.e., such that $x \in P_i^X$ if and only if $x \in C_v$ for all $v \in V_i^X$. It should be clear that if $i \neq j$ then $P_i^X \cap P_j^X = \emptyset$. Further it should be clear that $\cup_{i=1}^{|\mathcal{P}^X|} P_i^X = C(X)$, thus we have a partition of $C(X)$. In fact, assuming it is nonempty, if we further include an additional set $R \setminus C(X)$ corresponding to the locations which are not covered by X, we get a partition of the overall region R. It is also possible that the coverage sets of two or more sensors intersect on a set of measure zero. For simplicity, and to avoid unnecessary burdens, we assume that all sets of the partition have non-zero area, or remove sets of measure zero.

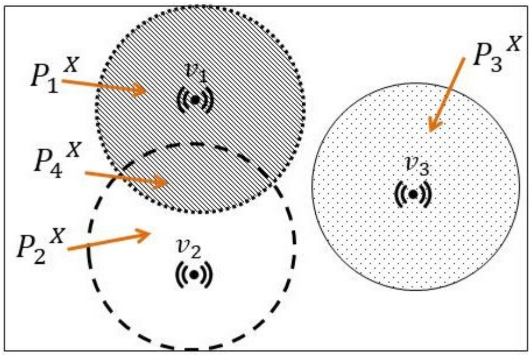


Figure 3.1: Three sensors observing their environments.

Fig. 3.1 illustrates three sensors $X = \{v_1, v_2, v_3\}$ which for simplicity have each an unobstructed disc coverage set. Sensors v_1 and v_2 have overlapping coverage regions. The figure also exhibits the four subsets in the induced partition, $\mathcal{P}^X = \{P_1^X, P_2^X, P_3^X, P_4^X\}$.

Without loss of generality we suppose each sensor node is simultaneously a consumer and a producer of information which may broadcast periodic updates regarding regions it is able to see. A consumer indicates its interest in information regarding various locations through a *spatial interest measure.* In turn the sum of the consumers' spatial interest measures captures the aggregate interest of consumers. These are formally defined below.

Definition 3.3.1 ***(Consumer's interest measure)*** *A consumer $v \in V$ indicates interest in timely information about the environment via a spatial interest measure $w_v(.)$ on R.*

Definition 3.3.2 ***(Aggregated consumers' interest measure)*** *The aggregated* *consumers' interest measure* $w(.)$ *is given by the sum of the consumers' interest measures, i.e.,* $w(.) = \sum_{v \in V} w_v(.)$ *on* R*.*

Definition 3.3.3 ***(Weighted coverage)*** *The overall weighted coverage of the region covered by a subset of sensors* X*, i.e.,* $C(X)$*, is given by* $w(C(X)) = \sum_{v \in V} w_v(C(X))$*.*

For example, if $w(.)$ corresponds to the area measure, then $w(C(X))$ denotes the area covered by the sensors in X, and if normalized, $w(C(X))/w(R)$ represents the fraction of the region R which is covered. The weight measure provides a flexible means to model the importance of various locations, and/or to model the relative importance of a region from the perspective of information sharing. Note that in general the weight measure could be continuous or discrete. In the latter case we envisage a measure placed at discrete locations corresponding to *anchor points* which based on the known geometry of the environment may have higher importance, e.g., intersections for incoming vehicles or locations obstructed by other vehicles.

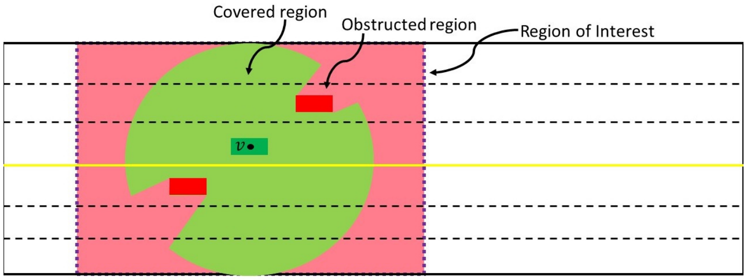


Figure 3.2: Region of interest and coverage of a consumer/producer node.

In some practical settings, a consumer v's interest may be limited to a smaller region, say R_v. For example, a vehicle with a response time of t_{interest} moving at a speed s would primarily care about what is happening in a region $s \cdot t_{\text{interest}}$ around it. Thus R_v might be modelled as a rectangle (centered at v's location) of length $2 \cdot s \cdot t_{\text{interest}}$ and width typically covering the road and surrounding areas. Fig. 3.2 illustrates the coverage of a sensor v (green region), obstructed by neighbouring vehicles (red region behind the vehicles), as well as its rectangular region of interest R_v. In this case, v's spatial interest measure would be supported by the red region. Assuming that sensor v's location on the road is $v = (x_v, y_v)$, where x_v and y_v stand for the $x - y$ coordinates of v in 2-D, and the origin 0 is at the center of R, then R_v is defined as, $R_v := R \cap \left(\left[-\frac{w_{road}}{2}, \frac{w_{road}}{2}\right] \times [x_v - s \cdot t_{\text{interest}}, x_v + s \cdot t_{\text{interest}}]\right)$, where w_{road} denotes the width of the road.

In some settings, consumers may only express an interest in obstructed regions corresponding to regions where they have significant uncertainty or blind spots, as depicted in Fig. 3.3. Thus for example, consumer v might place a point mass/anchor point in front of the truck. In the shown example, this anchor point falls in the coverage regions of the two other sensors (blue and green), which can thus in principle help out.

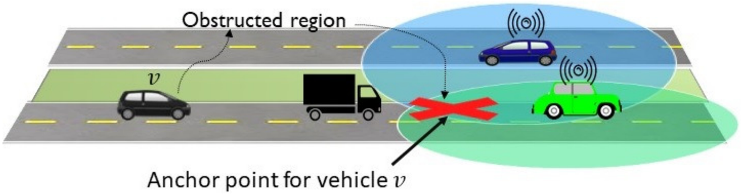


Figure 3.3: Consumer requesting timely updates about the obstructed anchor point.

3.3.2 Network capacity, sensor updates, and AoI

We shall assume that each producer v generates periodic updates regarding its coverage set C_v at a rate of r_v updates per second, i.e., the update interval is thus $\frac{1}{r_v}$ seconds. The updates are either broadcast to the other sensors or shared with a central controller. For any subset $X \subseteq V$ we let $\mathbf{r}(X) = (r_v : v \in X)$ denote the vector of update rates for the sensors in X. The delay for sensor v to access a shared communication medium and transmit its update is assumed to be exactly, or at most, d_v. Thus the fraction of time sensor v holds the medium is d_v over the update interval $1/r_v$, i.e., $d_v r_v$. The selected update rates for a set of sensors X must then satisfy a "capacity" constraint,

$$\sum_{v \in X} d_v r_v \leq 1, \tag{3.1}$$

ensuring the medium is not overbooked. Note in practice, depending on the character of the medium, one would require a back off $\sum_{v \in X} d_v r_v \leq 1 - \epsilon$ for some $\epsilon > 0$ to ensure minimal queuing and contention delays. The required back off will depend on the details of channel access and/or scheduler. For simplicity we will suppress ϵ in the sequel, and use (3.1) as the capacity constraint while assuming no overlapped/collided transmissions.

A natural metric that captures the freshness of the received updates is the Age of the Information (AoI) available at the consumers. Fig. 3.4 exhibits the time-varying AoI at a consumer for such a periodic update process from a single producer at rate r_v and with transmission delay d_v. In the sequel, we will consider both *average* age and the probability that the age exceeds a pre-specified threshold at a random time at a consumer v. For example the average age for the process shown in Fig. 3.4 is given by,

$$\textit{average} \text{ age of sensor } v = d_v + \frac{1}{2r_v}. \tag{3.2}$$

To keep things simple we will assume $d_v = d$ for all $v \in V$, i.e., they are either identical or bounded by d.

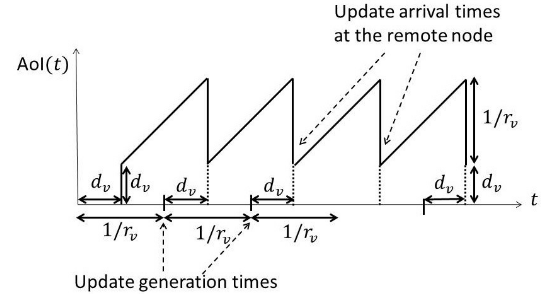


Figure 3.4: Time-varying age of information.

However since producers' coverage sets may overlap, consumers may receive updates from multiple producers for the same location, which may result in a reduction in the age of the available information. We define a generic age function for such overlapping regions as follows.

Definition 3.3.4 ***(Age of regions with overlapping sensor updates)*** *Suppose a set of producers X transmit periodic updates at rates $\mathbf{r}(X) = (r_v, v \in X)$. Recall that X induces a partition where the locations in P_i^X are covered by a set of sensors V_i^X. The age of P_i^X thus depends on the update rates of these sensors, i.e., $\mathbf{r}(V_i^X)$. With a slight abuse of notation we will define an* ***age function*** *which captures a* *proxy for the age (e.g., average or probability of exceeding a threshold) of the set* P_i^X *as*

$$\mathrm{age}(P_i^X) = \mathrm{age}(\mathbf{r}(V_i^X)), \tag{3.3}$$

with the intention of emphasizing its dependence on the associated sensors' update rates.

In the next section we shall explore the characteristics of the age as a function of the number of sensors and their associated update rates.

The age of a region as introduced in Definition 3.3.4 can be viewed as a dis-utility (cost) function that is tied to the resources allocated to the set of sensors observing this region. Consumers express interest in specific locations while sensor nodes having direct access to these regions become potential producers. We define the *weighted age function* as being an aggregated dis-utility function based on a linear combination of the aggregated consumers' spatial interest in particular regions and the age of these regions.

Definition 3.3.5 ***(Aggregated interest weighted age for a set of sensors)*** *Given a weighted coverage measure vector* $\mathbf{w} = (w(P_i^X),\ P_i^X \subseteq \mathcal{P}^X,\ i = 1, \ldots, |\mathcal{P}^X|)$ *on partition* $\mathcal{P}^X$ *induced by a set of sensors* X*, where* $\cup_{i=1}^{|\mathcal{P}^X|} P_i^X = C(X)$*, and sensor update rates* $\mathbf{r}(X)$*, the* ***aggregated interest weighted age*** *of the coverage set* $C(X)$ *associated with* X *is given by,*

$$a(X, \mathbf{r}(X)) := \sum_{i=1}^{|\mathcal{P}^X|} w(P_i^X)\mathrm{age}(P_i^X) = \sum_{i=1}^{|\mathcal{P}^X|} w(P_i^X)\mathrm{age}(\mathbf{r}(V_i^X)).$$

*If $w(.)$ is a measure corresponding to the area, we say $a(X, \mathbf{r}(X))$ is the **area weighted age** of $C(X)$. If it is further divided by $w(C(X))$ it will be referred to as the **normalized weighted age**.*

For simplicity we shall write $a(X, \mathbf{r}(X))$ as $a(X)$.

3.3.3 Problem formulation

There exists different approaches to achieving trade-offs between the weighted coverage of a subset of sensors and its associated aggregated interest weight. Our goal in the remainder of the chapter is to investigate one particular trade-off summarized as follows. We consider the setting where the target is to first meet the overall consumers' demand for updates about their regions of interest, and then to provide them with frequent updates about the covered regions. For that purpose in Section 3.5 we devise an algorithm that selects producers in V to first maximize the weighted coverage of the consumers' regions of interest and then once maximal weighted coverage is achieved, proceeds with selecting producers to minimize the weighted age of the covered regions.

3.4 Characterizing the Age Function

In this section we define and characterize the properties of two possible age functions, as introduced in Definition 3.3.4.

3.4.1 Definition and computation of the age function

We first consider two simple motivational examples. Recall that the age function depends on a vector of update rates of sensors which see a given location.

As discussed in the previous section, if periodic updates from only a single sensor at rate r_1 are available, then the *average* age, denoted $\overline{\text{age}}$ depends on the scalar r_1 and is given by,

$$\overline{\text{age}}(r_1) = d + \frac{1}{2r_1}.$$

Let A_1 be a random variable denoting the age of the saw-tooth function when viewed at a random time (see Fig. 3.4). Given the saw-tooth function's linear age growth, it should be clear that $A_1 \sim d + \frac{1}{r_1}U_1$, where $U_1 \sim \text{Unif}[0, 1]$.

We now define two age functions, the average age, $\overline{\text{age}}$, and the γ-age violation, age_γ, given by,

$$\overline{\text{age}}(r_1) = \mathbb{E}[A_1] \quad \text{and} \quad \text{age}_\gamma(r_1) = \mathbb{P}(A_1 > \gamma),$$

where $\gamma \geq 0$ is a target age one would not wish to exceed.

Definition 3.4.1 ***(Age functions)*** *Consider a region observed by n sensors generating periodic updates at rates $\mathbf{r} = (r_1, \ldots, r_n)$ and with associated transmission delays d such that $d \leq \frac{1}{r_v}$ for all $v = 1, \ldots, n$. Assuming the phases of the sensors' periodic updates are independent and uniformly distributed then the* **average age** *and* **γ-age violation** *functions of locations in a region seen by sensors with update rates $\mathbf{r}$ are*

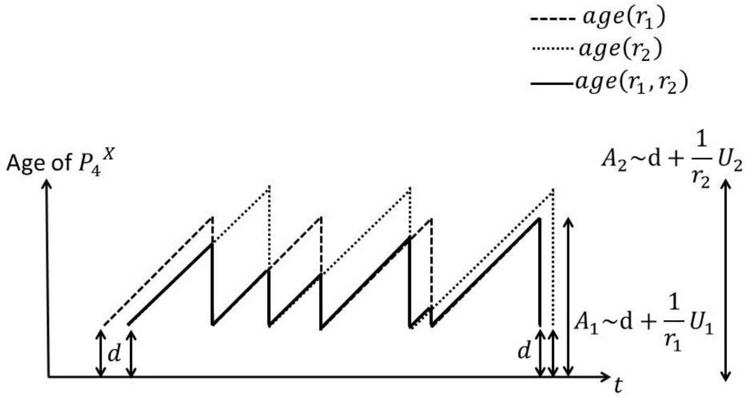
$$\overline{\text{age}}(\mathbf{r}) = \mathbb{E}[A] \quad \text{and} \quad \text{age}_\gamma(\mathbf{r}) = \mathbb{P}(A \geq \gamma),$$

where $A = \min_{v=1,\ldots,n}[A_v]$ and $A_v \sim d + \frac{1}{r_v}U_v$ and $U_v \sim \text{Unif}[0, 1]$ are independent of each other.

Suppose there are in fact updates from two sensors covering a given set in the partition, e.g., as shown in Fig. 3.1, sensors v_1 and v_2 are providing updates of region P_4^X with update rates $\mathbf{r} = (r_1, r_2)$. Without loss of generality, assume $r_1 \geq r_2$. As shown in Fig. 3.5, the dashed and dotted saw-tooth functions correspond to the updates of the two sensors. Assuming that the phases of the saw-tooth curves are randomly distributed, and no transmission fail, it is easy to see that the average age at a typical time is given by the minimum of the two functions, i.e.,

$$\overline{\text{age}}(\mathbf{r}) = \mathbb{E}\left[\min[A_1, A_2]\right] = d + \frac{1}{r_1}\left(\frac{1}{2} - \frac{1}{6}\frac{r_2}{r_1}\right),$$

where $A_1 \sim d + \frac{1}{r_1}U_1$ and $A_2 \sim d + \frac{1}{r_2}U_2$, and where U_1, U_2 are uniformly distributed and assumed to be i.i.d., and A_1, A_2 correspond to the ages of the updates from Sensors 1 and 2 observed at a random time. The reduction in age due to redundancy in the sensors' updates is clear. The probability of γ-age violation shares similar properties as the average age.



$age(r_1)$
$age(r_2)$
$age(r_1, r_2)$
Age of $P_4{}^X$
$A_2 \sim \text{d} + \frac{1}{r_2}U_2$
$A_1 \sim \text{d} + \frac{1}{r_1}U_1$
d
d
t

Figure 3.5: Age of partition P_4^X is the minimum of both age functions age(r_1) and age(r_2).

We now characterize the age functions.

Theorem 3.4.1 ***(Characterization of the age functions)*** *Consider a region observed by n sensors which generate periodic updates at rates $\mathbf{r} = (r_v : v = 1, \ldots, n)$. Without loss of generality, let $r_1 \geq r_2 \cdots \geq r_n$, then the average age function is given by*

$$\overline{\text{age}}(\mathbf{r}) = \mathbb{E}[A] = d + \int_{d}^{d+\frac{1}{r_1}} \prod_{v=1}^{n} z_v(y)\ dy, \tag{3.4}$$

where $z_v(y) = 1 - r_v(y - d)$, for $v = 1, \ldots, n$. The γ-age violation function is given by

$$\text{age}_{\gamma}(\mathbf{r}) = P(A \geq \gamma) = \begin{cases} 1, & \textit{if } 0 \leq \gamma \leq d, \\ \prod_{v=1}^{n} z_v(\gamma), & \textit{if } d < \gamma \leq d + \frac{1}{r_1}, \\ 0, & \textit{if } d + \frac{1}{r_1} < \gamma. \end{cases} \tag{3.5}$$

The proof of Theorem 3.4.1 is relegated to the Appendix.

Theorem 3.4.1 exhibits closed-form expressions of both the average age and γ-age of a region observed by multiple sensors that generate periodic updates about this region. The average age in Eq. (3.4) has two terms, d and $\int_{d}^{d+\frac{1}{r_1}} \prod_{v=1}^{n} z_v(y)\ dy$. The latter decreases as n increases, which is aligned with the fact that the larger the number of sensors periodically updating about a region, the lower the average age of the region. As n goes to infinity, the right term converges to 0, and thus the average age converges to d. This confirms that the average age of a region cannot drop below the transmission delay d no matter the number of producers updating about it.

As we will see, these age functions and coupling across sensors with overlapping coverage regions are somewhat complex, thus we will first characterize a few of their properties.

3.4.2 Properties of the age functions

The following corollary further characterizes the age functions.

Corollary 3.4.1 ***(Properties of the age functions).*** *Suppose that* $\mathbf{r} = (r_v : v = 1, \ldots, n)$*, where* $r_v = r$*, then the average age function is given by,*

$$\overline{\text{age}}(\mathbf{r}) = d + \frac{1}{n+1}\frac{1}{r}, \tag{3.6}$$

and the γ*-age violation function is given by,*

$$\text{age}_\gamma(\mathbf{r}) = \begin{cases} 1, & \textit{if } 0 \leq \gamma \leq d, \\ (1 - r(\gamma - d))^n, & \textit{if } d < \gamma \leq d + \frac{1}{r}, \\ 0, & \textit{if } d + \frac{1}{r} < \gamma. \end{cases} \tag{3.7}$$

Suppose now that $\mathbf{r}$ *is such that* $r_1 \geq r_2 \geq \ldots \geq r_n$*, and let* $\bar{\mathbf{r}} = (\bar{r}_v : v = 1, \ldots, n)$*, where* $\bar{r}_v = \bar{r} = \frac{1}{n}\sum r_v$*. We then have that*

$$\overline{\text{age}}(\bar{\mathbf{r}}) \geq \overline{\text{age}}(\mathbf{r}) \geq \overline{\text{age}}(n\bar{r}), \tag{3.8}$$

$$\text{age}_\gamma(\bar{\mathbf{r}}) \geq \text{age}_\gamma(\mathbf{r}) \geq \text{age}_\gamma(n\bar{r}). \tag{3.9}$$

The proof of this corollary is relegated to the appendix.

This corollary characterizes the decrease in the age in the number n of sensors with overlapping covered regions and a fixed update rate r. Indeed both the average/violation age functions decrease (as convex functions) to lower bounds d and 0 respectively. Naturally no matter how many sensors send updates about a location one can not reduce the average age below the delay d. The corollary also suggests that if sensors with equal update rates view the same location, it is preferable, in order to minimize the age, to replace them by a single sensor and shift the resources to it. For example, if two sensors with the same update rate $\bar{r}$ share the exact same coverage area, then the age will be minimized by having only one of the sensors send updates at twice the rate i.e., $2\bar{r}$. However in general, sensors will have only partial overlaps, making the optimization of rate allocation amongst sensors more subtle. We consider such trade-offs in the next two sections.

3.5 Sensor Selection: Weighted Coverage-Age Trade-offs

In this section, we consider a setting where sensors send/broadcast updates at the *same fixed* rate r to a centralized observer/each other. We assume that only a maximum number k out of N available sensors can be active. Based on Eq. (3.1) and for equal transmission delays d and fixed sensor update rate r, it should be clear that $k \leq N \leq \lfloor \frac{1}{rd} \rfloor$. The goal is to select a subset of sensors which realizes a good compromise between ensuring good coverage of the consumers' regions of interest and minimizing the weighted age of these covered regions. There exist multiple approaches to achieving such trade-offs. We propose and explore the one that attempts to achieve both maximal aggregated weighted coverage and minimal weighted average age of the covered regions. Before devising the algorithm that achieves this goal, we first introduce a modified average age function given by,

$$b(\mathbf{r}) = -\overline{\text{age}}(\mathbf{r}) + d + \frac{1}{r} = \frac{n}{n+1}\frac{1}{r}, \tag{3.10}$$

where $\overline{\text{age}}(\mathbf{r})$ corresponds to the age function with equal rates $\mathbf{r} = (r_v : v = 1, \ldots, n)$, where $r_v = r$, as determined in Eq. (3.6). Note that $b(\mathbf{r})$ is strictly positive, upper-bounded, concave and increasing with respect to n. We further introduce the following *"utility" function*:

$$u(X) := \sum_{i=1}^{|\mathcal{P}^X|} w(P_i^X) b(\mathbf{r}(V_i^X)). \tag{3.11}$$

We re-write the aggregated interest weighted age introduced in Definition 3.3.5 as a function of the average age. With slight abuse of notation, we let $a(X) = \sum_{i=1}^{|\mathcal{P}^X|} w(P_i^X)\overline{\text{age}}(\mathbf{r}(V_i^X))$ be the aggregated interest weighted average age for a set of sensors X. From Eq. (3.10), given that d and r are constants, it should be clear that maximizing $b(\mathbf{r})$ is equivalent to minimizing $\overline{\text{age}}(\mathbf{r})$. Therefore minimizing the weighted average age $a(X)$ is equivalent to maximizing the utility function in Eq. (3.11), i.e.,

$$\min_{X \subseteq V} a(X) \equiv \max_{X \subseteq V} u(X).$$

This transformation is useful because $u(.)$ can be shown to be a submodular set function as supported by the following result.

Theorem 3.5.1 ***(Characterization of the utility function)*** *The utility function $u(.)$ satisfies the following properties:*

(*Monotonicity)* *It is monotonically increasing, i.e., if $X \subset Y \subset V$ then, $u(X) \leq u(Y)$.*

(*Submodularity*) *It is submodular, i.e., if $X \subset Y \subset V$ and $v \notin Y$ then,*

$$u(X \cup \{v\}) - u(X) \geq u(Y \cup \{v\}) - u(Y). \tag{3.12}$$

The proof of this theorem can be found in the appendix.

We now propose Algorithm 1 that prioritizes the selection of the subset of producers that *first* maximizes the overall weighted coverage motivated by safety concerns, and then *second* maximizes the utility to further enhance the timeliness of the generated updates. The order in which the objectives are maximized is important because the target is to first meet the overall consumers' coverage demand and then to provide them with frequent updates about their interest regions.

Algorithm 1: Selecting sensors in V that maximize the weighted coverage of the consumers' regions of interest and minimize the weighted age of the covered regions.

```
1  $S_0 = \emptyset$
2  $t = 0$
3  while $|S_t| < k$ do
4      $V' = \{v' \in \arg\max_{v \in V \setminus S_t} (w(C(S_t \cup v)) - w(C(S_t)))\}$
5      $v'' \in \arg\max_{v' \in V'} (u(S_t \cup v') - u(S_t))$
6      $S_{t+1} = S_t \bigcup \{v''\}$
7      if $S_{t+1} = S_t$ then
8          break
9      end
10     $t \leftarrow t + 1;$
11 end
12 return $S_{t+1}$
```

As can be seen the algorithm begins with an empty set of selected sensors and proceeds by iteratively selecting producers. In each iteration, it first greedily selects the subset of sensors V' that achieve a maximal increase in weighted coverage (Line 4) then out of the sensors in V', it selects that which further maximizes the overall utility $u(.)$ (Line 5). If the maximal weighted coverage is achieved and the total number of selected producers is less than k, say $k' \leq k$, the algorithm proceeds to select $k - k'$ producers that will further improve on the utility function (Line

5). We thus distinguish two phases for the algorithm. In Phase 1 it greedily selects $k' \leq k$ sensors so as to maximize the weighted coverage while breaking ties by maximizing the utility function. In Phase 2, once the maximal weighted coverage is achieved and if $k' < k$, the focus switches to solely selecting $k - k'$ sensors to further maximize the utility function (Line 5). We point out that if the consumers' aggregated interest measure is uniform, the weighted coverage maximization phase simplifies to a coverage maximization problem. We further define the following notation.

- $S_{k'}$: Denotes the set of sensors of size k' selected by Algorithm 1 in phase 1.
- $S^*_{k'}$: Denotes the optimal set of sensors of size k' achieving a maximal weighted coverage and maximal utility.
- $S_k(S_{k'})$: Denotes a set of sensors of size k returned by Algorithm 1 by the end of phase 2 given that k' sensors were selected in phase 1. For simplicity we will write $S_k(S_{k'})$ as S_k.
- $S^*_k(S_{k'})$: Denotes the optimal set of sensors that achieves maximal utility given that k' sensors were already selected by Algorithm 1 in phase 1. For simplicity we will write $S^*_k(S_{k'})$ as S^*_k.
- $\hat{a}(X)$: Denotes the normalized weighted average age of the region covered by sensors X, i.e., $\hat{a}(X) := \frac{1}{w(C(X))}a(X)$.

This proposed algorithm is guaranteed to achieve both good weighted coverage and normalized overall consumer interest weighted age. The following theorem formalizes these performance guarantees.

Theorem 3.5.2 ***(Performance guarantees for Algorithm 1)*** *Given a set of sensors V of size $N \geq k$, and equal update rate r per sensor, Algorithm 1 satisfies the following lower and upper bounds on both the weighted coverage and the normalized weighted average age respectively,*

$$w(C(S_k)) \geq \left(1 - \frac{1}{e}\right) w(C(S_k^*)),$$

and

$$\begin{aligned} \hat{a}(S_k) \leq & \left(1 - \frac{1}{e}\right) \hat{a}(S_k^*) \\ & + \frac{1}{e}\left[\left(1 - \frac{1}{e}\right) \hat{a}(S_{k'}^*) + \frac{1}{e}\left(d + \frac{1}{r}\right)\right] \\ & - \frac{1}{e}\lambda\left[w(C(S_k)) - \left(1 - \frac{1}{e}\right) w(C(S_k^*))\right]. \end{aligned}$$

where e is the base of the natural logarithm. It should be clear that $w(C(S_k)) = w(C(S_{k'}))$ and $w(C(S_k^*)) = w(C(S_{k'}^*))$, given that maximal coverage has been reached with k' sensors. We provide a detailed proof of this theorem in the appendix.

The derived upper bound on $\hat{a}(S_k)$ is closely tied to the coverage performance (through the third term on the right). Achieving a good coverage by selecting $k' < k$ producers leaves room to selecting $k - k'$ producers in Phase 2 of the algorithm to further improve the weighted age of the covered regions. On the other hand, if maximal coverage was not achieved after selecting k producers, then a high age of the covered regions is expected, given that the algorithm did not reach Phase

2 and therefore did not select producers to solely improve the age of the covered regions.

3.6 Optimization of Sensor Update Rates

In this section, we consider a setting where the set of active sensors, say without loss of generality V is fixed, but their update rates $\mathbf{r} = (r_v : v \in V)$ can be jointly optimized subject to the communication constraint Eq. (3.1). Given a fixed set of sensors V, we let the weighted average age of the covered set $C(V)$ be,

$$a(V, \mathbf{r}(V)) \quad := \quad \sum_{i=1}^{|\mathcal{P}^V|} w(P_i^V)\overline{\text{age}}(\mathbf{r}(V_i^V)). \tag{3.13}$$

as introduced in Definition 3.3.5. Note that this section will focus on the case where age $= \overline{\text{age}}$, i.e., on selecting the sensor update rates so as to minimize the weighted age which can be formally stated as follows:

Problem 3.6.1 ***(Age minimization)***

$$\min_{\boldsymbol{r}}\{a(V, \mathbf{r}(V)) \mid \mathbf{r} \geq 0, \sum_{v \in V} dr_v \leq 1\}. \tag{3.14}$$

In the producer-consumer setting, solving Problem 3.6.1 is equivalent to having a third party optimally redistribute the resources amongst the selected producers subject to communication constraints, in an attempt to minimize the overall consumer interest weighted average age of these regions.

Proposition 3.6.1 ***(Age minimization for sensors with disjoint coverage)*** *Suppose the sensor coverage sets $(C_v, v \in V)$ are disjoint then the age minimization*

Problem 3.6.1 is convex and reduces to,

$$\min_{\boldsymbol{r}}\{\sum_{v\in V} w(C_v)(d+\frac{1}{2}\frac{1}{r_v}) \mid \mathbf{r}\geq 0, \sum_{v\in V} dr_v \leq 1\},$$

whose optimal joint update rates $\mathbf{r}^*$ *are given by*

$$r_v^* = \frac{\sqrt{w(C_v)}}{\sum_{u\in V}\sqrt{w(C_u)}} \times \frac{1}{d}.$$

The proof of this proposition follows from standard convex optimization tools and so is left out. The solution reveals the first basic insight that for sensors with disjoint coverage sets, the age minimizing rate allocation is proportional to the *square-root* of the weight (e.g., area) of the coverage set each sensor is tracking. Thus sensors covering disjoint regions with equal weights would lead to equal update rate allocations. The general case where sensors have overlapping coverage sets is more complex.

Proposition 3.6.2 ***(Characterization of the age minimization problem)*** *For the general age minimization Problem 3.6.1 where coverage sets may overlap, the objective function given in Eq. (3.13) is a weighted sum of a convex function and a non-convex/non-concave function, and hence belongs to the family of non-convex/non-concave functions.*

It is easy to see this by noting that the average age of a partition as given in Eq. (3.4) can be re-written as,

$$\begin{aligned}\overline{\text{age}}(\mathbf{r}) &= d + \int_d^{d+\frac{1}{r_1}} \prod_{v=1}^{n} z_v(y)\ dy \\ &= f(\mathbf{r}) + g(\mathbf{r}),\end{aligned}$$

where $f(\mathbf{r}) = d + \frac{1}{2r_1}$ and where

$$g(\mathbf{r}) = -\int_{d}^{d+\frac{1}{r_1}} z_1(x)\left[1 - \prod_{i=2}^{n} z_i(x)\right] dx.$$

It is clear that $f(\mathbf{r})$ is convex in r_1, while $g(\mathbf{r})$ is non-convex/non-concave in $\mathbf{r}$ which can be proved by finding the Hessian of the function $g(.)$ with respect to $\mathbf{r}$, $H \in \mathbb{R}^{n\times n}$, and either showing that H has a mix of positive and negative eigenvalues, or that $\mathbf{y}^T H\mathbf{y},$ for all $\mathbf{y} \in \mathbb{R}^{n\times 1}$ can either be positive or negative. We can then show that $\overline{\text{age}}(\mathbf{r})$ is a non-convex function, and since it is part of the objective function in Eq. (3.13), then the latter belongs as well to the same family of functions. It should be clear by now that in the case of a single sensor v observing a partition and updating at a rate r_v, the age of this partition is convex in r_v and given by $d + \frac{1}{2}\frac{1}{r_v}$. But whenever more than one sensor are observing the same partition, the function capturing the age of this partition belongs to the family of non-convex/non-concave functions. There exists a family of algorithms that addresses this type of optimization problems, from which we pick the Frank-Wolfe algorithm [33–35], described in Algorithm 2.

We summarize the FW algorithm for both the cases of convex and non-convex objective functions. In [33], under the assumption of a convex and continuously differentiable function, and a compact convex domain $\mathcal{D}$, the algorithm computes at each iteration t the maximal step it can take in the direction of the gradient of the function while satisfying the constraint $\mathbf{s} \in \mathcal{D}$, and then moves in the direction of this maximizer. This process, as explained in [33], intuitively makes sense since the algorithm finds the direction in which it can maximize the improvement in the function value while remaining feasible. Additionally, one key advantage

Algorithm 2: Frank-Wolfe Algorithm (with adaptive step sizes) [34]

```
Let $\mathbf{r}^{(0)} = (r_v, v \in \mathcal{V}) \in \mathcal{D}$
Let $\mathcal{D} = \{\sum_{v\in\mathcal{V}} dr_v \leq 1\}$
for $t = 0, \ldots, T$ do
    Compute $\mathbf{s}^{(t)} := \arg\min_{\mathbf{s}\in\mathcal{D}} \langle\, \mathbf{s}, \nabla a(\mathbf{r}^{(t)}) \rangle$
    Let $\mathbf{d}_t := \mathbf{s}^{(t)} - \mathbf{r}^{(t)}$
    Compute $g_t := \langle \mathbf{d}_t, -\nabla a(\mathbf{r}^{(t)}) \rangle$
    if $g_t \leq \epsilon$ then return $\mathbf{r}^{(t)}$
    Line-search: $\gamma_t \in \arg\min_{\gamma\in[0,1]} a(\mathbf{r}^{(t)} + \gamma \mathbf{d}_t)$
    Update $\mathbf{r}^{(t+1)} := \mathbf{r}^{(t)} + \gamma_t \mathbf{d}_t$
return $\mathbf{r}^{(T)}$
```

of this algorithm is that it does not need to project back into the constraint set, given that it never leaves it. On the other hand, Theorem 1 in [34] gives a simple proof that the Frank-Wolfe algorithm obtains a stationary point at a rate of $\mathcal{O}\left(1/\sqrt{t}\right)$ on non-convex objectives with a Lipschitz-continuous gradient. We refer the reader to [34] for more details on the convergence of FW on non-convex objectives.

3.7 Numerical Results

We develop a simulation framework to explore the optimization of coverage vs. normalized interest weighted average age trade-offs in a collaborative sensing application and present some results in this section.

3.7.1 Model

We shall present results for two representative road scenarios. In the first one, we consider a one-way highway which we model as a rectangular region R of length 500m and width 20m (which corresponds to 5 lanes). In the second one we consider a road intersection which we present later in this section. Vehicles are modeled as $4.8 \times 1.8\text{m}^2$ rectangles with omnidirectional sensors placed at the vehicle's center (rooftop). The unobstructed coverage set for each sensor is a disc of radius $r = 50\text{m}$ with area πr^2. The coverage area of a sensor does not include the regions off the road, and the only obstructions present are those associated with vehicles blocking each others' field-of-view. We assume that vehicles are randomly placed in lanes, with spacings of at least 10m between any two vehicles in the same lane. We assume vehicles (consumers) have an interest in specific locations which might be obstructed. We refer to these as anchor points. Specifically, we will define the consumers' interests as associated with particular anchor points on the highway. Our simulation results represent averages over randomly generated sensor locations. We ran 100 Monte-Carlo (MC) simulations and plotted the mean weighted coverage and weighted age as well as the confidence intervals corresponding to the standard deviation of the estimator resulting from the MC simulations.

3.7.2 Communication model

We assume that each vehicle/sensor is equipped with a 360° camera which samples at 30 frames per second with a corresponding data rate of 1 Mbyte/sec. Additionally, V2X technology is used to share sensor updates. We assume that a single producer accesses the medium at a time and broadcasts its update to all the consumers in the system. For simplicity we assume an operational bandwidth achieving a data rate of 6 Mbps, which results in a transmission delay of 44.44 msec per image frame. We combine both the channel access time (~ 20 msec) and update transmission time into a single deterministic value, d, and find $d = 64.44$ msec. We further assume no transmission failures so broadcasts are reliable.

3.7.3 Coverage and normalized interest weighted average age of the anchor points

We define the coverage of a typical consumer v as the percentage of v's anchor points that are covered. We say an anchor point is covered if either v directly sees it or if it receives updates about this anchor point from active producers that directly see it. The *normalized interest weighted average age of a typical consumer* v is the weighted average age of v's covered anchor points, normalized by their weighted coverage. We note that we will be interested in the coverage of a typical consumer and not in the weighted coverage. By contrast we use the weights when considering the weighted age.

3.7.4 Comparison of algorithms

We assume that there are N available sensors in R, all of them acting as both consumers and producers of information. We will evaluate the performance of three main algorithms.

- The *baseline* selects *all* N sensors to act as both producers and consumers of information, each of which has the same (possibly low) update rate to meet the capacity constraint. One would expect that this technique achieves the best coverage but performs poorly in terms of the normalized interest weighted average age of a typical consumer.

- The *sensor selection* algorithm selects $k \leq N$ producers, allowing for a higher update rate per sensor and hence fresher and more frequent updates for consumers. We point out that all N sensors are consumers of information.

- The *age minimization* algorithm optimally allocates the update rates amongst $k \leq N$ selected producers to further minimize the interest weighted average age of the covered regions.

We define and make use of the following three notions of aggregated consumers' interest measures.

- The *Uniform Discrete (UD) aggregated consumer interest measure* places equal weight on a set of discrete anchor points to reflect the same level of interest of consumers in those locations.

- The *General Discrete (GD) aggregated consumer interest measure* places unequal weights on a set of discrete anchor points to reflect different levels of consumer interest in those locations.

- The *Uniform Continuous (UC) aggregated consumer interest measure* corresponds to an interest weight corresponding to the area measure on the region R.

The UD sensor selection algorithm selects a set of producers that cover the consumers' anchor points which further keeps their interest weighted average age low. The GD sensor selection algorithm selects producers that cover the consumers' anchor points, given different weights on the anchor points. This algorithm exhibits the advantage of having consumers indicate their degrees of interest in locations by assigning weights proportional to their interest. It should be clear that when all the anchor points have an equal weight, the GD sensor selection algorithm reduces to the UD sensor selection algorithm. Finally, the UC sensor selection algorithm chooses a set of producers so as to cover as much as possible of the overall region R, while trying to keep the average age of the covered regions as low as possible, given a uniform consumer interest across the whole region R. This can only be achieved by judiciously optimizing producers' coverage overlaps. This existing tension between spreading producers across space and overlapping their coverage sets is what makes this optimization challenging. With such a weight measure, the algorithm selects the producers independently of the consumers' preferences, i.e., it is not consumer oriented, but it provides a choice of producers which is robust across possible consumer interests.

3.7.5 Highway Scenario

We randomly generate 5 anchor points on the highway and assign them equal weight to express an equal consumer interest in each.

We verify in our simulation results the robustness of the UD sensor selection algorithm in achieving a good coverage-age trade-off per typical consumer.

3.7.5.1 Effect of increasing the number of selected producers by UD and UC sensor selection algorithms on both age and coverage

In Figure 3.6, we fix the number of available sensors in R at $N = 70$, and plot the normalized interest weighted average age of the covered anchor points of a typical

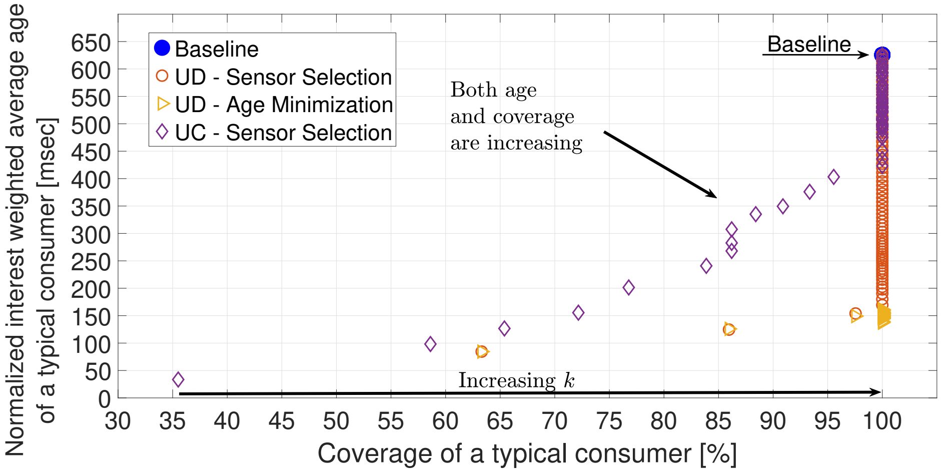


Figure 3.6: Normalized interest weighted average age vs. coverage of a typical consumer with anchor points having equal weight, for a fixed number of available sensor $N = 70$ and an increasing number of selected producers k by UD and UC algorithms from 1 to 70.

consumer versus the coverage of a typical consumer, as defined earlier. The baseline (blue dot at 100% coverage and 630 msec) achieves the highest coverage per typical consumer but also the largest age, as expected. For both UD and UC sensor selection algorithms, we increase the number of selected producers k from 1 to N which results in an increase in both the coverage and the age of a typical consumer until they saturate at the same coverage and age achieved by the baseline. We clearly see that for all k, the UD sensor selection algorithm (red circles) achieves an overall better coverage and age compared to the UC sensor selection algorithm (purple diamonds), with a remarkable 23% larger coverage and an age improvement of around 40 msec for $k = 5$. As should be expected, matching producers with consumers based on their demands achieves both good coverage and age, even when a small number of producers are selected. Optimizing the update rate allocation amongst the selected producers in the UD sensor selection algorithm (yellow triangles) further minimizes the age, which saturates at around 150 msec for all $k \geq 4$.

3.7.5.2 Advantage of using a consumer-oriented scheme when a fixed small number of producers is selected

We now consider the case where the number of available sensors N on the highway increases from 5 to 70. We plot the coverage and normalized interest weighted age of a typical consumer in Figures 3.7 and 3.8 respectively.

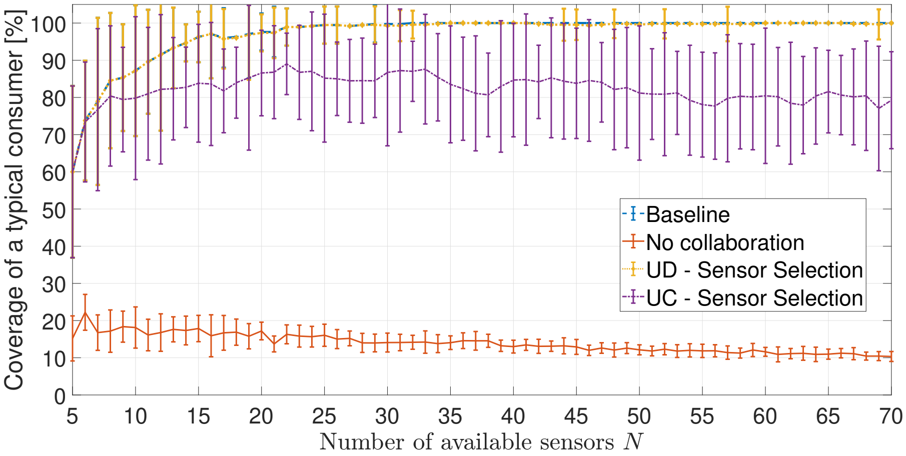


Figure 3.7: Coverage of a typical consumer with anchor points having equal weight, when there is an increasing number of available sensors N but a fixed number of selected producers $k = 5$ by both UD and UC algorithms.

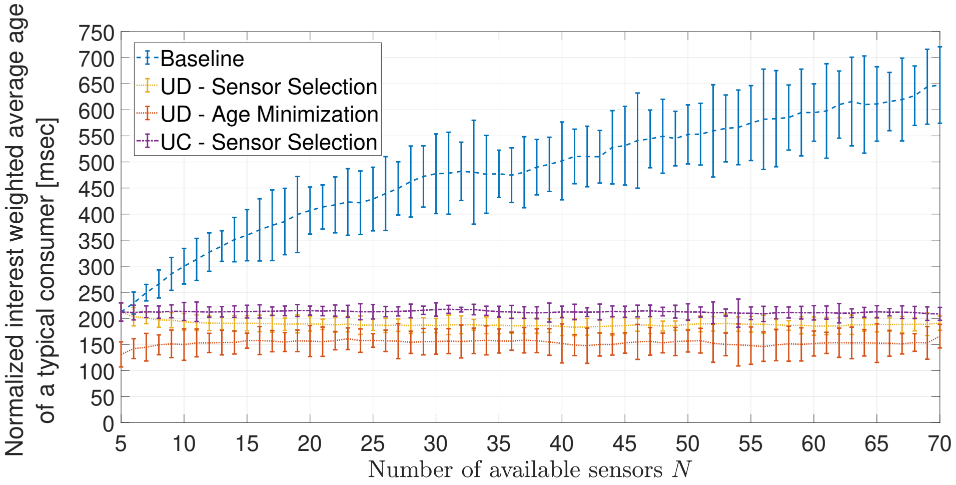


Figure 3.8: Normalized interest weighted average age of a typical consumer with anchor points having equal weight, when there is an increasing number of available sensors N but a fixed number of selected producers $k = 5$ by both UD and UC algorithms.

Without any collaboration, i.e., without any information sharing between the sensors, the coverage of a typical consumer is the lowest at around 20%, and decreases due to increased obstructions, while being the largest when all sensors are collaborating. We assume that UD and UC sensor selection algorithms select only $k = 5$ producers. In Figs. 3.7 and 3.8, we clearly see that the UD algorithm achieves both maximal coverage and minimal age (yellow dotted curves) as compared to the other algorithms. And with further optimization of the update rates among the selected producers, the age is further minimized (dotted red curve in Fig. 3.8). An interesting observation in Fig. 3.7 is the increase in the coverage achieved by the UC algorithm (purple curve) until $N = 22$, followed by a constant decrease, due to the fact that there is a larger set of available producers to select from, and hence this

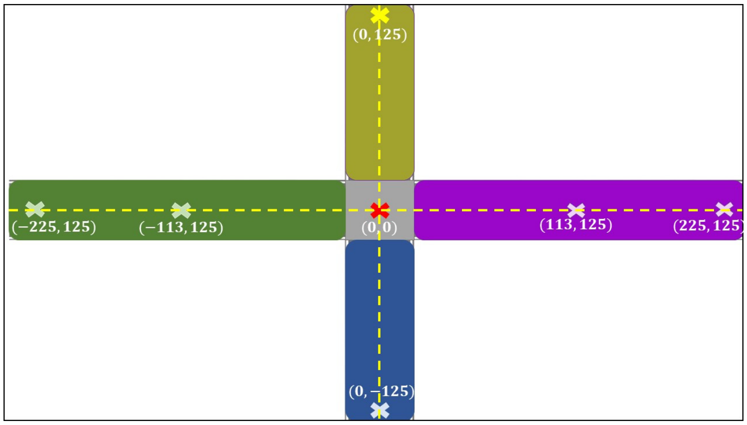
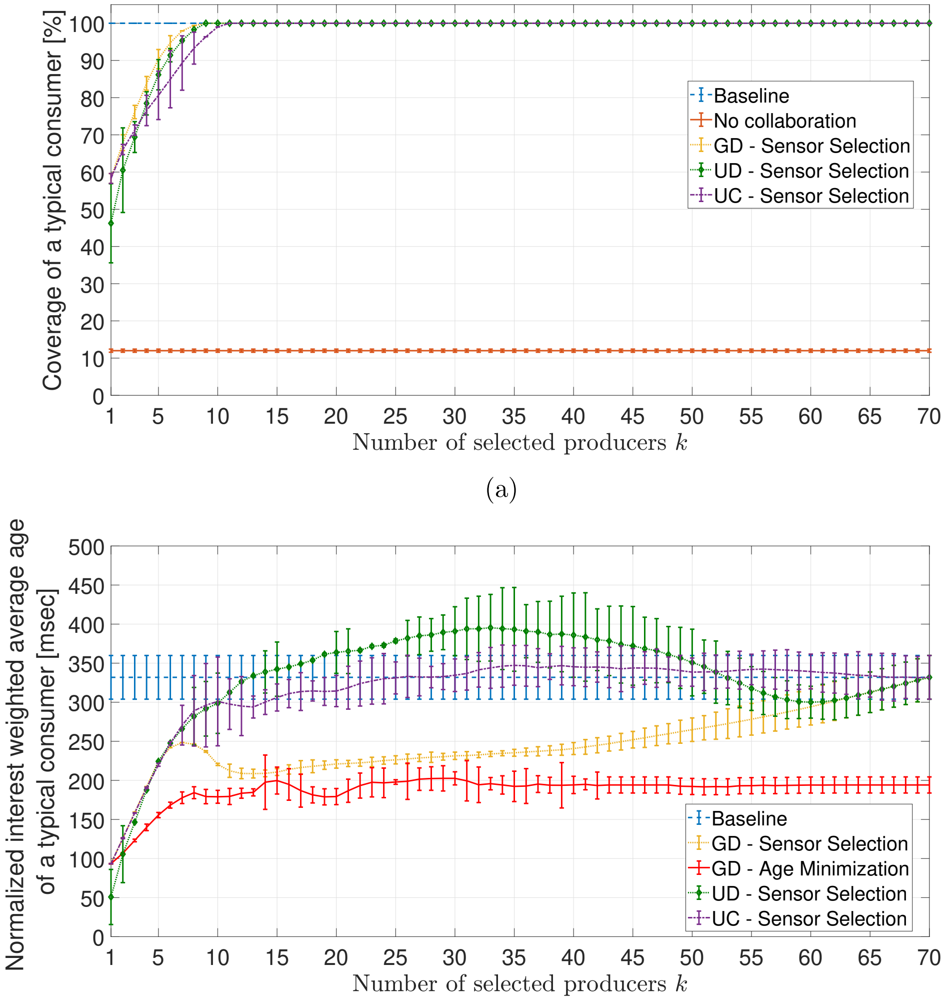


Figure 3.9: Road intersection with 7 pre-set anchor points.

scheme will focus on maximizing the coverage of the highway and unintentionally shift away from covering the consumers' anchor points.

3.7.6 Road Intersection Scenario

In the highway scenario, we assumed that all anchor points had an equal weight. We now consider a higher spatial correlation between the interest locations of different consumers reflected through different weights assigned to different locations. This results in a clear improvement in the age of the consumers' covered anchor points when the algorithm that selects the producers is oriented towards meeting the consumers' interest in timely updates. We therefore consider a second scenario which consists of a road intersection as depicted in Fig. 3.9. The horizontal two-way road has a width of 40m (corresponding to 10 lanes) and length of 450m, the vertical road has a width of 16m (4 lanes) and length equal to 250m. There are 7 pre-set anchor points in total. A consumer is interested in the anchors that fall within its own colored section and in the anchor at the intersection, as can be

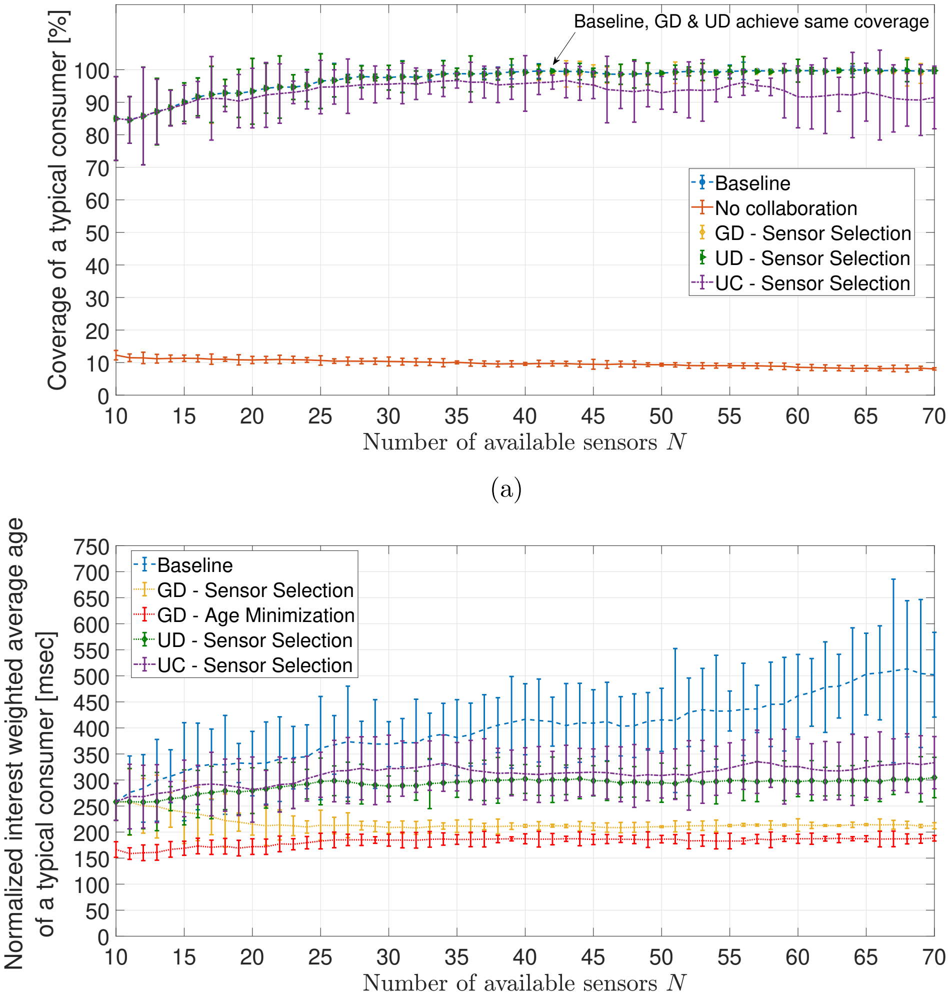


(a)

(b)

Figure 3.10: Coverage and normalized interest weighted average age of a typical consumer with anchor points having unequal weights, when there is a fixed number of available sensors $N = 70$ but an increasing number of selected producers k by GD, UD and UC algorithms from 1 to 70.

seen in Fig. 3.9. For example, vehicles falling in the green region express interest in the two green anchor points with coordinates $(-225, 125)$ and $(-113, 125)$ and in



(a)

(b)

Figure 3.11: Coverage and normalized interest weighted average age of a typical consumer with anchor points having unequal weights, when there is an increasing number of available sensors N from 10 to 70 but a fixed number of selected producers $k = 10$ by GD, UD and UC algorithms.

the red one falling at the origin. A consumer assigns a weight equal to 5 to the red anchor point at the intersection versus a weight of 1 to any other anchor point in its own section, which expresses the consumers' higher urge for updates about the intersection.

We once again assume there are N randomly placed vehicles equipped with 360^o cameras on any of the highways' lanes with a minimum distance of at least 10m between them.

We mainly compare the GD and UD sensor selection algorithms. Once producers are selected, we evaluate the coverage and normalized interest weighted average age of a typical consumer, assuming the aggregated interest measures are assigned according to the consumers' interests.

3.7.6.1 Effect of increasing the number of selected producers by GD, UD and UC sensor selection algorithms on both age and coverage

In Figure 3.10, we assume there are $N = 70$ available sensors. We increase the number of selected producers k from 1 to 70 for all of GD, UD and UC sensor selection algorithms. We see that the GD sensor selection algorithm achieves both a better coverage than all the others algorithms for $k \leq 6$ (Fig. 3.10a) and a major age improvement for $6 \leq k \leq 60$ (Fig. 3.10b). The clear drop in age achieved by the GD algorithm for $6 \leq k \leq 15$ is the result of selecting the sensors that solely minimize the age once maximal coverage is achieved. We also clearly see that both the UD and UC algorithms achieve a worse age than the one achieved by the baseline, and this occurs after the number of selected producers k achieves maximal coverage, which confirms that these two schemes are not consumer oriented. Finally, an interesting observation is that for $N = 70$, optimizing the rates among the 70 selected producers reduces the age by ~ 140 msec.

3.7.6.2 Advantage of using a consumer-oriented scheme when a fixed small number of producers is selected

Finally, in Fig. 3.11, we increase the number of available sensors from 10 to 70. GD, UD and UC sensor selection algorithms only select 10 producers at all time. As expected, the coverage achieved by all algorithms (except for the no-collaboration algorithm) are somewhat similar and maximal, with a clear age improvement achieved by the GD algorithm. With further optimization of the producers' update rates, the age achieved by this algorithm is itself further improved by around 30 msec for $N = 70$.

3.8 Chapter Conclusion and Future Work

In this chapter we have exhibited an approach to achieving trade-offs between coverage and timeliness in communication constrained collaborative sensing settings wherein spatially distributed sensor nodes can serve dual roles as producers and consumers of sensed information. The proposed framework allows quite a bit of flexibility in terms of capturing the underlying characteristics of information sharing, and suggests the development of a possible market place for sharing real-time sensor data in a context dependent manner, i.e., matching nodes' current interest, in order to minimize situational uncertainty so as to enhance vehicular flow and safety. A key aspect is the design of strategies to match consumers spatial interest in timely information to producers' overlapping coverage, subject to communication network capacity constraints. A key part of our future work is to make the sensor selection and update rate optimization more scalable to start addressing heterogeneous and dynamic environments.

Chapter 4

Scheduling “Last-Minute” Updates for Timely Decision-Making

The emergence of applications relying on networked systems has revolutionized the sensing industry and led the way towards modeling systems that rely on the timely sharing of information to support real-time decision making. Among these, a challenging set of examples is tied to automated vehicles, robots, UAVs, etc., that are constantly traveling through complex environments and requiring updates to smoothly navigate with a high degree of situational awareness. Such applications are often best supported when updates are delivered right on time. In the vehicular setting for example, cars driving at different speeds and heading towards an obstructed intersection may express interest in accurate information on the state of the intersection right before they reach it and thus generate requests for timely updates about such intersections. Sending an update to a vehicle early on may not accurately represent the state of the intersection by the time it gets there and will lead to poor decisions. On the other hand, scheduling an update transmission to the vehicle when it is close to the intersection is likely to be advantageous and results in better decisions.

A major challenge in such systems is the dynamic aspect of requests for timely updates. Requests may not only arrive arbitrarily but may also be short-lived, i.e., such that one can receive updates only for a finite length time window before making a critical decision at the end of the window preferably based on the freshest information update.

A key step in this direction is to define appropriate metrics that capture the freshness and timeliness of the last received update to ensure that it accurately represents the state of the time-varying process that the request is interested in. To that end, the Age of Information (AoI) has been proposed as a metric that measures the freshness of the updates at the receiver [7,24,25,27]. In contrast to prior work on AoI, we propose a novel setting where a request can only receive updates within a finite time window, which we refer to as the request's active window. The importance of scheduling an update transmission to a request close to the end of its active window is modelled through a reward function which depends on the time difference between the end of the request's active window and the time at which the last update was received. Such a reward model is therefore tied to the age of the last received update within the request's active window which reflects the freshness of this update. In this chapter [1] , we study scheduling policies that aim to optimize the rewards associated with scheduling such updates.

4.1 Related work

There has been substantial work on the scheduling of requests with deadlines. The Earliest Deadline First (EDF) policy [36] is the most well-known policy

[1] Part of the results in this chapter are submitted to *ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'23).*

for scheduling in real-time systems. It was proved to maximize the fraction of customers served prior to their respective deadlines when the service time is equal to a single slot. EDF requires that the customer with the earliest deadline be scheduled first and at most once. Meanwhile [37] considers a real time queuing system where packets have deadlines and the processing time of a packet is known upon its arrival. A predetermined fixed reward is associated with servicing each packet and the goal is hence to design a scheduling policy that maximizes the cumulative reward. Other works, e.g., [38], [39], introduce scheduling policies in systems with strict bounds on the service delays. In contrast to prior work, we allow a request to be scheduled more than once before the end of its active window, and we define a reward function that is tied to the time difference between the request's decision time and its last scheduled update.

More recent work addresses scheduling in collaborative sensing settings in an attempt to achieve real-time situational awareness and can be found in [1, 2, 40–42]. We extend on the prior works by assuming that requests for timely updates arrive arbitrarily and can only receive updates within a limited period of time before making a critical decision. Other recent work investigate scheduling sensing nodes to update a remote node under communication constraints with requirements on the AoI [43–46]. They consider applications where the AoI has to meet some freshness threshold, i.e., they impose either a hard or soft upper bound on the worst case AoI that can be achieved by any sensing node and devise policies that can schedule at most one node per slot to satisfy the constraints. On the other hand, [47] and [48] consider the setting where packets arrive arbitrarily over time and the algorithms only have access to information about packet arrivals. In particular, [47] devises a policy to minimize the energy consumption under the peak AoI constraint at all times, while [48] introduces a resource allocation problem that captures the trade-off between AoI, quality and energy associated with packet transmission and proposed a policy to minimize the three costs. Finally, [49] develops and implements a scheduling algorithm that enables the customization of WiFi networks to the needs of time-sensitive applications. They propose a scheduling approach which makes use of the most up-to-date data as opposed to all past sampled data points, similarly to the one suggested in our work. We differentiate ourselves from [49] by considering a setting where requests arrive arbitrarily over time, as opposed to having a fixed number of nodes requesting information, and additionally consider that the requests can receive updates only in a short period of time before making decisions based on the last received update.

4.2 Chapter Contributions

We explore a new class of scheduling problems associated with delivering information updates “just in time”. We consider a setting where requests for timely updates arrive arbitrarily and we assume that requests can receive updates within a finite-time window, which we refer to as the request’s active window. Our goal is to ensure that requests have updates that are as fresh as possible by the end of their active windows to enable accurate decisions to be made based on the states of the time-varying processes they are interested in. We hence define a reward function that captures the importance of scheduling an update transmission to a request as close to the end of its active window as possible. We propose to maximize the reward-rate under the assumption that only a single request can be scheduled at a time. In this setting, we investigate an adversarial setting where the number of new requests' arrivals as well as the length of a request's active window are unknown a-priori and only revealed once requests arrive, and thus use the competitive ratio as our performance metric. We derive a lower bound on the reward-rate achieved by any non-idling causal policy. We then propose two causal scheduling policies π^a and π^g, referred to as the adversarial and greedy policies respectively and further derive the competitive ratio of π^g with respect to the optimal genie-based policy. Finally, we validate our theoretical analysis with numerical evaluations.

4.3 System Model

4.3.1 Model for timely information requests

Consider requests for timely updates about time-varying processes that arrive arbitrarily to a time-slotted system. We let $\boldsymbol{\rho} = (\rho_i)_{i \in \mathbb{N}}$ denote the sequence of request arrivals, where the tuple $\rho_i = (a_i, s_i, e_i)$ is the i^{th} request, characterized by,

- a_i: Arrival time of request i,
- $s_i \geq a_i$: Release (or start) time of request i, which reflects the earliest time after which updates about the time-varying process become relevant to i,
- e_i: End time after which request i is no longer active,
- $[s_i, e_i]$: Active window within which updates in response to request i are permissible.

In particular, we let $\boldsymbol{\rho}_T$ denote the truncated sequence of requests that have end times prior to T. Updates scheduled for request i outside of $[s_i, e_i]$ have no value to i. We point out that one or more updates for a request can be scheduled while it is active. However, requests end up using only the most recent update they received within their active windows. It follows that an update scheduled closer to the start to service time of a request may age by the end of the active window and become stale and not accurately reflect the actual status of the time-varying process. Therefore, scheduling an update for active request i in a slot close to e_i is more beneficial than scheduling an update in a slot close to s_i. We further let $w_i = e_i - s_i + 1$ be the length of request i's active window and $w_{\max}$ be an upper bound on the length of the active window, i.e., for all i we have that $1 \leq w_i \leq w_{\max}$.

4.3.2 Scheduling updates

We consider a time-slotted system where the length of a time slot corresponds to the duration it takes to transmit an update. Further, we consider for simplicity a setting where a policy can schedule a single update per time slot. That said, recall that multiple updates can be scheduled sequentially for the same request within its active window. Below, we formally introduce the notation to be used in this chapter.

Definition 4.3.1 ***(Servicing a request)*** *We say that a policy π has serviced a request i if i is no longer active and π scheduled one or more updates for i within its active window $[s_i, e_i]$.*

We shall use the following notation.

- $(N_t)_{t\geq 1}$, where $N_t := \{i : i \in \mathbb{N}\ ,\ a_i = t\}$, is the set of new requests arriving at the beginning of slot t.

- Q_t is the set of active requests in slot t.

- $x^{\pi}_{i,t}$ is an indicator variable that takes value 1 if an update for an active request i is scheduled in slot t under a policy π and 0 otherwise.

- $T^{\pi}_{i,t} := \{\tau : s_i \leq \tau \leq t \leq e_i\ ,\ x^{\pi}_{i,\tau} = 1\}$ is the set of time slots in which updates for active request i are scheduled prior to and including t under a policy π.

- $H_t := \{i : i \in \mathbb{N}\ ,\ e_i \leq t - 1\}$ is the set of requests which are no longer active at t.

- $S^{\pi}_t := \{i : i \in H_t\ ,\ T^{\pi}_{i,e_i} \neq \emptyset\}$ is the set of requests in H_t that have been serviced by policy π.

4.3.3 Reward and age of last update

We let r^{π}_i denote the reward obtained for a request i that was serviced under policy π. It depends on the last slot in which an update for i was scheduled when it was active. Assume the last update for an active request i is scheduled in a slot close to its start time s_i, then the "age" of the update would increase by the end of the active window and the update would not provide timely information to i as would an update scheduled in a slot close to e_i. Therefore scheduling an update for request i in a slot closer to its end time e_i is deemed advantageous. We let d^{π}_i denote the time difference between the end time of active request i and the last slot in which an update is scheduled for i under policy π, i.e., $d^{\pi}_i = e_i - \max\limits_{t\in[s_i,e_i]} x^{\pi}_{i,t}t + 1$.

Definition 4.3.2 ***(Reward obtained from servicing*** i***)*** *The reward* r_i^π *obtained from servicing request* i *under a policy* π *depends on the last slot in which an update for* i *was scheduled and is collected when* i *is no longer active, i.e., at the end of slot* e_i*. It is modelled as,*

$$r_i^\pi = \begin{cases} f(d_i^\pi), & \text{if } T_{i,e_i}^\pi \neq \emptyset, \\ 0, & \text{otherwise}, \end{cases} \tag{4.1}$$

where $f(\cdot)$ *is a non-decreasing upper-bounded function of* d_i^π*.*

Scheduling an update for an active request i under a policy π in a slot close to its end time e_i results in a smaller $e_i - \max_{t\in[s_i,e_i]} x_{i,t}^\pi t$ and thus in a larger reward r_i^π. The cumulative reward at slot t under policy π is denoted by,

$$r^\pi(S_t^\pi) = \sum_{i\in S_t^\pi} r_i^\pi. \tag{4.2}$$

In the rest of the chapter, we consider both linear and convex reward functions $f(\cdot)$.

Definition 4.3.3 ***(Linear reward function)*** *A positive linear reward function* $f(x) = \alpha + \beta(w_{\max} - x + c)$ *associated with servicing request* i *under a policy* π *is given by,*

$$r_i^\pi = \alpha + \beta(w_{\max} - d_i^\pi + c), \tag{4.3}$$

where $\alpha, \beta, c \geq 0$*.*

The linear reward associated with servicing a request i under a policy π as defined above consists of two main components, α and $\beta(w_{\max} - d_i^\pi + c)$. The second component is bounded below and above as follows, $\forall i \in \mathbb{N}$,

$$\beta(w_{\max} - w_i + c) \leq \beta(w_{\max} - d_i^\pi + c) \leq \beta(w_{\max} + c).$$

Definition 4.3.4 ***(Convex reward function)*** *A positive convex function $f(x) = \alpha + h(-\beta(x - c))$ models the reward associated with servicing a request i under a policy π if,*

$$r_i^\pi = \alpha + h(-\beta(d_i^\pi + c)), \tag{4.4}$$

where $\alpha, \beta, c \geq 0$ and $h(\cdot)$ is a positive convex function.

Paralleling the way we defined the linear reward, the convex reward has two main components, α and $h(-\beta(d_i^\pi + c))$, where the second term is bounded above and below as follows, $\forall i \in \mathbb{N}$,

$$h(-\beta(w_i + c)) \leq h(-\beta(d_i^\pi + c)) \leq h(-\beta c).$$

4.3.4 Characterization of the scheduling problem

Our objective is to maximize the reward-rate obtained from servicing a sequence of requests $\boldsymbol{\rho}_T$ in a finite time window $[1, T]$, where without loss of generality, $t = 1$ corresponds to the first slot in which there are any arrival of new requests and T is the last slot after which there are no longer any active requests. For a sequence of requests $\boldsymbol{\rho}_T$, we let $g(\pi, \boldsymbol{\rho}_T) = \frac{r^\pi(S_{T+1}^\pi)}{H_{T+1}}$ be the reward-rate obtained under policy π. Formally, the problem is defined as follows.

Problem 4.3.1

$$\max_{\pi\in\Pi} g(\pi, \boldsymbol{\rho}_T) = \max_{\pi\in\Pi} \frac{r^{\pi}(S^{\pi}_{T+1})}{|H_{T+1}|} \tag{4.5}$$

$$s.t. \qquad \sum_{i\in Q_t} x^{\pi}_{i,t} \leq 1, \forall t \in [1, T], \tag{4.6}$$

$$x^{\pi}_{i,t} \in \{0, 1\}, \forall i \in Q_t, \forall t, \tag{4.7}$$

where $|H_{T+1}|$ corresponds to the number of requests in H_{T+1}, where Equation (4.6) limits the number of users that can be scheduled at a time to at most 1, and where Π is the set of causal non-idling policies defined as follows.

Definition 4.3.5 ***(Non-idling policy)*** *A policy $\pi \in \Pi$ is said to be non-idling if it only idles when there are no active requests.*

Our goal is to design a causal non-idling scheduling policy that maximizes the reward-rate in Problem 4.3.1.

Assumption 4.3.1 *In the remainder of the chapter we consider the regime where $\alpha > \beta(w_{\max} + c)$ and $\alpha > h(-\beta c)$ for both linear and convex reward functions respectively.*

Discussion. An interesting regime for both the linear and convex reward functions in Definitions 4.3.3 and 4.3.4, is that where $\alpha > \beta(w_{\max} + c)$ and $\alpha > h(-\beta c)$. Then a large reward of value α is obtained if an update to an active request i is scheduled in any slot within its active window and in addition to a smaller reward of value $\beta(w_{\max} - d^{\pi}_i + c)$ which depends on how close to e_i is the last slot in which an update to i was scheduled. Therefore, a policy whose target is to maximize the reward-rate defined in Problem 4.3.1 is driven to first maximizing the number of scheduled requests then, if possible, to schedule these requests as close as possible to the ends of their associated update windows. In particular, if $\beta = 0$, the linear reward function in Definition 4.3.3 reduces to $r_i^\pi = \alpha$ and Problem 4.3.1 corresponds to maximizing the fraction of updates scheduled within their active windows. Hence, an optimal policy π that solves this problem when $\beta = 0$ needs to schedule at most one update per active request, in any time slot within its active window. An optimal policy for this setting is the Earliest Deadline First (EDF) policy [36]. From Definition 4.3.2, the reward obtained from servicing a request i under a policy π depends on both the last slot in which an update for i is scheduled as well as on i's release time s_i, and is hence independent of its arrival time a_i. Therefore, and without loss of generality, we assume in the rest of the chapter that a request's release time is equal to its arrival time, i.e., for all $i \in \mathbb{N}$ we have that $s_i = a_i$.

4.4 Optimal offline policy

In this section we characterize the optimal offline policy π^* that solves Problem 4.3.1.

Definition 4.4.1 ***(Optimal genie-based offline policy*** π^****)*** *A policy* π^* *is optimal for Problem 4.3.1 if it maximizes the reward-rate.*

Policy π^* has knowledge of the requests' arrivals in the entire timeline and can therefore optimally service requests. Optimal offline policies are useful since they provide an upper bound on the reward-rate that can be achieved by causal policies. π^* schedules at most a single update transmission per active request, since an active request requires at most a single update being scheduled as close as possible to its end time.

4.5 Causal scheduling policies

The limitation of causal policies Π is that they have no knowledge of future request arrivals. Our goal is to devise online causal policies which have provable performance guarantees in terms of Competitive Ratio (CR) with respect to the optimal offline policy, defined as follows.

Definition 4.5.1 ***(Competitive ratio of a policy*** π***)*** *The competitive ratio of a policy* π *is given by,*

$$\mathrm{CR}_\pi = \min_{\boldsymbol{\rho}_T \in \mathbf{P}_T} \frac{g(\pi, \boldsymbol{\rho}_T)}{g(\pi^*, \boldsymbol{\rho}_T)}, \tag{4.8}$$

where $\mathbf{P}_T$ *is the set of all possible request arrivals in a time window of length* T*,* π^* *is the optimal offline policy that solves Problem 4.3.1, and* $g(\pi, \boldsymbol{\rho}_T)$ *and* $g(\pi^*, \boldsymbol{\rho}_T)$ *are the reward-rate expressions achieved by both* π *and* π^* *respectively. An online algorithm is q-competitive for some* $q \geq 1$ *if it achieves at least* $1/q$ *of the optimal offline value in the worst case, i.e., for all* $\boldsymbol{\rho}_T \in \mathbf{P}_T$*, we have that* $g(\pi, \boldsymbol{\rho}_T) \geq \frac{1}{q} g(\pi^*, \boldsymbol{\rho}_T)$*.*

We shall begin by providing a lower bound on the competitive ratio for any causal non-idling scheduling policy in Π.

4.5.1 Lower bound on the competitive ratio of any policy in Π

The following theorem states that any causal non-idling policy $\pi \in \Pi$ achieves a competitive ratio of at least $\frac{1}{w_{\max}}$.

Theorem 4.5.1 *For any causal non-idling policy $\pi \in \Pi$, and with a reward function that satisfies the condition in Assumption 4.3.1, the competitive ratio of π satisfies*

$$\mathrm{CR}_{\pi} \geq \frac{1}{w_{\max}}.$$

Proof 4.5.1 *Consider a causal non-idling policy $\pi \in \Pi$. According to Assumption 4.3.1, the worst scheduling strategy that π can follow is to schedule the same active request every slot until it is no longer active. Let $[1, T]$ be a time window of length $T \geq w_{\max}$, where T corresponds to the slot after which there are no longer any active requests. A lower bound on the cumulative reward $r^{\pi}(S^{\pi}_{T+1})$ obtained under π is $r^{\pi}(S^{\pi}_{T+1}) \geq \frac{T}{w_{\max}} f(0)$. The cumulative reward $r^{\pi^*}(S^{\pi^*}_{T+1})$ obtained under the optimal offline policy π^* is upper bounded by $T f(0)$. Therefore, $\forall\ T \geq w_{\max}$, the competitive ratio of π is lower bounded as follows, $\mathrm{CR}_{\pi} \geq \frac{r^{\pi}(S^{\pi}_{T+1})}{r^{\pi^*}(S^{\pi^*}_{T+1})} = \frac{1}{w_{\max}}$, which concludes the proof.*

4.5.2 Greedy policy π^g

The causal greedy policy π^g presented in the Algorithm 3 panel schedules in every slot, the request that would maximize the marginal increase in cumulative reward, and if need be, breaks ties arbitrarily (Line 5).

Theorem 4.5.2 ***(π^g maximizes the ratio of serviced requests)*** *Policy π^g*

Algorithm 3: Greedy policy π^g.

```
$Q_0 \leftarrow \emptyset$; $E_0 \leftarrow \emptyset$; $S_1^{\pi^g} \leftarrow \emptyset$; $H_1 \leftarrow \emptyset$;
for $t = 1, 2, ...$ do
    $N_t :=$ set of new request arrivals;
    $Q_t \leftarrow (Q_{t-1} \setminus E_{t-1}) \bigcup N_t$;
    $i^* \in \arg\max_{i \in Q_t} f(e_i - t) - f(e_i - \max_{\tau \in [a_i, t-1]} x_{i,\tau}^{\pi^g} \tau)$; break ties arbitrarily;
    $x_{i^*,t}^{\pi^g} \leftarrow 1$; $T_{i^*,t}^{\pi^g} \leftarrow T_{i^*,t-1}^{\pi^g} \bigcup \{t\}$;
    $E_t := \{i : i \in Q_t, e_i = t\}$ ;
    $D_t^{\pi^g} := \{i : i \in E_t, T_{i,t}^{\pi^g} \neq \emptyset\}$ ;
    $S_{t+1}^{\pi^g} \leftarrow S_t^{\pi^g} \bigcup D_t^{\pi^g}$;
    $H_{t+1} \leftarrow H_t \bigcup E_t$ ;
```

maximizes the ratio of serviced requests when the linear and convex reward functions satisfy the conditions in Assumption 4.3.1.

Proof 4.5.2 *The proof of this theorem follows from the optimality of the Earliest Deadline First (EDF) policy in maximizing the ratio of serviced requests for the discrete time $G/D/1{-}G$ queue where the service time is exactly one unit of time [36]. Following from Assumption 4.3.1, π^g prioritizes scheduling requests that have not been scheduled yet. If in a time slot t there are more than one active requests that have not been scheduled prior to t, π^g, similarly to EDF, schedules in slot t an update transmission to the request with the earliest end time. Otherwise, in the case where all active requests at time t have already been scheduled prior to t, π^g schedules the request that maximizes the marginal increase in the cumulative reward, which does not affect the ratio of serviced requests. This concludes the proof.*

Theorem 4.5.3 ***(Competitive ratio of policy** π^g**)*** *The competitive ratio of π^g*

is

$$\mathrm{CR}_{\pi^g} = \frac{f(w_{\max})}{f(0)}.$$

Proof 4.5.3 *According to Theorem 4.5.2, π^g maximizes the ratio of serviced requests. Let $N = |S^{\pi^g}_{T+1}|$ be the total number of requests that have been serviced under π^g in a finite time window of length T. It follows that a lower bound on the cumulative reward $r^{\pi^g}(S^{\pi^g}_{T+1})$ obtained under π^g is $r^{\pi^g}(S^{\pi^g}_{T+1}) \geq Nf(w_{\max})$. The cumulative reward $r^{\pi^*}(S^{\pi^*}_{T+1})$ obtained under the optimal offline policy π^* is upper bounded by $Nf(0)$. Therefore, for all $T \geq w_{\max}$, the competitive ratio of π^g is lower bounded as follows, $\mathrm{CR}_{\pi^g} \geq \frac{r^{\pi^g}(S^{\pi^g}_{T+1})}{r^{\pi^*}(S^{\pi^*}_{T+1})} = \frac{f(w_{\max})}{f(0)}$, which concludes the proof.*

4.5.3 Our proposed causal scheduling heuristic policy π^a

We propose a causal scheduling policy π^a. We shall refer to it as the adversarial policy.

During every slot t, π^a assigns to every slot in the interval $[t, \max_{i \in Q_t} e_i]$ a single request that is active during this slot. We say that at time t, π^a tentatively schedules updates for active requests in slots within the interval $[t, \max_{i \in Q_t} e_i]$. By the end of slot t, an update is sent to the request scheduled in t, while the remaining slots in $(t, \max_{i \in Q_t} e_i]$ are freed from any tentative schedules.

Definition 4.5.2 ***(Tentatively scheduling updates for an active request)*** *Policy π^a tentatively schedules updates to an active request $i \in Q_t$ if during slot t it assigns slots within the interval $[t, e_i]$ to potentially transmit updates to request i in those slots.*

We clearly define additional notation specific to π^a.

- $\hat{x}^{\pi^a}_{i,t,t'}$ is an indicator that takes value 1 if at slot t, an update for active request i is tentatively scheduled in slot $t' \in [t, e_i]$.
- $\hat{T}^{\pi^a}_{i,t,t'} := \{\tau : t \leq t' \leq \tau \leq e_i \ , \ \hat{x}^{\pi^a}_{i,t,\tau} = 1\}$ is the set of time slots within the interval $[t', e_i]$ for any $t' \in [t, e_i]$, in which active request i is tentatively scheduled.

Note. As long as a request i is active, no reward is yet collected for request i. A reward associated with servicing a request i is only obtained when i is no longer active. We introduce a specific notion for the reward associated with active request i under policy π^a which we refer to as the tentative reward, defined as follows.

Definition 4.5.3 ***(A request's tentative reward under policy*** π^a***)*** *The tentative reward $\hat{r}^{\pi^a}_{i,t,t'}$ of an active request $i \in Q_t$ on slot t is a function of the slots in $T^{\pi^a}_{i,t-1}$ in which i was actually scheduled prior to t under π^a and the slots $\hat{T}^{\pi^a}_{i,t,t'}$ in which i is tentatively scheduled after $t' \in [t, e_i]$ under π^a, and is given by,*

$$\hat{r}^{\pi^a}_{i,t,t'} = \begin{cases} f\left(e_i - \max(\max_{\tau \in [a_i, t-1]} x^{\pi^a}_{i,\tau}\tau, \max_{\tau \in [t', e_i]} \hat{x}^{\pi^a}_{i,t,\tau}\tau)\right), \\ \qquad\qquad\qquad\qquad \text{if } T^{\pi^a}_{i,t-1} \cup \hat{T}^{\pi^a}_{i,t,t'} \neq \emptyset, \\ 0, \text{ otherwise.} \end{cases}$$

At any time t, an active request i may have been actually scheduled one or many times prior to but not including t, and may have been tentatively scheduled after t. Hence from Definition 4.5.3, the tentative reward depends on the latest time slot in which i is either tentatively scheduled or has been actually scheduled.

We similarly define the aggregate reward associated with a set of active requests.

Definition 4.5.4 ***(Requests' aggregate tentative reward under policy π^a)*** *The aggregate tentative reward at slot t associated with the set of active requests Q_t under policy π^a, is the sum of the requests' tentative rewards and is given by $\hat{r}^{\pi^a}_{t,t'}(Q_t) = \sum_{i \in Q_t} \hat{r}^{\pi^a}_{i,t,t'}$.*

We consider the setting where the reward obtained from servicing a request is either linear or convex as introduced in Definitions 4.3.3 and 4.3.4 respectively. π^a is presented in Algorithm 4, which proceeds as follows.

Algorithm 4: Policy π^a.

```
$Q_0 \leftarrow \emptyset$; $E_0 \leftarrow \emptyset$; $S_1^{\pi^a} \leftarrow \emptyset$; $H_1 \leftarrow \emptyset$;
for $t = 1, ...$ do
    $N_t :=$ set of new request arrivals;
    $Q_t \leftarrow (Q_{t-1} \setminus E_{t-1}) \bigcup N_t$;
    $x^{\pi^a}_{i,t'} \leftarrow 0, \forall t' \in [t, e_i], \forall i \in N_t$;
    $T^{\pi^a}_{i,t} \leftarrow \emptyset, \forall i \in N_t$;
    $\hat{x}^{\pi^a}_{i,t,t'} \leftarrow 0, \forall t' \in [t, e_i], \forall i \in Q_t$;
    $\hat{T}^{\pi^a}_{i,t,t} \leftarrow \emptyset, \forall i \in Q_t$;
    $\tau \leftarrow \max_{i \in Q_t} e_i$;
    while $\tau \geq t$ do
        $A_\tau \leftarrow \{i : i \in Q_t, \tau \in [a_i, e_i]\}$;
        Go to Tentative schedule ▷ Get $i^*$;
        $\hat{x}^{\pi^a}_{i^*,t,\tau} \leftarrow 1$; $\hat{T}^{\pi^a}_{i^*,t,\tau} \leftarrow \hat{T}^{\pi^a}_{i^*,t,\tau+1} \bigcup \{\tau\}$;
        if $\tau = t$ then
            $x^{\pi^a}_{i^*,t} \leftarrow 1$; $T^{\pi^a}_{i^*,t} \leftarrow T^{\pi^a}_{i^*,t-1} \bigcup \{t\}$;
        $\tau \leftarrow \tau - 1$;
    $E_t := \{i : i \in Q_t, e_i = t\}$ ;
    $D^{\pi^a}_t := \{i : i \in E_t, T^{\pi^a}_{i,t} \neq \emptyset\}$ ;
    $S^{\pi^a}_{t+1} \leftarrow S^{\pi^a}_t \bigcup D^{\pi^a}_t$;
    $H_{t+1} \leftarrow H_t \bigcup E_t$ ;
```

For every slot t, π^a operates in a backwards manner starting from the last slot in which it can tentatively schedule an update, $\max_{i \in Q_t} e_i$, all the way back

Algorithm 5: Tentative schedule.

```
1 Input: $(\hat{x}^{\pi^a}_{i,t,t'}, \forall t' \in [\tau+1, e_i], \forall i \in Q_t)$, $(\hat{T}^{\pi^a}_{i,t,\tau+1}, \forall i \in Q_t)$,
    $(x^{\pi^a}_{i,t'}, \forall t' \in [a_i, t-1], \forall i \in Q_t)$.
2 Output: $i^*$.
3 for $i \in A_\tau$ do
4     for $j \in A_\tau \setminus \{i\}$ do
5         $\tilde{T}^{\pi^a}_{i,t,\tau} \leftarrow \hat{T}^{\pi^a}_{i,t,\tau+1} \bigcup \{\tau\}$;
6         $\tilde{x}^{\pi^a}_{i,t,t'} \leftarrow \hat{x}^{\pi^a}_{i,t,t'}, \forall t' \in [\tau+1, e_i]$; $\tilde{x}^{\pi^a}_{i,t,\tau} \leftarrow 1$;
7         $m^{\pi^a}_{t,\tau}(i,j) = \frac{f(e_i - \max_{t' \in [\tau, e_i]} \tilde{x}^{\pi^a}_{i,t,t'} t') + f(e_j - \max_{t' \in [a_j, t-1]} x^{\pi^a}_{j,t'} t')}{f(e_j - \tau) + \hat{r}^{\pi^a}_{i,t,\tau+1}}$;
8 $i^* \in \arg\max_{i \in A_\tau} \left[ \min_{j \in A_\tau \setminus \{i\}} m^{\pi^a}_{t,\tau}(i,j) \right]$; break ties by selecting $i^*$ with smallest $e_{i^*}$;
    break ties arbitrarily;
```

to t. For every slot τ between t and $\max_{i \in Q_t} e_i$, π^a first determines A_τ, the subset of active requests in Q_t that can be tentatively scheduled in slot τ. Let $i \in A_\tau$ and $j \in A_\tau \setminus \{i\}$ be two requests that will be active in slot τ. In Algorithm 5, for every τ, π^a evaluates $m^{\pi^a}_{t,\tau}(i,j)$ (Line 7). The reasoning behind $m^{\pi^a}_{t,\tau}(i,j)$ is intuitive and described as follows. π^a evaluates through $m^{\pi^a}_{t,\tau}(i,j)$ if it is more advantageous in terms of aggregate tentative reward to tentatively schedule request i in slot τ instead of request j. Slots between $\tau + 1$ and e_j in which updates to request j are tentatively scheduled under π^a are ignored when evaluating $m^{\pi^a}_{t,\tau}(i,j)$, in an attempt to characterize the importance of tentatively scheduling an update to request j solely in slot τ. A detailed derivation of $m^{\pi^a}_{t,\tau}(i,j)$ is provided as follows. The numerator of $m^{\pi^a}_{t,\tau}(i,j)$ corresponds to the sum of the following two components:

- Tentative reward of request i if i is tentatively scheduled in any slots in $\hat{T}^{\pi^a}_{i,t,\tau+1} \bigcup \{\tau\}$.

- Tentative reward of request j if j was only scheduled prior to slot t.

The denominator of $m^{\pi^a}_{t,\tau}(i,j)$ corresponds to the sum of the following two components:

- Tentative reward of request j if j is tentatively scheduled in slot τ.
- Tentative reward of request i if i is tentatively scheduled in any slots in $\hat{T}^{\pi^a}_{i,t,\tau+1}$ and/or scheduled in any slots in $T^{\pi^a}_{i,t-1}$.

Therefore, $m^{\pi^a}_{t,\tau}(i,j)$ evaluates the ratio between the aggregate tentative rewards resulting from (1) tentatively scheduling request i at least once after and including slot τ while only considering slots in which request j was actually scheduled prior to t and (2) tentatively scheduling request j only in slot τ while considering all slots strictly greater than τ in which i was tentatively scheduled to receive updates and/or all slots prior to t in which updates to request i were scheduled.

π^a then evaluates $m^{\pi^a}_{t,\tau}(i,j)$ for every request $i \in A_\tau$, and for all $j \in A_\tau \setminus \{i\}$ (Lines 3-7 of Algorithm 5), and determines the request $j \in A_\tau \setminus \{i\}$ for which the ratio $m^{\pi^a}_{t,\tau}(i,j)$ is minimized, i.e., π^a's potential decision to tentatively schedule request i in slot τ instead of request j deemed the least advantageous.

π^a finally tentatively schedules in slot τ the request i^* such that $\min_{q \in A_\tau \setminus \{i^*\}} m^{\pi^a}_{t,\tau}(i^*,q) \geq \max_{j \in A_\tau \setminus \{i^*\}} \min_{q \in A_\tau \setminus \{j\}} m^{\pi^a}_{t,\tau}(j,q)$ (Line 8 of Algorithm 5).

Figure 4.1 provides an example of requests scheduled and tentatively scheduled to receive updates under policy π^a. Requests i_3 and i_4 became active at the beginning of slot $t = 3$. We observe in the left subfigure of Figure 4.1 that requests

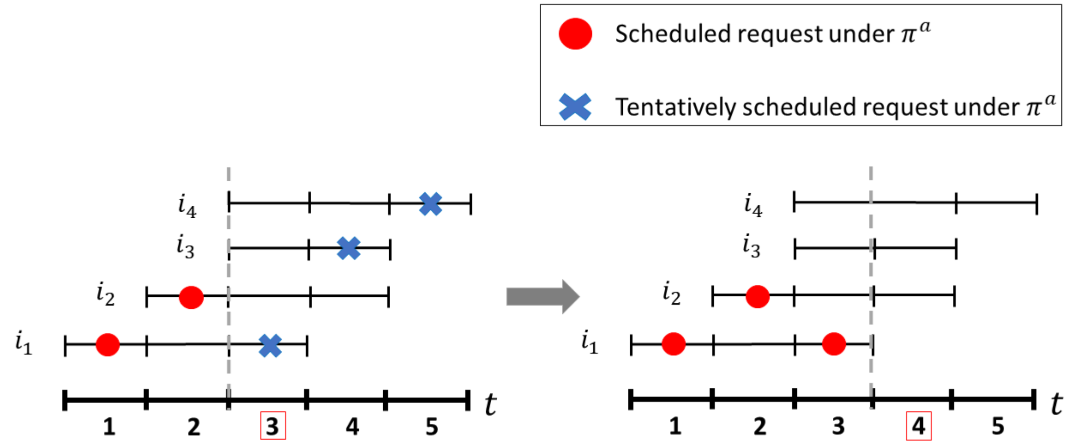

● Scheduled request under π^a
✖ Tentatively scheduled request under π^a
i_4
i_3
i_2
i_1
1
2
3
4
5
t
i_4
i_3
i_2
i_1
1
2
3
4
5
t

Figure 4.1: Scheduling and tentatively scheduling under policy π^a of requests with maximal update window of $w_{\max} = 3$ and receiving a linear reward as defined in Definition 4.3.3, with $\alpha = 1, \beta = 0.1, c = 0$.

i_1 and i_2 were scheduled to receive updates under π^a in slots $t = 1$ and $t = 2$ respectively. Additionally at $t = 3$, requests i_4, i_3 and i_1 are tentatively scheduled to receive updates under π^a in slots $t = 5$, $t = 4$ and $t = 3$ respectively. Once all slots in the interval $[3, 5]$ have been reserved to tentatively schedule updates for active requests, π^a then schedules an update to request i_1 in slot $t = 3$ (right subfigure of Figure 4.1). The same scheduling process is repeated as long as there are active requests.

4.5.4 Discussion of π^g and π^a

We provide insights on both proposed causal policies, π^g and π^a. We first define $m^{\pi^g}_{t,t}(i,j)$ for two active requests i and j in Q_t as follows,

$$m^{\pi^g}_{t,t}(i,j) = \frac{f(e_i - t) + f(e_j - \max\limits_{t' \in [a_j, t-1]} x^{\pi^g}_{j,t'} t')}{f(e_j - t) + f(e_i - \max\limits_{t' \in [a_i, t-1]} x^{\pi^g}_{i,t'} t')}. \tag{4.9}$$

If π^g schedules request $i \in Q_t$ in slot t then the following corollary applies.

Corollary 4.5.1 *Policy π^g schedules an update transmission in slot t to active request $i \in Q_t$ if and only if $\min_{j \in Q_t \setminus \{i\}} m^{\pi^g}_{t,t}(i,j) \geq 1$.*

Proof 4.5.4 *Since π^g schedules an update transmission in slot t to active request $i \in Q_t$, it follows that $\forall j \in Q_t \setminus \{i\}$, $f(e_i - t) - f(e_i - \max_{t' \in [a_i, t-1]} x^{\pi^g}_{i,t'} t') \geq f(e_j - t) - f(e_j - \max_{t' \in [a_j, t-1]} x^{\pi^g}_{j,t'} t')$, which implies that,*

$$\forall j \in Q_t \setminus \{i\}, \quad \frac{f(e_i - t) + f(e_j - \max\limits_{t' \in [a_j, t-1]} x^{\pi^g}_{j,t'} t')}{f(e_j - t) + f(e_i - \max\limits_{t' \in [a_i, t-1]} x^{\pi^g}_{i,t'} t')} \geq 1.$$

This is equivalent to saying that,

$$\min_{j \in Q_t \setminus \{i\}} \frac{f(e_i - t) + f(e_j - \max\limits_{t' \in [a_j, t-1]} x^{\pi^g}_{j,t'} t')}{f(e_j - t) + f(e_i - \max\limits_{t' \in [a_i, t-1]} x^{\pi^g}_{i,t'} t')} \geq 1.$$

We can similarly prove the other direction of the condition, which concludes the proof.

It follows from Corollary 4.5.1 that π^g schedules in every slot the request $i^* \in \arg\max\limits_{i \in Q_t} \left[\min\limits_{j \in Q_t \setminus \{i\}} m^{\pi^g}_{t,t}(i,j) \right]$. In comparison with π^a, if we set $\forall t, \forall i \in Q_t, \hat{T}^{\pi^a}_{i,t,t+1} = \emptyset$, then $\forall i, j \in Q_t$, $m^{\pi^a}_{t,t}(i,j)$ is equal to $m^{\pi^g}_{t,t}(i,j)$ and π^a becomes equivalent to π^g. In other words, if π^a does not tentatively schedule requests, then it is equivalent to the greedy request scheduling policy π^g.

According to the above derivation, π^g does not tentatively schedule active requests at time t and takes a restrictive approach by prioritizing the scheduling of any active requests that have not been scheduled prior to t. On the other hand, π^a allows for more flexibility in rescheduling active requests, where an active request that has already been scheduled prior to some slot t, can be rescheduled in t, even if this might be at the expense of not scheduling other active requests that have not been scheduled yet prior to t. It follows that π^a does not make any assumptions about the future load of requests' arrivals and its implications on the tentative schedules. We therefore expect π^a to achieve a higher reward-rate than π^g in systems with small loads and requests with large active window's lengths. On the other hand, we expect that both π^a and π^g would achieve a similar reward-rate rate in systems with high-loads and with requests that have small active windows, balanced on the one hand by π^g's urgency to schedule new requests as soon as they become active, and on the other hand by π^a attempting to schedule/reschedule requests as close as possible to the end of their active windows.

4.6 Numerical evaluations

We conducted numerical evaluations to explore the performance of our proposed policies π^a and π^g in maximizing the reward-rate introduced in Problem 4.3.1 and evaluate their results with respect to other baseline policies which we introduce below.

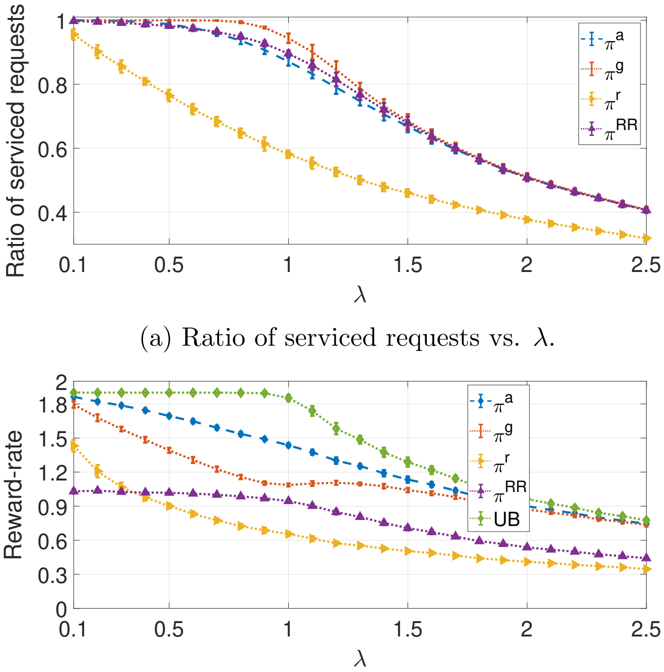
4.6.1 Model

We shall present results for a convex reward function that satisfies the conditions in Assumption 4.3.1. We let $h(x) = e^x$ in Definition 4.3.4 and set $\alpha = 1$, $\beta = 2$ and $c = -\frac{1}{2}\ln(0.9)$. It follows that the convex exponential reward obtained after servicing request i under policy π is $r_i^\pi = 1 + 0.9e^{-2(e_i - \max_{t\in[s_i,e_i]} x_{i,t}^\pi t)}$. The maximal achievable reward in this setting is equal to 1.9 whereas the smallest reward obtained after servicing a request is $1 + 0.9e^{-2w_{\max}}$. We consider the setting where the number of new request arrivals at the beginning of every slot is drawn from a Poisson distribution with intensity λ, whereas a request's active window length is generated from a discrete uniform distribution $\sim U[1, w_{\max}]$. We run simulations over a finite-time of length $T = 1000$, where T is the last slot after which there are no longer any active requests in the system. Our simulation results represent averages over randomly generated requests' arrivals as well as requests' active windows lengths. We ran 100 Monte-Carlo (MC) simulations and plotted both the mean ratio of serviced requests for any policy π, i.e., $\frac{|S^{\pi}_{T+1}|}{|H_{T+1}|}$, and the reward-rate as well as the confidence intervals corresponding to the standard deviation of the estimator resulting from the MC simulations.

4.6.2 Scheduling policies

In addition to presenting the results for all of π^a and π^g, we consider two baseline policies. π^r is a causal policy that randomly schedules an active request in a slot. π^{RR} schedules active requests in a round-robin-like fashion which we describe as follows. At the beginning of every slot, the set of new requests is considered for scheduling right after the set of requests that arrived prior to this slot and have not been scheduled yet.

Due to the brute-force nature of the optimal offline policy π^* we provide instead an upper bound (UB) on the maximal achievable reward-rate, which is equal to the product between the maximal ratio of serviced requests (achieved by π^g) and the maximal achievable reward which is equivalent to 1.9 in this setting,



(a) Ratio of serviced requests vs. λ.

(b) Reward-rate vs. λ.

Figure 4.2: Ratio of serviced requests and reward-rate when the reward is convex, $w_{\max} = 30$ and λ is increasing.

normalized by the total number of requests H_T.

4.6.3 On the impact of the load of requests with fixed maximal window length

We fix $w_{\max} = 30$ and increase λ from 0.1 to 2.5. A first interesting observation is that all of π^a, π^g and π^{RR} maximize the ratio of serviced requests in both regimes where $\lambda \leq 0.4$ and $\lambda \geq 1.6$. Whereas for $0.4 \leq \lambda \leq 1.6$, π^g maximizes the ratio of serviced requests. Another interesting observation is that for $\lambda \leq 0.7$, the ratio of serviced requests achieved by π^g is constant and equal to 1, whereas the reward-rate achieved by π^g is significantly decreasing in that range. The following behavior is due to the fact that π^g maximizes the number of serviced requests at the expense of scheduling requests closer to the end of their active windows. Therefore, this justifies that π^g is not suitable for the reward-rate maximization in low-load systems.

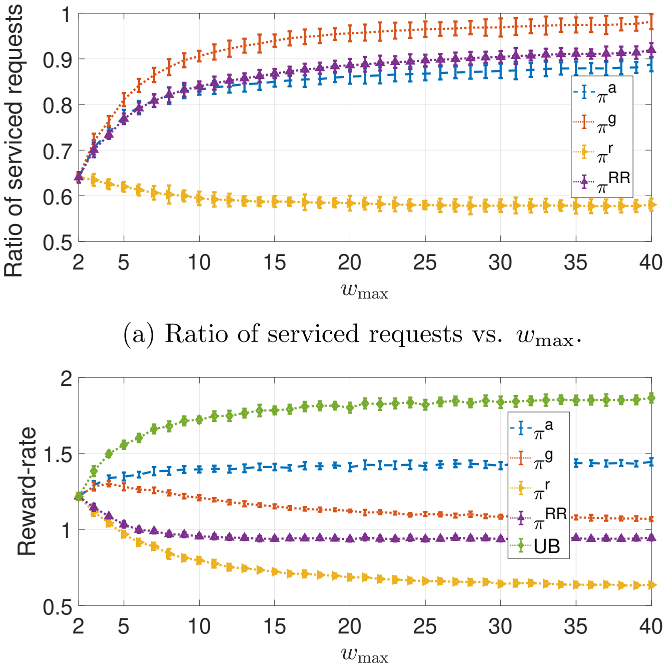
On the other hand, π^a achieves the largest reward-rate among all proposed causal policies for $\lambda \leq 2.1$ even if it is linearly decreasing as λ increases. The linear decrease in the reward-rate is justified since π^a may reschedule active requests in slots closer to their end times in an attempt to maximize the cumulative reward, which may come at the expense of servicing requests that have not been scheduled yet.

Finally, for $\lambda \geq 2.1$, both π^a and π^g achieve the largest reward-rate, close to the maximal achievable reward-rate, which is aligned with our previous analysis suggesting that while π^a is superior in systems with low-loads, both π^a and π^g achieve similar performance in systems with high requests' arrival rate.

4.6.4 On the impact of increased update window flexibility

We fix $\lambda = 1$ and increase $w_{\max}$ from 2 to 40. The results are shown in Figure 4.3. A first observation is that as $w_{\max}$ increases, the ratio of serviced requests under π^a, π^g and π^{RR} increases because of the additional flexibility in scheduling requests in more slots. For $w_{\max} \leq 3$, we observe that both π^a and π^g achieve the same reward-rate. For $w_{\max} \geq 4$, π^a achieves in a higher reward-rate than any of the other proposed causal policies. An interesting observation is that the reward-rate achieved by π^g is clearly decreasing as $w_{\max}$ increases. There are two factors that concurrently lead to the following phenomenon, the first one being that for $w_{\max} \geq 4$, the ratio of serviced requests under π^g slowly increases as $w_{\max}$ increases, and the second one being that the minimal reward obtained from servicing a request is decreasing as $w_{\max}$ increases.

That said, and as aligned with our previous discussions, π^a is superior to π^g in such settings with fixed arrival rate but requests with large active windows, since it allows for more flexibility in scheduling updates as close as possible to the requests' end times, whereas π^g is driven towards maximizing the ratio of scheduled requests with less priority to scheduling those requests closer to their end times.



(a) Ratio of serviced requests vs. $w_{\max}$.

(b) Reward-rate vs. $w_{\max}$.

Figure 4.3: Ratio of serviced requests and reward-rate when the reward is convex, $\lambda = 1$ and $w_{\max}$ is increasing.

4.7 Chapter Conclusion and Future Work

In this chapter we have developed a model where updates are required by requests prior to making timely decisions regarding time-varying processes they're interested in. Requests can receive updates only within time windows of finite length. A key aspect is the design of a reward function that captures the importance of scheduling the freshest update transmission to a request as close as possible to the decision time as well as scheduling policies that achieve high rewards in adversarial settings. A key part of our future work is to extend our model to a real-time information market that includes multiple servers and allows for multiple updates' transmissions per slot by matching requests to servers through efficient algorithms.

Part II

Timely Information Sharing in Multi-User 360° Virtual Reality Communication Constrained Settings

Chapter 5

Robust Multi-User 360° Virtual Reality Video Delivery

The ability that Virtual Reality (VR) systems have to deliver immersive experiences to customers is increasingly becoming attractive and has shifted high-end VR manufacturers from designing wired head-mounted devices (HMDs) to enabling fully wireless solutions, which have leveraged the advantages of edge servers to meeting VR systems' demands. Such approaches have unlocked the advantages of wireless technologies with high-throughput in providing timely delivery of massive amounts of data to users' VR devices and have paved the way towards more interactive VR experiences among multiple users. But with such advancements come major challenges. Providing customers with such VR experiences is burdensome from multiple perspectives. For instance, the variability in users' wireless channels may lead to disappointment in the overall users' experience by limiting the transfer of data streams, e.g., users traveling in a car/train watching VR videos on their headsets may be more prone to Quality-of-Experience (QoE) disruptions. Further, high-quality 360° VR videos require massive amounts of data to be constantly streamed, which may overwhelm shared communication resources and result in poor user experience. Motivated by the fact that users only view a portion of the 360° video content, viewport-adaptive streaming solutions that only stream in high quality what the users are viewing or likely to view have emerged as a key technique to efficiently serve users and save on communication resources.

With that in mind, we propose in this chapter [1] a new approach to users' viewing predictions, which we refer to as a statistical model for the users' viewing processes in 360° VR video applications that not only predicts the portion of the video content that's most likely to be viewed in the future, but predicts as well a set of possible viewports over a future time-window along with their probabilities of being viewed by the users. The advantage of such a model is that it captures in a more detailed manner the level of certainty regarding the users' viewport predictions which may support more robust decision-making in deciding how to serve users. General settings may include users with high and/or low uncertainty regarding their predicted viewing patterns. Users with high prediction uncertainty may burden the communication medium due to the huge amount of data sent which may still result in poor users' QoE due to a small hit rate of their viewports. On the other hand, users with low uncertainty may have overall better VR experiences due to the small amount of data required to achieve such good VR performance. The question of VR QoE fairness arises in such settings where users with different levels of uncertainty regarding their predictions and experiencing potentially different network conditions share the same resources. Such challenges encountered in fully immersive VR systems need to be tackled through careful design of metrics and policies that account for the variability in the network channel and prioritizes QoE fairness while servicing multiple users sharing the same resources.

[1] Part of the results in this chapter are in preparation for future submission.

5.1 Related work

Recent VR systems have shifted from tethered connections between the edge server and the users' VR devices to untethered designs. Such designs require high data rate transmissions over congested wireless resources which induces new challenges that should be taken into consideration such as users' mobility, wireless blockages, etc., and therefore require careful design of the wireless system model [50–53].

360° VR videos have emerged as a popular use case of VR applications. Viewport prediction [54–57] has become a core research topic for such VR applications as a means to relieve the load on the communication medium by only transmitting portions of the video that are most likely to be viewed by the users. Most of the literature considers predicting the viewport that is most likely to be viewed in the next video frame, or the sequence of viewports most likely to be viewed in future video frames. By contrast, we propose predicting statistical models for users' viewing processes which capture the different possible future viewing paths that a user might take, with their associated probabilities. Further, the emerging role of the edge for wireless VR in untethered VR settings has been extensively studied in the literature. For instance, prediction of users' future viewports can be offloaded to the edge server. In fact, [58–63] explore fundamental trade-offs between caching at the edge, computing and communication in wireless VR applications.

Tile-based video streaming [64–66] has become popular since only tiles that fall in a user's viewport are streamed to the user's VR device. The end-to-end delay in real-time streaming of tile-based viewports is significant due to the fact that the edge server needs to predict the users' future viewports based on their past 3DoF poses and transmit the predicted tiles in a timely manner to the VR headsets, where stitching of the tiles occur before being displayed. A more recent work, [67], proposes an alternative streaming approach, CoRE, that samples video frames non-linearly in both the spatial and temporal domains. A CoRE frame samples a user's predicted field-of-view with full resolution and gradually decreasing resolution at the periphery. A CoRE chunk additionally consists of a main part at full frame rate and an extension part at a gradually decreasing frame rate. Our proposed approach to predicting users' statistical models for their viewing processes can be incorporated in such VR video streaming approaches to further enhance their robustness to both view prediction errors and to network fluctuations.

Multi-user VR settings have also recently attracted a lot of attention. For instance, [57, 65, 68–70] consider multi-user VR settings where multiple users share the same Access Point (AP). Different users' Quality-of-Experience (QoE) metrics have been proposed, to either capture the ratio of the correctly predicted and displayed tiles [68, 69], or the quality of the displayed tiles to the users [57], or a linear combination of the quality of the perceived content, the average VR content delivery delay and the variance of the perceptual quality [70]. The above mentioned work aim to maximize the users' QoE subject to communication and computation constraints. We propose in our work a QoE metric that accounts for the proportion of high quality tiles viewed by the users as well for the QoE fairness among all users sharing the same edge.

5.2 Chapter Contributions.

We consider a multi-user VR setting where users can initiate service at any time and select a 360° VR video to watch and are serviced by a nearby edge/BS. We propose a viewport-adaptive tile-based video streaming approach compatible with the Dynamic Adaptive Streaming over HTTP (DASH) protocol, that schedules tiles for transmission based on the directions the users are predicted to look at. We introduce a novel viewport predictor that estimates the statistical model for users' viewing paths represented as trees which consist of different sequences of viewports that a user may view in the future, along with the viewports' predicted probabilities of being viewed. We propose a video-streaming framework that prioritizes the delivery of the low-definition content of the videos to guarantee that all users' possible viewports are covered, and then aims at enhancing the users' VR experience through the delivery of the high-definition predicted tiles. A user's quality of experience (QoE) is hence tied to the availability in its VR device of the fraction of HD tiles that fall in its current viewport. We further propose a multi-user system utility function that accounts for QoE fairness among users sharing the same resources and devise an HD tile scheduling policy that proactively schedules the transmission of users' predicted future views which achieves a degree of robustness to network capacity fluctuations. The scheduling policy further leverages multicast services provided by the BS to transmit users' predicted overlapping views. Simulations based evaluations showcase the robustness of our proposed viewport prediction model and scheduling policy against heterogeneity in users' predicted viewings and network fluctuating capacities. Finally we explore through preliminary theoretical

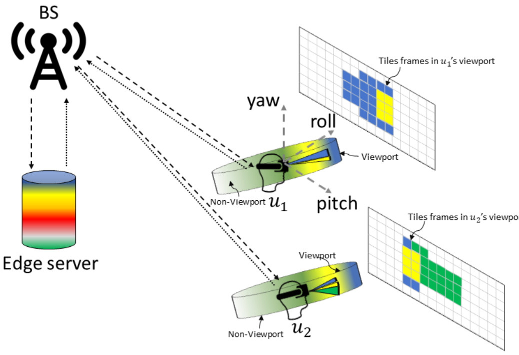


Figure 5.1: BS, edge and two users equipped with VR mobile devices viewing the same video frame and having overlapping viewports.

analysis in idealized settings, the impact of both the users' predictability in their viewings and their network capacities on their mean QoE.

We consider a setting where a catalog of 360° VR videos are stored at a edge node co-located with a wireless Base Station (BS). Users equipped with mobile VR headsets initiate viewing sessions at random times by selecting the 360° VR video they wish to watch. For simplicity we assume that users are served by the same fixed edge/BS but more generally mobile users, e.g., within a car/bus, would be handed off to other edge/BS resources when appropriate. Figure 5.1 depicts an example of edge, BS and associated users equipped with VR headsets.

All 360° VR videos at the edge have a frame rate of $\frac{1}{\beta}$ frames per second, i.e., the time between two successive video frames is β seconds. The user's mobile VR device tracks its 3DoF pose and reports it to the edge every β seconds as well. We shall consider a time-slotted system where each slot corresponds to a video frame which is indexed by τ and is of length β seconds. Users initiate 360° VR video viewing sessions at possibly different times and leave the network when the VR video they are watching ends. We let a^u be the time at which user u starts watching its selected video and we let w^u be the length of user u's viewing session in terms of the number of slots. We refer to users watching a 360° VR video on their VR headsets as active users. We let U_τ denote the set of active users at the start of slot τ.

Remark 5.2.1 *For simplicity, we assume that new users can become active only at the beginning of a slot τ.*

For simplicity, we do not consider the pausing/rewinding of the 360° VR videos, but they are natural extensions of our model.

5.2.1 User's viewport and video compression model

A user's viewport $\boldsymbol{v}$ in a 360° VR video frame is a 3-D vector with its components being the pitch, roll and yaw angles respectively, which corresponds to the direction in which the user is looking at. We let $\mathcal{V}$ denote the set of all viewports/viewing directions that any user can look at. The viewport viewed by an active user u in slot τ is denoted by $\boldsymbol{v}_\tau^{u,*}$.

Remark 5.2.2 *In the sequel we refer to a user's viewport as the direction in which the user is, or may be, looking at.*

Each 360° VR video is temporally segmented into video chunks in a manner which is compliant with the DASH paradigm [71], and each video chunk is spatially divided into tiles. Temporal video segmentation facilitates temporal compression by

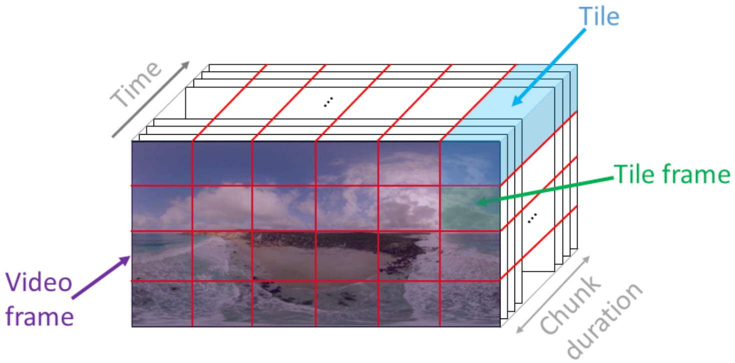


Figure 5.2: Video chunk, tiles and tile frames.

exploiting redundancy and allows for the efficient delivery of relevant content [72]. Each tile has the same duration and same number of frames as a video chunk. An example of a video frame, video chunk, its constituent tiles and tile frames is illustrated in Figure 5.2.

VR video encoding is performed using a layered approach [73,74], where the availability of the lower layer tile encoding results in Low-Definition (LD) content and the *additional* availability of the higher layer tile encoding results in High-Definition (HD) content. Both low/high encodings of all tiles are available at the edge.

We let $\mathcal{F}_\tau^u$ denote the set of tile frames in the video frame to be displayed to user u at time τ. We let $h^u(\boldsymbol{v},\tau)$ be the function that maps viewport $\boldsymbol{v} \in \mathcal{V}$ that is viewed, or may be viewed by user u in slot τ to a subset of tile frames in $\mathcal{F}_\tau^u$. We note that a user's viewport maps to different sets of tile frames in different video frames of the same video. We let d_f^u for $f \in \mathcal{F}_\tau^u$, be the time at which tile frame f may be viewed by user u and we refer to this as the *viewing deadline* of f.

Hence, for $f \in \mathcal{F}_\tau^u$, $d_f^u = \tau$. We further let $\mathcal{T}_\tau^u$ denote the set of tiles that belong to the video chunk that u is watching at time τ. We let $g(\cdot)$ be a one-to-one mapping function that maps a tile frame to the tile it belongs to. We finally let b_t denote the size in bits of HD tile $t \in \mathcal{T}_\tau^u$.

5.2.2 VR video delivery and user's Quality of Experience model

The edge/BS coordinates the delivery of VR video tiles in each slot τ, i.e., in between the time to display two successive video frames to active users U_τ. The edge/BS schedules the transmissions at the granularity of LD and HD tiles rather than scheduling LD and HD tile frames or video chunks. Every slot τ the edge server prioritizes the delivery of the LD tiles which are sent sequentially using a traditional video streaming protocol, e.g., DASH [71]. The edge guarantees the transmission of all LD tiles that belong to the next video chunk of each active user, motivated by the small size of the LD tiles and to provide robustness to fast changes in users' viewports. The edge/BS then predicts the set of viewports that users may view in future slots, maps them to the corresponding set of tiles and finally opportunistically schedules the transmission of a subset of the HD tiles to the users depending on the available capacity at the BS. HD tiles are sent to users to further enhance their VR experience. The delivery of HD tiles is supported by low-latency transport protocol such as QUIC [75], which can be utilized for streaming video content. We assume that users' headset devices have two buffers, one that caches LD tiles and another that caches HD tiles. For the remainder of this work, we will solely focus on the scheduling of HD tiles' transmissions from the edge/BS to the users, as only HD tiles cached in a user's device's buffer will account for the user's QoE. black

We let $Q_\tau^{u,\pi}$ denote the set of HD tiles cached in user u's device buffer at time τ under HD tile scheduling policy π, and assume that u's buffer can cache in slot τ at most l_τ^u bits tied only to HD tiles. We hence propose to model a user's QoE in a VR session as a function of the set of HD tiles cached in its device's buffer and available prior to their viewing deadlines.

Definition 5.2.1 (User's QoE) *A user's QoE is defined as a function of the fraction of HD tile frames whose tiles are cached in the user's device's buffer at their deadline and that fall in the viewport viewed by the user. User u's QoE under an HD tile scheduling policy π is defined as,*

$$\mathrm{QoE}^{u,\pi} = \frac{1}{w^u} \sum_{\tau=a^u}^{a^u+w^u} q\left(\frac{1}{|h^u(\boldsymbol{v}_\tau^{u,*},\tau)|} \sum_{f \in h^u(\boldsymbol{v}_\tau^{u,*},\tau)} \mathbb{1}_{\{g(f)\in Q_\tau^{u,\pi}\}} \right),$$

where $\mathbb{1}_{\{g(f)\in Q_\tau^{u,\pi}\}}$ takes value 1 if HD tile $g(f)$ is in $Q_\tau^{u,\pi}$ and 0 otherwise, $|h^u(\boldsymbol{v}_\tau^{u,*},\tau)|$ is the number of tile frames in $\boldsymbol{v}_\tau^{u,*}$ and where $q(\cdot)$ is a strictly increasing concave function which provides some flexibility in modeling a user's QoE in 360° VR applications.

So, a user's QoE marginal gain for a time slot may decrease as the instantaneous hit rate of tiles cached in the user's buffer increases. This is captured through the $q(\cdot)$ function.

We now propose a multi-user system utility function that realizes QoE tradeoffs across dynamic users and defined as follows.

Definition 5.2.2 (Multi-user system utility) *The multi-user system utility is defined as the sum of the natural logarithmic function of the users' QoEs. The utility function under an HD tile scheduling policy π is defined as*

$$v(\mathbf{QoE}^{\pi}) = \sum_{u \in U} \log\left(\mathrm{QoE}^{u,\pi}\right),$$

where $\mathbf{QoE}^{\pi} := (\mathrm{QoE}^{u,\pi} : u \in U)$ and $U = \bigcup_{\tau} U_{\tau}$.

The natural logarithmic function encourages a degree of QoE fairness across users.

5.2.3 Edge/BS resources

The BS provides both unicast and multicast transmission services. Its resources may be shared with other high priority traffic as well and thus possibly only limited congested wireless resources are available to transmit HD tiles. We assume that the edge has a rough estimate of the available wireless resources to transmit HD tiles in slot τ once LD tiles and other high priority traffic are delivered. We let θ_{τ}^{u} denote the estimate of the maximal number of bits that the edge can transmit to user u throughout slot τ if all available network resources are allocated to u. We further let $\rho_{\tau} \leq 1$ be the fraction[1] of the available resources that can be used for multicast transmissions. The edge/BS finally proactively schedules the transmission of a subset of HD tiles to their corresponding users based on its estimate of the available resources, and receives feedback regarding their successful delivery. The edge guarantees that no tile is delivered more than once to the same user. If an HD tile t is scheduled in slot τ for multicast transmission to a subset of active users $A_{\tau} \subseteq U_{\tau}$, the edge/BS estimates the maximal number of bits that it can transmit to all users

[1] For example, up to 60 percent of a Long-Term Evolution (LTE) frequency carrier may be dedicated to Multimedia Broadcast/Multicast Service (MBMS) [76, 77]

in A_τ. For simplicity, we shall assume that the edge/BS adapts the transmission rate to match that of the user with the worst transmission rate in A_τ. Hence, the proportion of resources reserved for the multicast transmission of t to users in A_τ in slot τ is $\max_{u \in A_\tau} \frac{b_t}{\theta_\tau^u}$.

Finally at the beginning of slot $\tau + 1$, the edge discards any HD tiles that were not successfully transmitted in slot τ to their corresponding users. Note that tiles cached in the user's buffer and that belong to a specific video chunk are deleted once frames therein are done being viewed by the user.

5.3 Multi-user system utility maximization

We define the following notation to properly account for unicast and multicast transmissions at the BS.

- $\mu_\tau^{\pi,u} \leq \theta_\tau^u$, is the unicast transmission rate assigned to user $u \in U_\tau$ under policy π in slot τ.
- B_τ^π is the set of HD tiles that are multicast to subsets of users in U_τ in slot τ under π.
- $M_{\tau,t}^\pi \subseteq U_\tau$ is the subset of users to which HD tile $t \in B_\tau^\pi$ is multicast to in slot τ under π.
- $R_{\tau,t}^\pi \subseteq U_\tau$ is the subset of users to which HD tile t is unicast to in slot τ under π.

We now formulate the multi-user system utility maximization problem.

Problem 5.3.1 (Multi-user system utility maximization).

$$
\begin{aligned}
&\max_{\pi \in \Pi} v(\mathbf{QoE}^{\pi}) \\
\text{s.t. } &\sum_{t \in B_{\tau}^{\pi}} \max_{u \in M_{\tau,t}^{\pi}} \frac{b_t}{\theta_{\tau}^{u}} \leq \rho_{\tau}, \ \forall \tau, && (5.1) \\
&\sum_{u \in U_{\tau}} \frac{\mu_{\tau}^{u,\pi}}{\theta_{\tau}^{u}} \leq 1 - \sum_{t \in B_{\tau}^{\pi}} \max_{u \in M_{\tau,t}^{\pi}} \frac{b_t}{\theta_{\tau}^{u}}, \ \forall \tau, && (5.2) \\
&\sum_{t \in Q_{\tau}^{u,\pi}} b_t \leq l_{\tau}^{u}, \forall u \in U_{\tau}, \forall \tau, && (5.3)
\end{aligned}
$$

where Π *denotes the set of all policies that schedule the transmission of HD tiles to users.*

Equation (5.1) enforces a constraint on the proportion of available resources that can be used for multicasting, while Equation (5.2) is a constraint on the fraction of time that a user's channel can be reserved for unicasting. Equation (5.3) enforces a constraint on the number of HD tiles that can be cached in a user's buffer.

5.4 Proposed algorithm for Problem 5.3.1

We now propose an algorithm geared at maximizing the multi-user system utility in Problem 5.3.1. It comprises of three steps repeated every slot τ as follows. First, the edge predicts the set of viewports that the user may view in future slots through a predictor which we refer to as a Statistical Model Predictor (*SMP*) formally described in Subsection 5.4.1, then maps the tile frames in future predicted viewports to their corresponding sets of HD tiles (Subsection 5.4.2). Second, the edge/BS predicts the communication resources available in slot τ for the transmission of predicted HD tiles. For simplicity, we will assume that a crude estimate of the available resources is available at the edge/BS every slot τ. Third, the edge/BS leverages its viewports and capacity predictions to schedule the HD tiles to be either unicast or multicast. We refer to our policy as π^F as it is geared at achieving a fair allocation of QoE across users. π^F is described in detail in Subsection 5.4.3. We refer to our proposed algorithm as *SMP*+π^F.

5.4.1 Statistical model for a user's viewing process

We propose to predict a statistical model for a user's future viewing process given the past viewing history. The model corresponds to a tree of depth ω and branching factor of at most α, where each tree node corresponds to a possible viewport seen at a future time slot– to reduce the model's complexity and recognizing that viewports may shift relatively slowly as compared to a frame duration, the tree's nodes at different depths correspond to viewports sampled every γ slots.

Figure 5.3 exhibits an estimated statistical model for user u's viewing process with $\alpha = 2, \omega = 2$ and $\gamma = 5$ given the viewing history up to and including slot τ.

We shall let $\hat{\mathcal{V}}_{\tau}^{u} := \{n_{\tau,i}^{u} : i \in \{1, \ldots, m_{\tau}^{u}\}\}$ denote the set of nodes in the tree for user u's statistical model as estimated at time τ whence for future slots $\tau + \gamma, \ldots, \tau + \omega\gamma$, which comprises of m_{τ}^{u} nodes, and where $n_{\tau,i}^{u} = (\hat{\boldsymbol{v}}_{\tau,i}^{u}, k_{\tau,i}^{u}, \hat{p}_{\tau,i}^{u})$ is a node indexed by i and characterized by (1) a predicted viewport $\hat{\boldsymbol{v}}_{\tau,i}^{u} \in \mathcal{V}$; (2) a slot $k_{\tau,i}^{u} \in \{\tau + \gamma, \ldots, \tau + \omega\gamma\}$; and (3) an estimated probability $\hat{p}_{\tau,i}^{u}$ given the viewing process history from $\tau - \delta + 1$ up to τ for $\delta > 1$. We note that two nodes in the same tree level may have the same characteristics, but different viewing trajectories.

We propose a predictor which we refer to as the Statistical Model Predictor (*SMP*) that any given slot say τ takes as input the sequence of δ most recently viewed viewports, i.e., $\boldsymbol{V}_{[\tau-\delta+1,\tau]}^{u,*} = (\boldsymbol{v}_{\tau-\delta+1}^{u,*}, \boldsymbol{v}_{\tau-\delta+2}^{u,*}, \ldots, \boldsymbol{v}_{\tau}^{u,*})$, and outputs a statistical

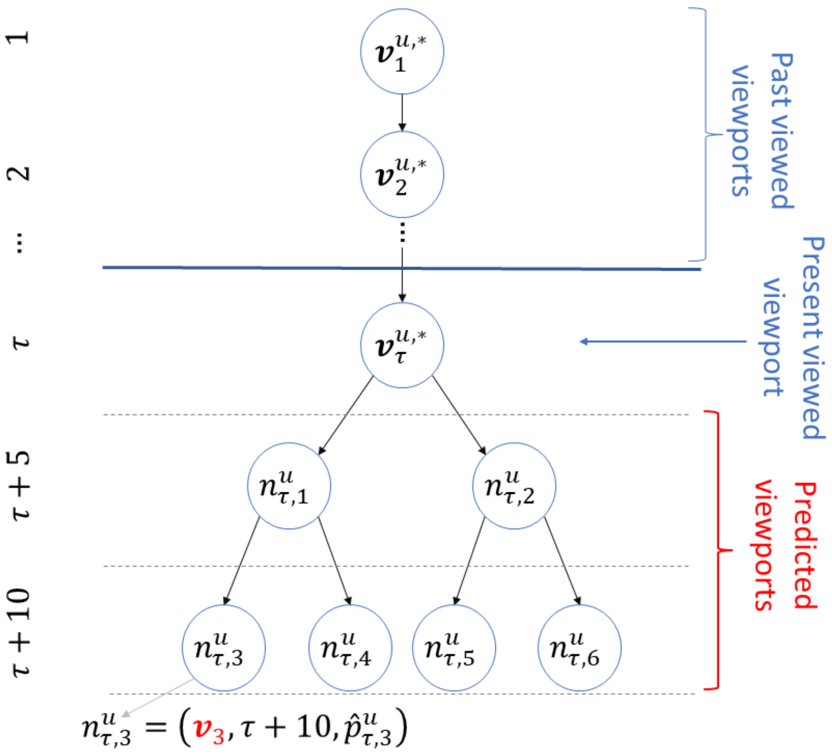


Figure 5.3: Past and present viewed viewports and the estimated statistical model for user u's viewing processes at time $\tau = m$ with $\alpha = 2, \omega = 2$ and $\gamma = 5$.

model for the user's viewing process for slots $\tau + \gamma, \ldots, \tau + \omega\gamma$. Note that the overall prediction window is $\omega\gamma$. Ideally to obtain accurate predictions of the future, ω should be large and γ should be small. The *SMP*'s main block is a Deep Recurrent Neural Network (DRNN) that takes as input a sequence of δ viewports up to the most recent slot τ and estimates the probability of viewing each possible viewport in slot $\tau + \gamma$.

The statistical model is predicted through recursive executions of the DRNN. In slot τ, the first execution of the DRNN estimates the viewports' probabilities of what is viewed in slot $\tau + \gamma$. Next, for each of the α viewports with the highest estimated probabilities of being viewed in slot $\tau + \gamma$ is concatenated to $\boldsymbol{V}^{u,*}_{[\tau-\delta+2,\tau]}$. Each newly formed sequence of viewports is then fed to the DRNN to estimate the statistical model for the viewing process in slot $\tau+2\gamma$. This process is repeated over ω time slots delivering the tree model for the user's viewing process. Details regarding the architecture of the DRNN and *SMP* are provided in Appendix C.1.1. The number of recursive DRNN executions the *SMP* requires to estimate a statistical model of depth ω and branching factor α is thus at most $\frac{\alpha^{\omega}-1}{\alpha-1}$. This structure permits one to increase the value of γ while reducing ω so as to increase the depth of the future predictions while reducing the complexity of the *SMP*.

Recall that a viewport at a given time maps to a set of tile frames. We let $\hat{\mathcal{F}}^u_\tau$ denote the set of tile frames corresponding to the tuples, i.e., viewports at given times, in $\hat{\mathcal{V}}^u_\tau$. Further we let $\hat{\mathcal{T}}^u_\tau$ denote the set of tiles that tile frames in $\hat{\mathcal{F}}^u_\tau$ map to. Additional notation is provided in Table 5.1.

$\mathcal{V}$	Set of viewports/viewing directions.
$\mathcal{F}_\tau^u$	Set of tile frames in the video frame displayed to active user u at time τ.
$\mathcal{T}_\tau^u$	Set of tiles that belong to the video chunk that active user u is watching at time τ.
d_f^u	Deadline of tile frame $f \in \mathcal{F}_\tau^u$. We let $\boldsymbol{d} := (d_f^u : f \in \mathcal{F}_\tau^u, \tau \in \{a^u, \dots, a^{u+w^u}\}, u \in U_\tau)$.
$\boldsymbol{v}_\tau^{u,*}$	Viewport looked at by user u in slot τ.
$h^u(V, \tau)$	Function that maps viewport V that is looked at, or may be looked at by user u in slot τ to a subset of tile frames in $\mathcal{F}_\tau^u$.
$\boldsymbol{V}_{[\tau-n+1,\tau]}^{u,*}$	Sequence of viewports looked at by user u between slots $\tau - n + 1$ and τ.
$\hat{\mathcal{V}}_\tau^u$	Set of nodes in the tree for user u's statistical model as estimated at time τ whence for future slots $\tau + \gamma, \dots, \tau + \omega\gamma$, which comprises of m_τ^u nodes.
$n_{\tau,i}^u = (\hat{\boldsymbol{v}}_{\tau,i}^u, k_{\tau,i}^u, \hat{p}_{\tau,i}^u)$	Node indexed by i and characterized by (1) a predicted viewport $\hat{\boldsymbol{v}}_{\tau,i}^u \in \mathcal{V}$; (2) a slot $k_{\tau,i}^u \in \{\tau + \gamma, \dots, \tau + \omega\gamma\}$; and (3) an estimated probability $\hat{p}_{\tau,i}^u$ given the viewing process history from $\tau - \delta + 1$ up to τ for $\delta > 1$.
$\hat{\mathcal{F}}_\tau^u$	Set of tile frames that viewports and time tuples in $\hat{\mathcal{V}}_\tau^u$ map to. We let $\hat{\boldsymbol{\mathcal{F}}}_\tau := (\hat{\mathcal{F}}_\tau^u : u \in U_\tau)$.
$\hat{\mathcal{T}}_\tau^u$	Set of tiles that tile frames in $\hat{\mathcal{F}}_\tau^u$ map to. We let $\hat{\boldsymbol{\mathcal{T}}}_\tau := (\hat{\mathcal{T}}_\tau^u : u \in U_\tau)$.
$\eta_{\tau,f}^u$	Weight of HD tile frame $f \in \hat{\mathcal{F}}_\tau^u$. We let $\boldsymbol{\eta}_\tau := (\eta_{\tau,f}^u : f \in \hat{\mathcal{F}}_\tau^u, u \in U_\tau)$.
$\lambda_{\tau,t}^u$	Weight of HD tile $t \in \hat{\mathcal{T}}_\tau^u$. We let $\boldsymbol{\lambda}_\tau := (\lambda_{\tau,t}^u : t \in \hat{\mathcal{T}}_\tau^u, u \in U_\tau)$.
$e_{\tau,t}^u$	Deadline of HD tile $t \in \hat{\mathcal{T}}_\tau^u$. We let $\boldsymbol{e}_\tau := (e_{\tau,t}^u : t \in \hat{\mathcal{T}}_\tau^u, u \in U_\tau)$.
$Q_\tau^{u,\pi}$	Set of HD tiles cached in active user u's device buffer at time τ under selection and scheduling policy $\pi \in \Pi$. We let $\boldsymbol{Q}_\tau^\pi := (Q_\tau^{u,\pi} : u \in U_\tau)$.

Table 5.1: Notation.

5.4.2 Weights and deadlines of predicted HD tile frames and tiles

Given a constraint on the resources available for the transmission of HD tiles, the edge can schedule only a subset of the predicted tiles and so the most relevant HD tiles will need to be selected and scheduled. We hence propose to associate a weight with each predicted tile frame based on the probabilities associated with predicted viewports that map to this tile frame, in an approach that would simply capture its priority/importance in comparison with other tile frames. Note that a tile frame may fall in more than one viewport predicted to be viewed by a user in a particular video frame and hence the weight of a tile frame is defined as follows.

Definition 5.4.1 (Weight of a predicted HD tile frame) *The weight of a predicted HD tile frame $f \in \hat{\mathcal{F}}_{\tau}^{u}$ is defined as follows,*

$$\eta_{\tau,f}^{u} = \sum_{i:n_{\tau,i}^{u} \in \hat{\mathcal{V}}_{\tau}^{u}} \frac{p_{\tau,i}^{u}}{|h^{u}(\hat{\boldsymbol{v}}_{\tau,i}^{u}, k_{\tau,i}^{u})|} \mathbb{1}_{\{h^{u}(\hat{\boldsymbol{v}}_{\tau,i}^{u}, k_{\tau,i}^{u})\}}.$$

Note that not all viewports will map to the same number of tile frames in a specific video frame [64].

Remark 5.4.1 *In general, the viewport's probability of being viewed may be distributed non-homogeneously over the tile frames that belong to the viewport, based, for example, on their proximity to the center of the viewport. In Definition 5.4.1, for simplicity we assume that the viewport's probability of being viewed is divided uniformly over all such tile frames.*

In turn, the weight of a tile is formally defined below.

Definition 5.4.2 (Weight of a predicted HD tile) *The weight of an HD tile $t \in \hat{\mathcal{T}}_\tau^u$ is the sum of the weights of all HD tile frames in $\hat{\mathcal{F}}_\tau^u$ that map to tile t and defined as follows,*

$$\lambda_{\tau,t}^u = \sum_{f \in \hat{\mathcal{F}}_\tau^u} \eta_{\tau,f}^u \mathbb{1}_{\{g(f)=t\}}.$$

The viewing deadlines of HD tile frames can be used to define deadlines for HD tiles. The *viewing deadline* of a predicted HD tile is defined below.

Definition 5.4.3 (Viewing deadline of a predicted HD tile) *The viewing deadline of an HD tile $t \in \hat{\mathcal{T}}_\tau^u$ is tied to the earliest tile frame deadline among all tile frames in $\hat{\mathcal{F}}_\tau^u$ that map to t and is given as,*

$$e_{\tau,t}^u = \min_{f \in \hat{\mathcal{F}}_\tau^u : g(f)=t} d_f^u.$$

5.4.3 Our proposed HD tiles scheduling policy π^F

Algorithm 6: π^F

```
Input : $\tau, \epsilon, \omega, \gamma, U_\tau$, $\boldsymbol{\theta}_\tau$, $\rho_\tau$, $\boldsymbol{l}_\tau$, $\boldsymbol{r}^{\pi^F}_{\tau-1}$, $\boldsymbol{Q}^{\pi^F}_\tau$, $\hat{\boldsymbol{\mathcal{F}}}_\tau$, $\hat{\boldsymbol{\mathcal{T}}}_\tau$, $\boldsymbol{d}$, $\boldsymbol{\eta}_\tau$, $\boldsymbol{\lambda}_\tau$, $\boldsymbol{e}_\tau$
Let $S^{\pi^F}_\tau \leftarrow \emptyset$, $B^{\pi^F}_\tau \leftarrow \emptyset$ ;
Let $R^{\pi^F}_{\tau,t} \leftarrow \emptyset$, $M^{\pi^F}_{\tau,t} \leftarrow \emptyset, \forall t \in \hat{\boldsymbol{\mathcal{T}}}_\tau$ ;
Let $L^{\pi^F}_\tau \leftarrow \emptyset$ ▷ *Queue of tuples of HD tiles and their associated users* ;
Let $\tilde{\boldsymbol{\mathcal{T}}}_\tau \leftarrow \hat{\boldsymbol{\mathcal{T}}}_\tau$;
$r^{u,\pi^F}_\tau \leftarrow$
  $(1-\epsilon) r^{u,\pi^F}_{\tau-1} + \epsilon \cdot q\left(\frac{1}{|h^u(V^{u,*}_\tau,\tau)|}\sum_{f \in h^u(V^{u,*}_\tau,\tau)} \mathbb{1}_{\{g(f) \in Q^{u,\pi^F}_\tau\}}\right), \forall u \in U_\tau$
while $\tilde{\boldsymbol{\mathcal{T}}}_\tau$ is non-empty do
    Go to Algorithm 7 ▷ Get $u^*, t^*$
    if $M^{\pi^F}_{\tau,t^*}$ is non-empty then
        if *Adding $u^*$ to $M^{\pi^F}_{\tau,t^*}$ satisfies Constraints 5.1, 5.2, 5.3* then
            $M^{\pi^F}_{\tau,t^*} \leftarrow M^{\pi^F}_{\tau,t^*} \bigcup \{u^*\}$;
            Update $l^{u^*}_\tau$;
            $\tilde{\boldsymbol{\mathcal{T}}}_\tau \leftarrow \tilde{\boldsymbol{\mathcal{T}}}_\tau \setminus \{t^*\}$;
            continue;
    else if $R^{\pi^F}_{\tau,t^*}$ is non-empty then
        if *Adding both $u^*$ and a user $u' \in R^{\pi^F}_{\tau,t^*}$ to $M^{\pi^F}_{\tau,t^*}$ satisfies*
          *Constraints 5.1, 5.2, 5.3* then
            $M^{\pi^F}_{\tau,t^*} \leftarrow M^{\pi^F}_{\tau,t^*} \bigcup \{u^*, u'\}$, $B^{\pi^F}_\tau \leftarrow B^{\pi^F}_\tau \{t^*\}$,
              $R^{\pi^F}_{\tau,t^*} \leftarrow R^{\pi^F}_{\tau,t^*} \setminus \{u'\}$;
            Update $\mu^{\pi^F,u'}_\tau, l^{u^*}_\tau$;
            $\tilde{\boldsymbol{\mathcal{T}}}_\tau \leftarrow \tilde{\boldsymbol{\mathcal{T}}}_\tau \setminus \{t^*\}$;
            continue;
    if *Adding $u^*$ to $R^{\pi^F}_{\tau,t^*}$ satisfies Constraints 5.2, 5.3* then
        $R^{\pi^F}_{\tau,t^*} \leftarrow R^{\pi^F}_{\tau,t^*} \bigcup \{u^*\}$;
        Append $(t^*, u^*)$ to $L^{\pi^F}_\tau$ ;
        Update $\mu^{\pi^F,u^*}_\tau, l^{u^*}_\tau$;
    $\tilde{\boldsymbol{\mathcal{T}}}_\tau \leftarrow \tilde{\boldsymbol{\mathcal{T}}}_\tau \setminus \{t^*\}$;
Tuples in $L^{\pi^F}_\tau$ are popped from the head of the queue. If tile $t$ in the
  tuple $(t, u)$ is in $B^{\pi^F}_\tau$, then it is multicast to all users in $M^{\pi^F}_{\tau,t}$. Else $t$ is
  unicast to user $u$;
```

Algorithm 7: Select the next HD tile and user.

```
  Input  : $\tau, \epsilon, U_\tau$, $\boldsymbol{r}_\tau^{\pi^F}$, $\boldsymbol{Q}_\tau^{\pi^F}$, $\hat{\boldsymbol{\mathcal{F}}}_\tau$, $\tilde{\boldsymbol{\mathcal{T}}}_\tau$, $\boldsymbol{d}$, $\boldsymbol{\eta}_\tau$, $\boldsymbol{\lambda}_\tau$, $\boldsymbol{e}_\tau$
  Output: $u^*, t^*$
1 Evaluate $\hat{\nabla}_{\tau,t}^{u,\pi^F}$ $\triangleright$ Equation (C.2) in Appendix C.2;
2 Let $\hat{\boldsymbol{r}}_\tau^{\pi^F} \leftarrow \boldsymbol{r}_\tau^{\pi^F}$ ;
3 $\hat{r}_{e_{\tau,t}^u-1}^{u,\pi^F}$ is computed for all $t \in \tilde{\mathcal{T}}_\tau^u$, for all $u \in U_\tau$ as;
4 for $i \in \{\tau+1, \cdots, e_{\tau,t}^u - 1\}$ do
5     $\hat{r}_i^{u,\pi^F} \leftarrow (1-\epsilon)\hat{r}_{i-1}^{u,\pi^F} + \epsilon \cdot q\left(\sum_{f \in \hat{\mathcal{F}}_\tau^u} \eta_{\tau,f}^u \mathbb{1}_{\{g(f) \in Q_\tau^{u,\pi^F}, d_f^u = i\}}\right)$
6 $u^*, t^* \leftarrow \underset{u \in U_\tau, t \in \tilde{\mathcal{T}}_\tau^u}{\arg\max}\ \hat{\nabla}_{\tau,t}^{u,\pi^F} \cdot \frac{1}{\hat{r}_{e_{\tau,t}^u-1}^{u,\pi^F}}$
```

Every slot τ, our policy π^F schedules a subset of predicted HD tiles for either unicast or multicast transmissions based on the estimate of the available resources at the BS, namely $\boldsymbol{\theta}_\tau := (\theta_\tau^u : u \in U_\tau)$ and ρ_τ, with the intent of maximizing the multi-user QoE utility while achieving QoE fairness among users.

π^F keeps track of an estimate of the experienced QoE up to time τ for every user $u \in U_\tau$ denoted r_τ^{u,π^F}. Every τ, π^F updates r_τ^{u,π^F} for all $u \in U_\tau$ as a weighted average between $r_{\tau-1}^{u,\pi^F}$ and the concave function $q(\cdot)$ of the fraction of HD tile frames cached in u's device's buffer that fall in the viewport viewed by u in slot τ (Line 5).

π^F next selects an HD tile t^* and the user u^* it has to be sent to (Line 7), and then determines whether t^* can be multicast to u^* and other active users in the set $M_{\tau,t^*}^{\pi^F}$ if Constraints 5.1, 5.2 and 5.3 in Problem 5.3.1 are satisfied (Lines 8-20). Otherwise, π^F unicasts tile t^* to user u^* if Constraints 5.2 and 5.3 in Problem 5.3.1 are satisfied (Lines 23-27). Else, tile t^* is not scheduled for transmission to u^*. A queue $L_\tau^{\pi^F}$ keeps track of the order in which selected tiles will be scheduled for transmission to their corresponding users which is tied to the order in which they were selected (Line 30).

We let $\hat{c}^u_\tau(\cdot)$ be the function that evaluates user u's expected future cumulative QoE over sampled slots $\tau+\gamma,\ldots,\tau+\omega\gamma$, given the set of HD tiles cached in Q^{u,π^F}_τ and let X^u_τ be a random variable that denotes the viewport viewed by user u in slot τ. It follows that

$$\begin{aligned}&\hat{c}^u_\tau(Q^{u,\pi^F}_\tau)\\ &=\sum_{\alpha=1}^{\omega}\hat{\mathbb{E}}_\tau\left[q\left(\frac{1}{|h^u(X^u_{\tau+\alpha\gamma},\tau+\alpha\gamma)|}\sum_{f\in h^u(X^u_{\tau+\alpha\gamma},\tau+\alpha\gamma)}\mathbb{1}_{\{g(f)\in Q^{u,\pi^F}_\tau\}}\right)\right],\end{aligned}$$

where $\hat{\mathbb{E}}_\tau[\cdot]$ denotes the expectation taken with respect to the predicted process over sampled times and given the set of HD tiles cached in Q^{u,π^F}_τ, where $h^u(X^u_{\tau+\alpha\gamma},\tau+\alpha\gamma)$ corresponds to the set of tile frames that fall in viewport $X^u_{\tau+\alpha\gamma}$ in slot $\tau+\alpha\gamma$, and where $g(f)$ maps tile frame f to its corresponding tile.

The marginal increase in the expected future cumulative QoE of user u over sampled slots after scheduling HD tile t to user u and given the set of HD tiles cached in u's device's buffer is denoted as $\hat{\nabla}^{u,\pi^F}_{\tau,t}(Q^{u,\pi^F}_\tau)=\hat{c}^u_\tau(Q^{u,\pi^F}_\tau\bigcup\{t\})-\hat{c}^u_\tau(Q^{u,\pi^F}_\tau)$. For simplicity, we will suppress the argument of $\hat{\nabla}^{u,\pi^F}_{\tau,t}(Q^{u,\pi^F}_\tau)$ and refer to it as $\hat{\nabla}^{u,\pi^F}_{\tau,t}$. (An approximation of $\hat{\nabla}^{u,\pi^F}_{\tau,t}$ is provided in Appendix C.2). We point out that if $q(\cdot)$ is a linear function such that $q(x)=x$, then $\hat{\nabla}^{u,\pi^F}_{\tau,t}=\lambda^u_{\tau,t}$, i.e., it is equivalent to the weight of HD tile t evaluated in slot τ.

In Algorithm 7, π^F greedily selects the tile t^* and the user u^* that maximizes $\hat{\nabla}^{u^*,\pi^F}_{\tau,t^*}/\hat{r}^{u^*,\pi^F}_{e^{u^*}_{\tau,t^*}-1}$ (Line 7), where $\hat{r}^{u^*,\pi^F}_{e^{u^*}_{\tau,t^*}}$ is the estimated QoE of user u^* at the deadline of t^*, i.e., at $e^{u^*}_{\tau,t^*}$, which is geared at achieving a fair allocation of QoE across users. A detailed explanation of the derivation of $\hat{r}^{u,\pi^F}_{e^{u^*}_{\tau,t^*}}$ is relegated to Appendix C.3.

Remark 5.4.2 *π^F schedules HD tiles in a manner similar to the traditional proportional fairness [78] with the main difference that π^F's goal is to maximize the multi-user system utility instead of maximizing the sum of the logarithmic values of the rate allocations of all the users.*

5.5 Simulations

We conducted simulation based evaluations to explore the performance of our proposed policy *SMP*+π^F. We evaluate *SMP*+π^F's performance with respect to other baseline policies which we introduce in Subsection 5.5.4. We assume in our simulations that all LD tiles are successfully delivered to the active users' headsets prior to their deadlines and only focus on scheduling HD tiles. We further assume that the scheduled HD tiles under any policy are successfully transmitted to their corresponding users, i.e., the prediction of network resources is accurate.

5.5.1 Dataset Preparation

We consider the dataset made available by [79] which provides the 3DoF poses of 50 different users tracked while they were watching HD 360° VR videos from a catalog of 10 videos. The videos are encoded at 30 fps, are 60 seconds long and have a 4K resolution each. The equirectangular (EQR) projection of each of the video frames is divided into 200 square tiles of 192x192 pixels. The dataset also provides the ground-truth mapping of the users' 3DoF poses in each video frame to the corresponding set of observed tile frames. For each user and each 360° VR video, there is hence a sequence of 1800 3DoF poses and the corresponding tile frames they map to. We consider 35 users from the dataset for training.

5.5.2 Video segmentation

We assume that each 360° VR video is segmented into video chunks of 1 second each. It follows that each tile in a video chunk comprises of 30 tile frames.

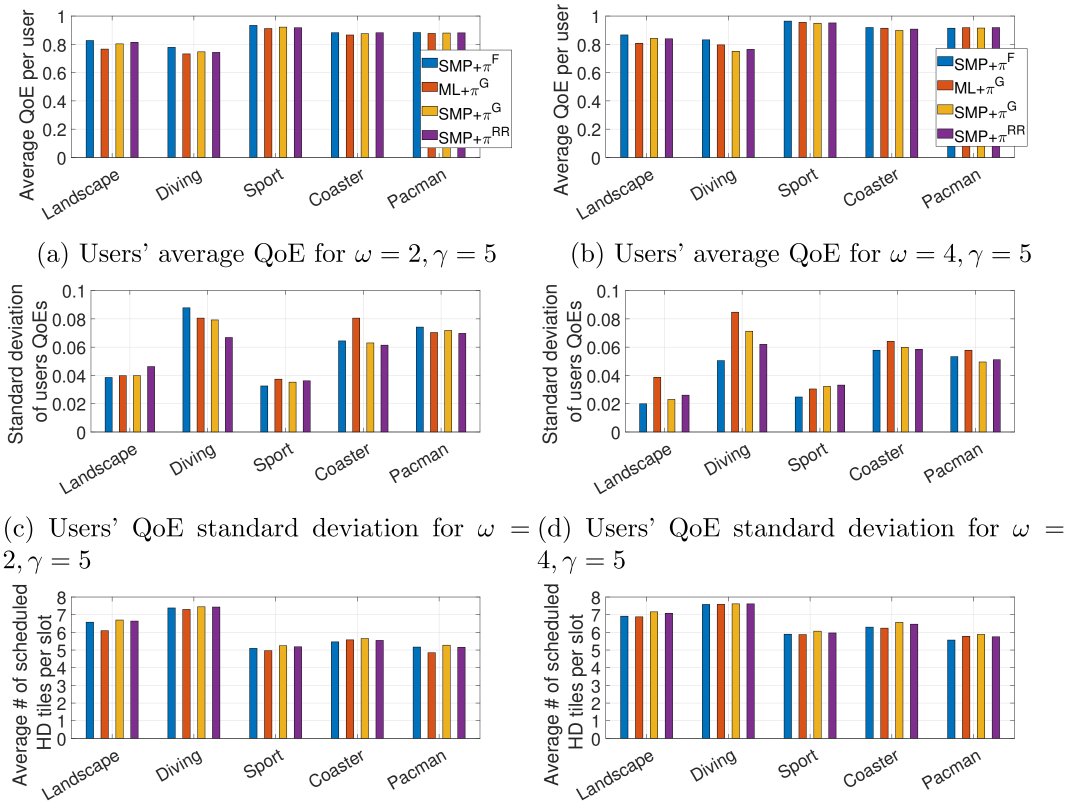
5.5.3 *SMP* parameters

We set $\alpha = 2, \gamma = 5$ and $\omega = 2$ and 4.

5.5.4 Baseline algorithms

In addition to *SMP*+π^F we consider three other baseline algorithms, namely *ML*+π^G, *SMP*+π^G and *SMP*+π^{RR}. All baseline policies differ from *SMP*+π^F in the HD tile and user selection, i.e., the way they select t^* and u^* in Line 7 of Algorithm 6 but otherwise proceed similarly to *SMP*+π^F. *ML*+π^G additionally differs from *SMP*+π^F in the viewports' prediction methodology.

- *ML*+π^G: Determines among the predicted viewports which is most-likely to be viewed in slots $\tau + \gamma, \ldots, \tau + \omega\gamma$ and uses a greedy scheduling policy, i.e., prioritizes scheduling (1) tiles with earliest deadlines and (2) tiles with the largest weights. Every slot τ, the sequence of viewports most-likely to be viewed in slots $\tau + \gamma, \ldots, \tau + \omega\gamma$ is predicted for all active users. The predictor proposed for this approach is inspired by the DRNN proposed in [57] with



(a) Users' average QoE for $\omega = 2, \gamma = 5$

(b) Users' average QoE for $\omega = 4, \gamma = 5$

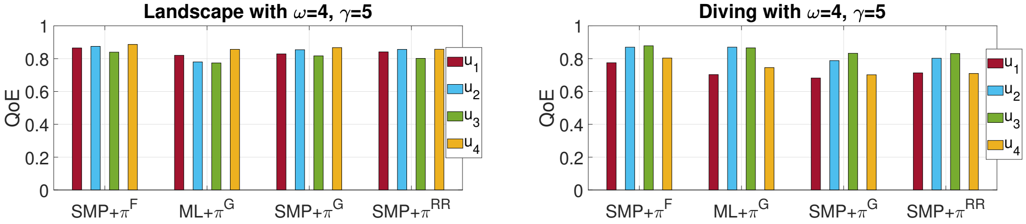
(c) Users' QoE standard deviation for $\omega = 2, \gamma = 5$

(d) Users' QoE standard deviation for $\omega = 4, \gamma = 5$

(e) Average number of HD tiles scheduled at the MEC for $\omega = 2, \gamma = 5$

(f) Average number of HD tiles scheduled at the MEC for $\omega = 4, \gamma = 5$

Figure 5.4: Users' average QoE and standard deviation as well as average number of HD tiles scheduled at the MEC per slot for $\omega = 2, 4$ and fixed $\gamma = 5$, when users are watching the same 360° VR video and experience the same network capacity.



(a) User's QoE results in "landscape" video for $\omega = 4, \gamma = 5$

(b) User's QoE results in "diving" video for $\omega = 4, \gamma = 5$

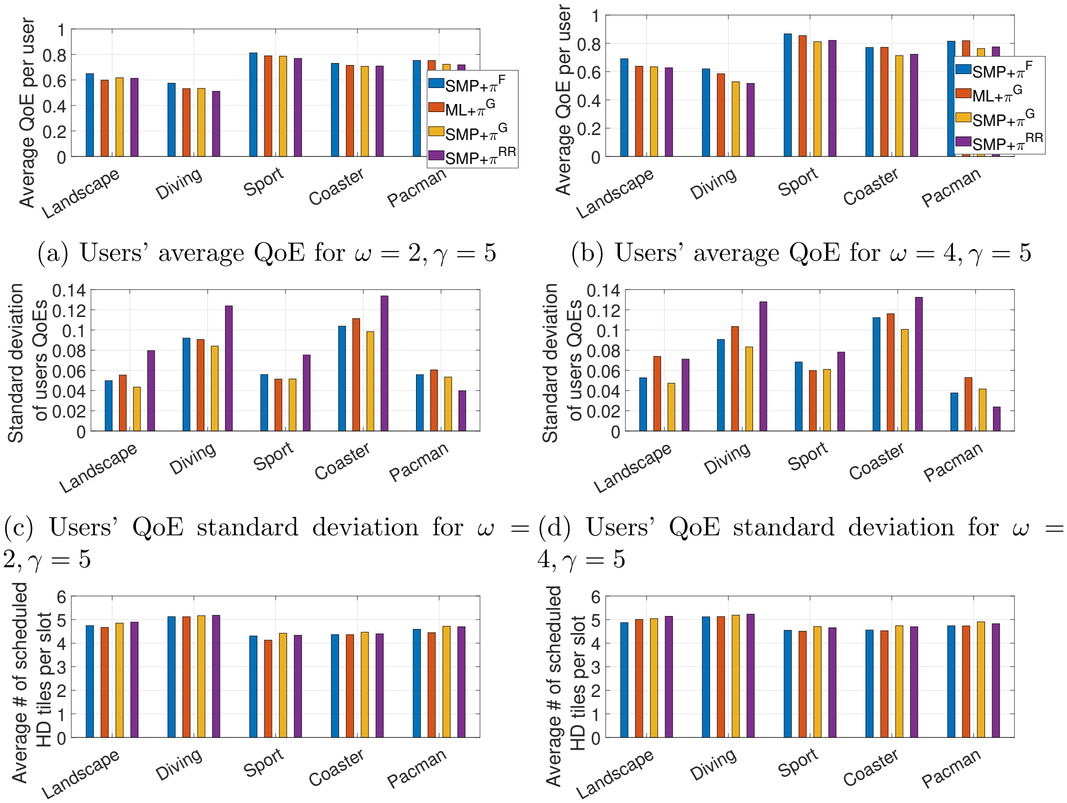
Figure 5.5: Users' QoE results for $\omega = 4$ and $\gamma = 5$ when users are watching "landscape" and "diving" videos respectively while experiencing the same network capacity.

the main difference summarized as follows. The input to the DRNN in [57] consists of the n most recent 3DoF poses and its output consists of the set of tile frames predicted to be viewed in slots $\tau + \gamma, \ldots, \tau + \omega\gamma$, whereas our *ML* predictor consists of a DRNN that takes as an input the n most recently observed viewports and outputs the viewport that is most-likely to be viewed in any of the future slots.

All predicted viewports are then assigned a probability of 1 and the HD tiles' weights and deadlines are evaluated as in Definitions 5.4.2 and 5.4.3 respectively. When selecting a subset of the predicted HD tiles, *ML*+π^G prioritizes the set of predicted tiles with the earliest deadline, then selects for each active user the tile with the largest weight and finally selects the tile t^* and its associated user u^* that maximize $\hat{\nabla}^{u^*,\pi^G}_{\tau,t^*}/\hat{r}^{u^*,\pi^G}_{e^{u^*}_{\tau,t^*}-1}$.

- *SMP*+π^G: Estimates a user's statistical model for its viewing process and selects t^* and u^* under policy π^G.

- *SMP*+π^{RR}: Estimates a user's statistical model for its viewing process and



(a) Users' average QoE for $\omega = 2, \gamma = 5$

(b) Users' average QoE for $\omega = 4, \gamma = 5$

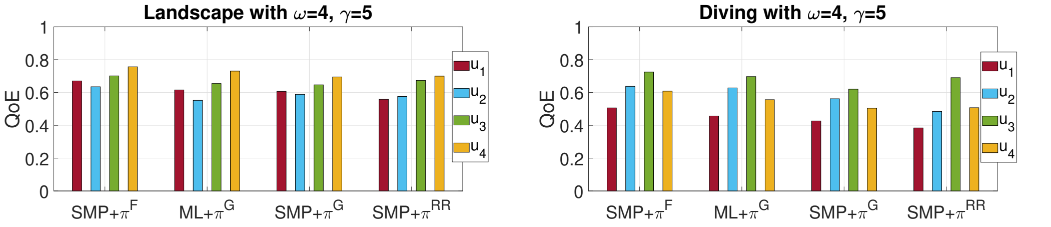
(c) Users' QoE standard deviation for $\omega = 2, \gamma = 5$

(d) Users' QoE standard deviation for $\omega = 4, \gamma = 5$

(e) Average number of HD tiles scheduled at the MEC for $\omega = 2, \gamma = 5$

(f) Average number of HD tiles scheduled at the MEC for $\omega = 4, \gamma = 5$

Figure 5.6: Users' average QoE and standard deviation as well as average number of HD tiles scheduled at the MEC per slot for $\omega = 2, 4$ and fixed $\gamma = 5$, when users are watching the same 360° VR video and experience different network capacities.



(a) User's QoE results in "landscape" video for $\omega = 4, \gamma = 5$

(b) User's QoE results in "diving" video for $\omega = 4, \gamma = 5$

Figure 5.7: Users' QoE results for $\omega = 4$ and $\gamma = 5$ when users are watching "landscape" and "diving" videos respectively while experiencing different network capacities.

selects active users in a round-robin-like fashion then selects for each selected user u^* the HD tile t^* with the largest weight.

5.5.5 Network conditions

In our simulation, we will use the traces for users' time-varying capacity provided in [80], which were collected by [81]. These provide millisecond level wireless capacity accuracy. We use the file named "*Verizon-LTE-driving.down*" which provides traces for around 1400 seconds, and evaluate the maximal transmission rate for every slot τ, i.e., over successive 33 milliseconds time windows.

5.5.6 Model

We consider a setting where 4 randomly selected users from the testing dataset $u_1, \ldots, u_4$, start watching 360° VR videos at the same time and for a duration of 30 seconds. We thus simulate our results for the first 30 seconds of any VR video that the users are watching. We further assume that each HD tile has a size of 0.1 Mbits and set $\epsilon = 10^{-3}$. We finally let $q(\cdot)$ be a linear function where

$q(x) = x$. We provide bar plots for the users' QoE results, users' average QoE and QoE standard deviation achieved under the proposed policies. We further provide bar plots for the average number of HD tiles scheduled per slot by each policy.

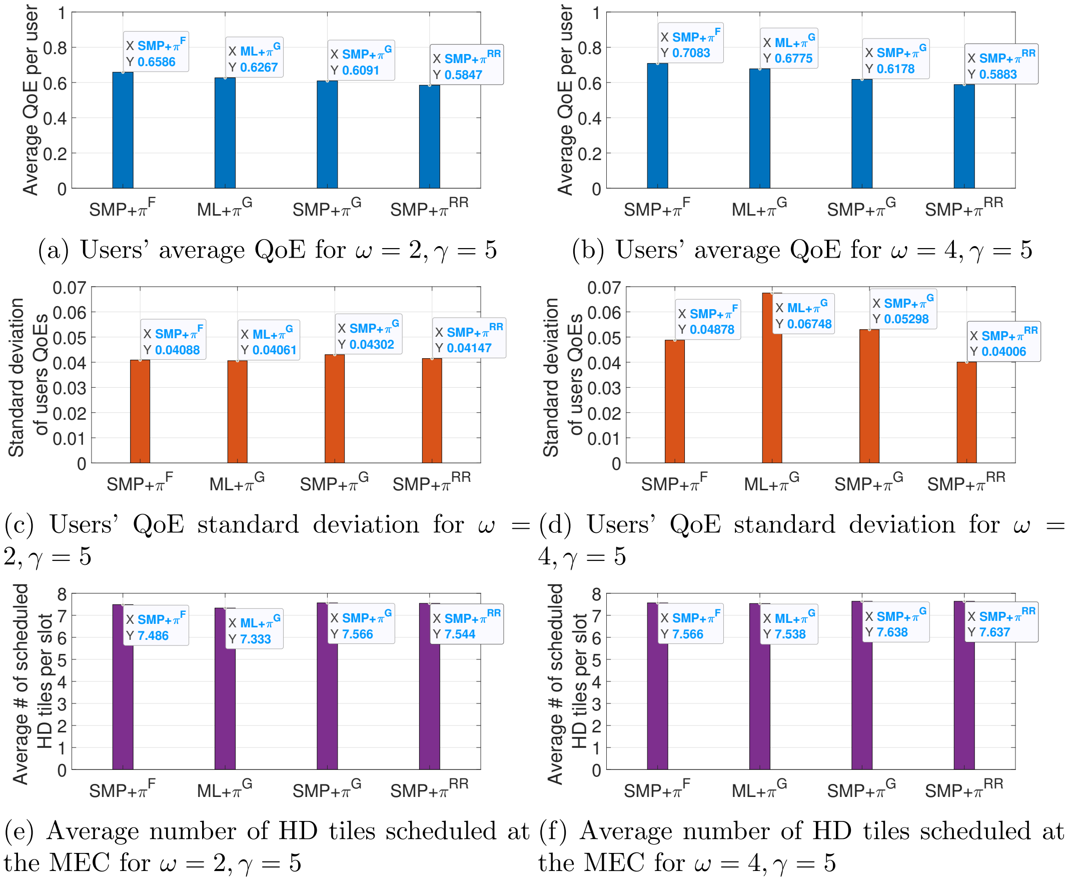
5.5.7 Users watching the same 360° VR video

We first consider the setting where all active users are watching the same 360° VR video and we provide results for 5 different VR videos, namely "landscape", "diving", "sport", "coaster" and "pacman".

5.5.7.1 Homogeneous network capacity

We assume that the maximal number of bits available for the transmission of HD tiles is the same for all active users throughout their VR sessions, i.e., $\theta_\tau^{u_i} = \theta_\tau$ for all τ and for all $i \in \{1, \cdots, 4\}$. We further set $\rho_\tau = 0.6$ for all τ. We use the network capacity trace from time 800 to 830 seconds since it exhibits fluctuations and has a mean capacity of 25.01 Mbps or an equivalent of 0.833 Mbits per slot. The results are provided in Figure (5.4).

We first observe that as ω increases from 2 to 4, the average number of HD tiles scheduled per slot increases for all VR videos and all proposed policies due to the increased number of predicted HD tiles and the available capacity to support their transmissions (Figs. (5.4e) and (5.4f)). All policies achieve somewhat similar average number of transmissions per slot. This phenomenon is accompanied by an increase in the users' average QoE (Figs. (5.4a) and (5.4b)) which confirms that proactively scheduling HD tiles whose tile frames are predicted to be viewed



(a) Users' average QoE for $\omega = 2, \gamma = 5$

(b) Users' average QoE for $\omega = 4, \gamma = 5$

(c) Users' QoE standard deviation for $\omega = 2, \gamma = 5$

(d) Users' QoE standard deviation for $\omega = 4, \gamma = 5$

(e) Average number of HD tiles scheduled at the MEC for $\omega = 2, \gamma = 5$

(f) Average number of HD tiles scheduled at the MEC for $\omega = 4, \gamma = 5$

Figure 5.8: Users' average QoE and standard deviation as well as average number of HD tiles scheduled at the MEC per slot for $\omega = 2, 4$ and fixed $\gamma = 5$, when users u_1, u_2, u_3 and u_4 are watching different 360° VR videos, namely "diving", "coaster", "sport" and "landscape" respectively and experience same network capacity.

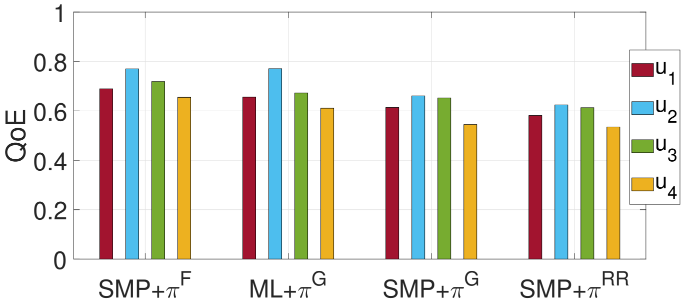
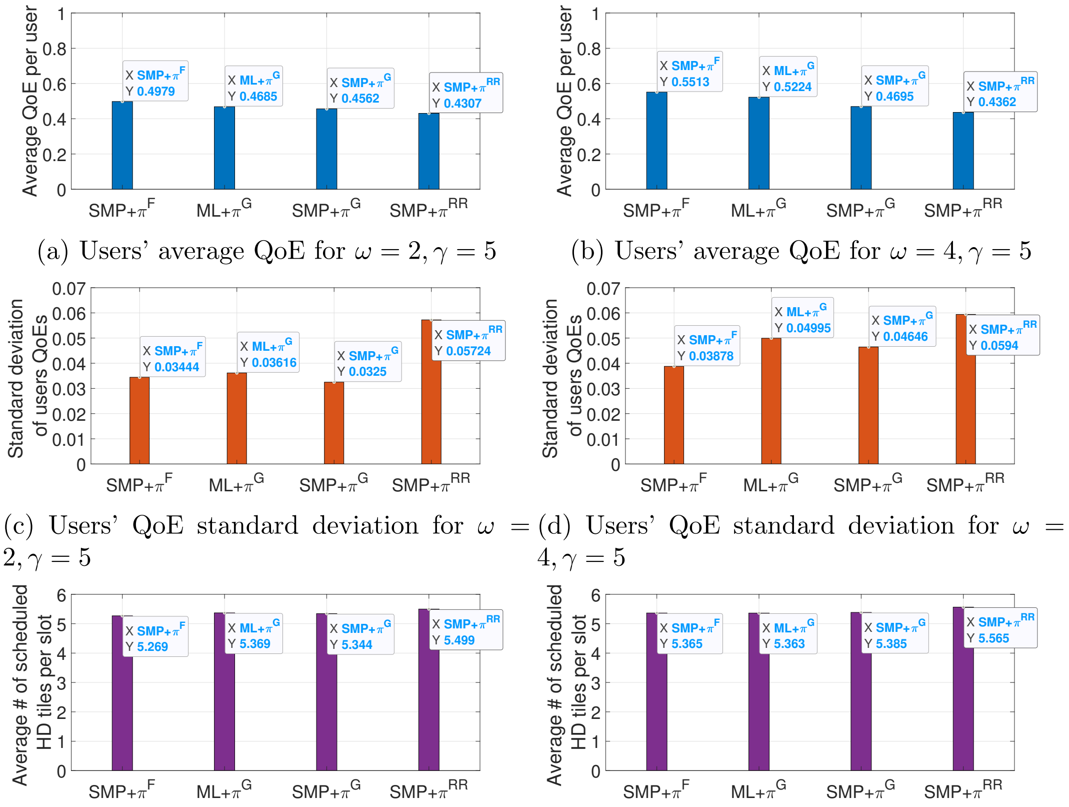


Figure 5.9: Users' QoE results for $\omega = 4$ and $\gamma = 5$ when users u_1, u_2, u_3 and u_4 are watching different 360° VR videos, namely "coaster", "sport", "landscape" and "diving" respectively and experience same network capacity.

in future slots provides robustness to network capacity fluctuations. Further, the increase in ω has resulted in an increase in the number of overlapping predicted HD tiles across users. As a result, SMP+π^F, SMP+π^G and SMP+π^{RR} have leveraged multicast services to achieve a more fair QoE among the users which has resulted in a decrease in the standard deviation of users' QoEs achieved under those policies (Figs. (5.4c) and (5.4d)). On the other hand, we observe that for some videos, the standard deviation of users' QoEs for ML+π^G has increased as ω increased. One can deduce that some users have benefited from the increase in the depth of predictions more than others.

For $\omega = 4$ in Figs. (5.4b) and (5.4d), SMP+π^F achieves an overall higher average QoE and a lower standard deviation of users' QoE as compared to SMP+π^G and ML+π^G. This observation has showcased that neither solely scheduling HD tiles whose tile frames belong to viewports along the most-likely viewport path nor greedily scheduling HD tiles by prioritizing predicted tiles with earliest deadlines have proven to achieve best performance. Since SMP+π^G is expected to achieve



(a) Users' average QoE for $\omega = 2, \gamma = 5$

(b) Users' average QoE for $\omega = 4, \gamma = 5$

(c) Users' QoE standard deviation for $\omega = 2, \gamma = 5$

(d) Users' QoE standard deviation for $\omega = 4, \gamma = 5$

(e) Average number of HD tiles scheduled at the MEC for $\omega = 2, \gamma = 5$

(f) Average number of HD tiles scheduled at the MEC for $\omega = 4, \gamma = 5$

Figure 5.10: Users' average QoE and standard deviation as well as average number of HD tiles scheduled at the MEC per slot for $\omega = 2, 4$ and fixed $\gamma = 5$, when users u_1, u_2, u_3 and u_4 are watching different 360° VR videos, namely "diving", "coaster", "sport" and "landscape" respectively and experience different network capacities.

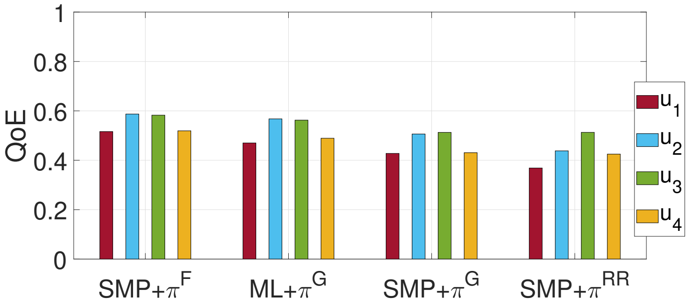


Figure 5.11: Users' QoE results for $\omega = 4$ and $\gamma = 5$ when users u_1, u_2, u_3 and u_4 are watching different 360° VR videos, namely "coaster", "sport", "landscape" and "diving" respectively while experiencing different network capacities.

highest QoE in settings where users have a higher unpredictability in their viewing paths while $ML{+}\pi^G$ is expected to achieve highest QoE in settings where users have high predictability in their viewing paths, we can conclude from our previous observations that some users have experienced throughout their VR sessions high unpredictability in their viewing paths while others have had more predictable paths. In fact, one can clearly see in Fig. (5.5a) that users u_2 and u_3 have had a higher QoE under $SMP{+}\pi^G$ than under $ML{+}\pi^G$, which showcases that u_2 and u_3 have high unpredictability in their viewings in the "landscape" VR video, while users u_1 and u_4 have had high predictable paths which resulted in similar QoE achieved under $SMP{+}\pi^G$ and $ML{+}\pi^G$ for these two users. On the other hand, in Fig. (5.5b), we clearly see that u_2 experienced a higher QoE under $ML{+}\pi^G$ as compared to $SMP{+}\pi^G$, which showcases that u_2 has higher predictability in its viewing process in the "diving" VR video. Another interesting observation in Fig. (5.4d) is that $SMP{+}\pi^G$ achieved a higher standard deviation as compared to $SMP{+}\pi^F$ which confirms that some users have benefited from greedily scheduling their predicted tiles more than others.

The heterogeneity in the unpredictability in users' viewings has boosted the role of *SMP*+π^F in maximizing the average QoE while guaranteeing a fair QoE allocation among all users.

5.5.7.2 Heterogeneous network capacities

We now assume that users experience different variations in their network capacities and that $\theta_\tau^{u_1} = \theta_\tau^{u_2}$ and $\theta_\tau^{u_3} = \theta_\tau^{u_4}$. We further set $\rho_\tau = 0.6$ for all τ. We use the network capacity trace from time 90 to 120 seconds with a mean capacity of 13.94 Mbps or an equivalent of 0.46 Mbits per slot, as the maximal number of bits that can be sent per slot for users u_1 and u_2 and the network capacity trace from time 800 to 830 seconds as the maximal number of bits that can be sent per slot for users u_3 and u_4. The results are provided in Fig. (5.6).

For $\omega = 4$, we observe that the average number of HD tiles scheduled per slot has naturally dropped between Figs. (5.4f) and (5.6f) for all policies and all videos. A drop in the average QoE is observed as well between Figs. (5.4b) and (5.6b). On the other hand, we observe an increase in standard deviation between Figs. (5.4d) and (5.6d) as a result of the different network capacities experienced by the users, which has led to a higher QoE for users with good capacity and lower QoE for users with lower capacity, and has therefore increased QoE unfairness among the active users.

Fig. (5.6b) clearly shows that the users' average QoE achieved under *SMP*+π^F is higher than the one achieved under any other policies, which confirms that

SMP+π^F is robust to the heterogeneity in users' network conditions.

We further observe in Fig. (5.6d) that the standard deviation of users' QoEs achieved under *SMP*+π^{RR} is the worst among all policies followed by *ML*+π^G for similar reasons provided in subsection 5.5.7.1. An interesting observation is that *SMP*+π^G achieves an overall smaller standard deviation in users' QoEs than *SMP*+π^F. This observation is supported in Fig .(5.7).

5.5.8 Users watching different 360° VR videos

We now consider the setting where users u_1, u_2, u_3 and u_4 are watching VR videos "coaster", "sport", "landscape" and "diving" respectively. Such a setting explores the performance of the proposed scheduling policies when only unicast transmissions can take place.

5.5.8.1 Homogeneous network capacity

We consider the same parameter values as in Subsection 5.5.7.1. The results are provided in Fig. (5.8).

For $\omega = 4$ in Fig. (5.8), we observe that *SMP*+π^F schedules on average the same number of HD tiles per slot but achieves the highest average QoE among all policies and the lowest standard deviation of users' QoEs as compared to the one achieved under *SMP*+π^G and *ML*+π^G. We further provide the results for users' QoE in Fig. (5.9).

5.5.8.2 Heterogeneous network capacity

We consider the same parameter values as in Subsection 5.5.7.2. The results are provided in Fig. (5.10) and Fig. (5.11).

For $\omega = 4$, we clearly observe that $SMP{+}\pi^F$ achieves once again the highest average QoE and the lowest standard deviation of users' QoEs.

A key observation in our results is that $SMP{+}\pi^F$ schedules approximately on average the same number of HD tiles per slot as any of the other proposed baseline policies, but achieves an overall higher average QoE and a lower standard deviation of the users' QoEs in almost all settings. We deduce that $SMP{+}\pi^F$ is robust to the heterogeneity in users' video selections and is robust to both the heterogeneity and the fluctuations in the users' network capacities.

5.6 Analysis of the impact of unpredictability in a user's viewing and network capacity on mean QoE

We consider in this section a single user setting and provide some key intuitive results on the impact of network capacity variability and the uncertainty in the user's viewing process have on the user's mean QoE.

5.6.1 Stationary Viewing Processes

In this subsection, we introduce a family of idealized statistical models which we refer to as stationary viewing processes that we define as follows.

Definition 5.6.1 (Stationary Viewing Process) *User u is said to have a Stationary Viewing Process with parameters $n^u, \boldsymbol{b}^u, \boldsymbol{q}^u$, denoted* $\mathrm{SVP}(n^u, \boldsymbol{b}^u, \boldsymbol{q}^u)$*, with*

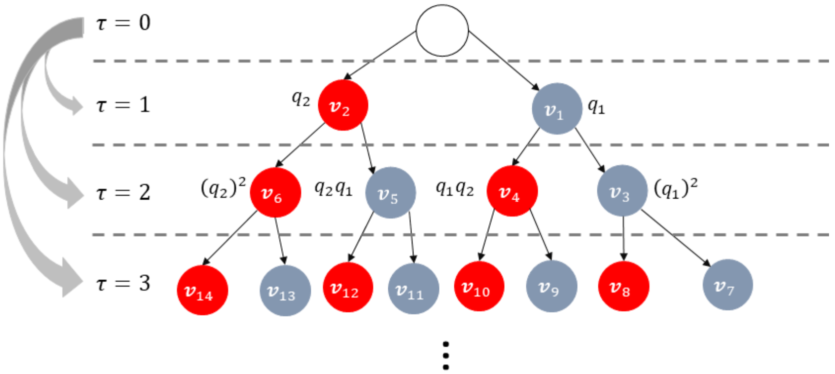


Figure 5.12: Stationary viewing process with $n = 2$.

$\boldsymbol{b}^u := (b_i^u : i \in \{1, \ldots, n^u\})$ *and* $\boldsymbol{q}^u := (q_i^u : i \in \{1, \ldots, n^u\})$*, if at each slot a node branches to* n^u *nodes representing viewports where the size of the tile frames associated with the* i^{th} *node's viewport is* b_i^u *bits for* $i \in \{1, \cdots, n^u\}$*, and the* i^{th} *branch is taken with probability* q_i^u*. Without loss of generality,* $q_1^u \geq q_2^u \geq \ldots \geq q_{n^u}^u$*.*

Figure 5.12 provides an example of a stationary viewing process.

Assumption 5.6.1 *For the remainder of the section, we will assume that each node's viewport maps to a single unique tile frame in the tree which in turn maps to a unique tile in the tree. A viewport's probability of being viewed is hence equal to the weight of the tile frame (and therefore the tile) it maps to (check Definitions 5.4.1 and 5.4.2). We will further assume that all tiles have the same size, i.e.,* $b_i^u = b^u$ *for* $i \in \{1, \ldots, n^u\}$*. With some abuse of notation, we let* $\mathrm{SVP}(n^u, b^u, \boldsymbol{q}^u)$ *denote the stationary viewing process of user* u *for the remainder of the section.*

5.6.2 Typical user's mean QoE and Greedy-by-Level scheduling policy π^G

Given that a user u's viewing process is characterized by an $\mathrm{SVP}(n^u, b^u, \boldsymbol{q}^u)$, one can define the mean QoE of u as follows. Let X_τ^u be a random variable that denotes the viewport viewed by user u in slot τ. With some abuse of notation, and following from Assumption 5.6.1, we let X_τ^u be the random variable that denotes the tile frame/tile that falls in user u's viewport in slot τ. We let $\overline{\mathrm{QoE}}^{u,\pi}$ denote the user's mean QoE achieved under policy π,

$$\overline{\mathrm{QoE}}^{u,\pi} = \frac{1}{w^u}\mathbb{E}\left[\sum_{\tau=1}^{w^u} \mathbb{1}_{\{X_\tau^u \in Q_\tau^{u,\pi}\}}\right]. \tag{5.4}$$

We propose in what follows an HD tile scheduling policy which we refer to as greedy-by-level policy π^G. In a single active user setting, π^G schedules HD tiles associated with the viewports in the user's stationary viewing process by prioritizing (1) tiles with the earliest deadlines and (2) tiles with the largest weights, i.e., probabilities of being viewed. We use π^G as the tile scheduling policy for the remainder of the section due to the simplicity it provides in the theoretical analysis.

Assumption 5.6.2 *For the remainder of the section, we consider a communication constrained setting where the maximal number of bits that can be transmitted to a user u with* $\mathrm{SVP}(n^u, b^u, \boldsymbol{q}^u)$ *does not exceed $n^u b^u$ bits per slot.*

It follows from Assumptions 5.6.1 and 5.6.2 that policy π^G can only schedule in slot τ to user u, tile frames/tiles predicted to be viewed in slot $\tau+1$. As a result, $\mathbb{E}\left[\mathbb{1}_{\{X_\tau^u \in Q_\tau^{u,\pi^G}\}}\right]$ is equal for all τ, since only tiles scheduled to user u in slot $\tau-1$ and available in Q_τ^{u,π^G} will account towards the evaluation of $\mathbb{E}\left[\mathbb{1}_{\{X_\tau^u \in Q_\tau^{u,\pi^G}\}}\right]$. Therefore the mean QoE of a user u with $\mathrm{SVP}(n^u, b^u, \boldsymbol{q}^u)$ achieved under π^G is given as,

$$\overline{\mathrm{QoE}}^{u,\pi^G} = \mathbb{E}\left[\mathbb{1}_{\{X_\tau^u \in Q_\tau^{u,\pi^G}\}}\right]. \tag{5.5}$$

5.6.3 Preliminary analysis

Suppose that all HD tiles scheduled for transmission subject to the available capacity are successfully received at the corresponding user. We provide in what follows two key claims capturing the impact that variability in a user's viewing process as well as variability in the user's available capacity have on its overall VR experience. Under π^G,

- a user experiencing a capacity with lower variability sees better performance.
- a user with lower variability in both its viewing process and capacity sees better performance.

We provide two results that support these claims.

Theorem 5.6.1 *Consider a user u with viewing process modeled as* $\mathrm{SVP}(n^u, b^u, \boldsymbol{q}^u)$ *experiencing a time-varying capacity modeled by i.i.d. random variables with the same distribution as K, such that $K \leq n^u b^u$ almost surely. Let $\mathbb{E}[K] = \kappa$ bits/slot and where $m^* \in \{0, \cdots, n^u\}$ is such that $m^* b^u \leq \kappa < (m^*+1)b^u$. Then $\overline{\mathrm{QoE}}^{u,\pi^G}$ is maximized if*

$$K = \begin{cases} m^* b^u, & \text{w.p. } \frac{(m^*+1)b^u - \kappa}{b^u}, \\ (m^*+1)b^u, & \text{w.p. } 1 - \frac{(m^*+1)b^u - \kappa}{b^u}, \end{cases}$$

The proof of Theorem 5.6.1 is relegated to Appendix C.4.

Theorem 5.6.2 *Consider two users u_1 and u_2 with viewing processes modeled as* $\mathrm{SVP}(n^{u_1}, b^{u_1}, \boldsymbol{q}^{u_1})$ *and* $\mathrm{SVP}(n^{u_2}, b^{u_2}, \boldsymbol{q}^{u_2})$ *respectively, such that $n^{u_1} = n^{u_2} = n, b^{u_1} = b^{u_2} = b$ and $\boldsymbol{q}^{u_1} \succ \boldsymbol{q}^{u_2}$, i.e., for any $m \in \{1, \cdots, n\}$, $\sum_{j=1}^{m} q_j^{u_1} \geq \sum_{j=1}^{m} q_j^{u_2}$. Consider two network regimes with i.i.d. random variables with the same capacity distributions as K_1 and K_2 respectively, such that $0 \leq K_1, K_2 \leq nb$, and $K_2 \geq^{icx} K_1$. Let $\overline{\mathrm{QoE}}_1^{u_1,\pi^G}$ and $\overline{\mathrm{QoE}}_2^{u_2,\pi^G}$ be the mean QoE of users u_1 and u_2 achieved under π^G in the networks with capacity distributions K_1 and K_2 respectively. Then $\overline{\mathrm{QoE}}_1^{u_1,\pi^G} \geq \overline{\mathrm{QoE}}_2^{u_2,\pi^G}$.*

The proof of Theorem 5.6.2 is relegated to Appendix C.5.

Theorem 5.6.1 clearly states that under the assumption of a stationary viewing process and network constraints on both the mean and peak capacity of a user, the mean QoE of the user under policy π^G is maximized if the capacity is constant and equal to its mean. Further, Theorem 5.6.2 suggests that higher predictability in a user's viewing process and lower capacity fluctuations in its network enhances the user's VR performance.

5.7 Chapter Conclusion

In this chapter, we proposed various 360° VR video delivery algorithms based on estimating statistical models for users' future viewing paths. This allows us to consider going beyond myopic approaches that focus only on the next most-likely viewport, and to build strategies that are robust to uncertainty in users' viewing processes, estimation errors and users' capacity variations. We also observe that some video content exhibit higher unpredictability in users' viewing processes which along with the heterogeneity in the users' capacity may lead to unfair QoE allocations across users. To overcome these challenges, we propose and compare various delivery algorithms exhibiting the possible benefits of leveraging a deeper prediction of the future towards achieving improved and fairer QoE across users with time-varying capacities. We further provided insights on the advantages tied to the predictability in a user's viewing process coupled with reduced capacity variations in maximizing a user's VR experience. Such insights open the door towards further exploration of more interactive VR applications where users' future viewing processes are more prone to unpredictability and may lead to a higher number of tile transmissions that do not benefit the users' VR performance, i.e., not actually viewed. A key aspect of our future work lies in the need to devise scheduling policies that account for the depth of users' predictions in such highly unpredictable settings and their impact on both the network utilization as well as on the users' VR experiences.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we explored the advantages of algorithms and metrics addressing the timely sharing of information updates in a multitude of applications that rely on real-time information. We mainly focused on two classes of networks, the first consisted of sensing nodes requesting timely updates to enhance their situational awareness regarding their environments and the second consisted of users sharing constrained resources in multi-user 360° VR applications and proactively sending data regarding their predicted future viewing directions to enhance their QoEs. We showed how different techniques and novel algorithms can be leveraged to optimize resource allocation, situational awareness, overall QoE and QoE fairness among nodes/users in a wide variety of networks and scenarios. We proposed several new approaches towards addressing today's growing problems in wireless sensing systems that service providers could rely on to enhance their users' safety in navigation applications and users' QoEs in the infotainment industry.

Several valuable insights can be drawn from the two parts of this thesis. In the first part, we demonstrated that there is potentially great value in enabling sensor nodes (road-side units, vehicles) to share timely information on the state of a dynamic environment in order to reduce situational uncertainty so as to enhance vehicular flow and safety. We further established that considerable gains are achievable in terms of (1) improved network situational awareness in heterogeneous sensor nodes experiencing time-varying network capacity and topology, and (2) improved matching between producers and consumers of information in non-homogeneous spatial interest settings.

In the second part of this thesis, we showcased the importance of characterizing the perceptual relevance of users' predicted views in 360° VR video applications through the estimation of their probabilities of being viewed. Such characterization allowed for a proactive scheduling and caching of views that are most likely to be viewed by users which has led to enhanced VR performance.

6.2 Future Work

The work conducted in this thesis contributes to the current state of the art research whose focus lies in sensing applications and opens new doors for further investigation in what is likely to be a vast field of future applications. Sensing technologies promise to be fundamental in shaping the capabilities and advancements in both 6G wireless networks and the autonomous cars/robots towards enabling safer, more efficient and intelligent systems, as well as enhanced user experience. We propose in what follows three directions of interest for future research.

First, collaborative sensing systems in real-world settings such as in autonomous vehicular applications involve a higher degree of motion tied to the dynamic aspect of the sensing nodes. Some of the theoretical analysis in this thesis, even if mainly proven for static and fixed sets of sensing nodes, paves the way towards investigating more practical settings that account for the interplay between both dynamic spatial and temporal dimensions. A first attempt at capturing such dynamic aspects in both the heterogeneity in the arrival time of requests and the finite time window in which they can be updated was conducted in this thesis which builds the foundation towards further generalization to more real-life scenarios.

Second, throughout this thesis we have been drawing on objective models tied to the delay and "age" of the timely information. A key area of future work is tied to the design of better metrics that capture the value of timely information, for example in terms of saving time, energy, relevance of the information, increasing safety on a transportation network, etc. Better metrics would allow for more relevant direct optimization of such systems. However, such connections/metrics are very difficult to design, hence the focus of this thesis on more natural performance metrics.

Third, the VR/AR field is gaining increasing interest. In particular, interactive VR/AR applications are possibly going to be computationally draining while requiring more communication efficient techniques to allow for the timely sharing of data with the users. The role and optimization of the edge is hence likely to be of increasing interest. Computationally expensive jobs might best be offloaded to the edge which can coordinate intelligently the data transmissions among users and possibly share/cache computational tasks. Further, the needed depth of predictions of users' viewing patterns should be carefully investigated, as users with high unpredictability in their viewings will overload the network in both computations and communication, and may lead to large amount of wasted data. Such observations call for the investigation of the trade-offs between the depth in users' predictions and the associated computation, network utilization and users' VR performance, especially in such applications where large amounts of predicted data has to be constantly streamed.

Appendix

Appendix A

Chapter 2 Proofs

A.1 Proof of Lemma 2.4.1

$$\begin{aligned}
P\big(\sum_{n\in\mathcal{N}} X_n \frac{r_n}{\mu_n} > 1\big) &= P\Big(\sum_{n\in\mathcal{N}} X_n \frac{r_n}{\mu_n} - \mathbb{E}\big[\sum_{n\in\mathcal{N}} X_n \frac{r_n}{\mu_n}\big] > 1 - \mathbb{E}\big[\sum_{n\in\mathcal{N}} X_n \frac{r_n}{\mu_n}\big]\Big) \\
&\overset{\text{(a)}}{=} P\big(\sum_{n\in\mathcal{N}} X_n \frac{r_n}{\mu_n} - \sum_{n\in\mathcal{N}} \frac{\rho_n}{\mu_n} > 1 - \sum_{n\in\mathcal{N}} \frac{\rho_n}{\mu_n}\big) \\
&\overset{\text{(b)}}{\leq} \exp\left(- \frac{2\big(1 - \sum_{n\in\mathcal{N}} \frac{\rho_n}{\mu_n}\big)^2}{\sum_{n\in\mathcal{N}} (\frac{r_n}{\mu_n})^2} \right)
\end{aligned}$$

where,

(a) follows from $\mathbb{E}[X_n] = \frac{\rho_n}{r_n}$, for all $n \in \mathcal{N}$.

(b) Follows from the independence of X_n's and Hoeffding upper bound.

Constraining the upper bound to be less than ϵ results in the constraint given in Equation (2.8).

A.2 Proof of Proposition 2.4.1

One can verify that the cost function is jointly convex in $\boldsymbol{\rho}$ and $\mathbf{r}$. The congestion constraint corresponds to a convex set in both $\boldsymbol{\rho}$ and $\mathbf{r}$. We shall initially relax the constraint $\boldsymbol{\rho} \leq \boldsymbol{r} \leq \boldsymbol{\mu}$ and define the Lagrangian associated with the relaxed Problem 2.4.1.

$$L(\boldsymbol{\rho}, \boldsymbol{r}; \lambda) = \left[\sum_{n \in \mathcal{N}} (\frac{a_n}{\rho_n} + \frac{b_n}{r_n})\right] + \lambda\left(1 - \sum_{n \in \mathcal{N}} \frac{\rho_n}{\mu_n} - \omega \left\|\boldsymbol{r}\right\|_{\boldsymbol{\mu},2}\right)$$

where $\lambda \geq 0$ is the dual variable. Let

$$f(\lambda) \triangleq \min_{\boldsymbol{\rho}, \boldsymbol{r}} L(\boldsymbol{\rho}, \boldsymbol{r}; \lambda).$$

The Lagrangian dual problem is defined by

$$h \triangleq \max_{\lambda \geq 0} f(\lambda)$$

Taking the partial derivative of $L(\boldsymbol{\rho}, \boldsymbol{r}; \lambda)$ w.r.t. ρ_n and r_n respectively and setting them equal to 0 gives

$$\rho_n^* = \sqrt{\frac{a_n \mu_n}{\lambda}},\; r_n^* = \mu_n \sqrt[3]{\frac{b_n \left\|\boldsymbol{r}^*\right\|_{\boldsymbol{\mu},2}}{\lambda \omega \mu_n}}$$

Given optimal $\boldsymbol{r}^*$, we solve for $\left\|\boldsymbol{r}^*\right\|_{\boldsymbol{\mu},2}$

$$\begin{aligned}
r_n^* &= \mu_n \sqrt[3]{\frac{b_n \left\|\boldsymbol{r}^*\right\|_{\boldsymbol{\mu},2}}{\lambda \omega \mu_n}} \\
(\frac{r_n^*}{\mu_n})^2 &= \left(\frac{b_n}{\mu_n \lambda \omega}\right)^{2/3} \left\|\boldsymbol{r}^*\right\|_{\boldsymbol{\mu},2}^{2/3} \\
\sum_n (\frac{r_n^*}{\mu_n})^2 &= (\sum_n (\frac{b_n}{\mu_n})^{2/3}) \left(\frac{1}{\lambda \omega}\right)^{2/3} \left\|\boldsymbol{r}^*\right\|_{\boldsymbol{\mu},2}^{2/3} \\
\left\|\boldsymbol{r}^*\right\|_{\boldsymbol{\mu},2}^2 &= (\sum_n (\frac{b_n}{\mu_n})^{2/3}) \left(\frac{1}{\lambda \omega}\right)^{2/3} \left\|\boldsymbol{r}^*\right\|_{\boldsymbol{\mu},2}^{2/3}
\end{aligned} \tag{A.1}$$

which leads to

$$\left\|\boldsymbol{r}^*\right\|_{\boldsymbol{\mu},2} = \sqrt{\left\|\boldsymbol{b}\right\|_{\boldsymbol{\mu},\frac{2}{3}}} \left(\frac{1}{\omega \lambda}\right)^{1/2}$$

We plug-in ρ_n^* and $\|\boldsymbol{r^*}\|_{\boldsymbol{\mu},2}$ into the constraint function and solve for λ

$$\lambda = \left[\sqrt{\|\boldsymbol{a}\|_{\boldsymbol{\mu},\frac{1}{2}}} + \sqrt{\omega \|\boldsymbol{b}\|_{\boldsymbol{\mu},\frac{2}{3}}}\right]^2$$

We finally find ρ_n^* and r_n^*,

$$\begin{aligned}
\rho_n^* &= \frac{\sqrt{a_n/\mu_n}}{\sqrt{\|\boldsymbol{a}\|_{\boldsymbol{\mu},\frac{1}{2}}} + \sqrt{\omega \|\boldsymbol{b}\|_{\boldsymbol{\mu},\frac{2}{3}}}}\mu_n \ , \\
\|\boldsymbol{r^*}\|_{\boldsymbol{\mu},2} &= \frac{1}{\sqrt{\omega}}\left(\frac{\sqrt{\|\boldsymbol{b}\|_{\boldsymbol{\mu},\frac{2}{3}}}}{\sqrt{\|\boldsymbol{a}\|_{\boldsymbol{\mu},\frac{1}{2}}} + \sqrt{\omega \|\boldsymbol{b}\|_{\boldsymbol{\mu},\frac{2}{3}}}}\right), \text{ and} \\
r_n^* &= \frac{1}{\sqrt{\omega}}\left(\frac{b_n}{a_n}\right)^{\frac{1}{2}}\left(\frac{\|\boldsymbol{b}\|_{\boldsymbol{\mu},\frac{2}{3}}}{b_n/\mu_n}\right)^{\frac{1}{6}}\rho_n^*
\end{aligned}$$

We re-write ρ_n^* and r_n^* in terms of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as defined in Proposition 2.4.1,

$$\rho_n^* = \frac{\alpha_n}{\| \boldsymbol{\alpha} \|_1 + \sqrt{\omega} \| \boldsymbol{\beta} \|_{\frac{4}{3}}}\mu_n, \quad r_n^* = \frac{1}{\sqrt{\omega}}\frac{\beta_n}{\alpha_n}\left(\frac{\| \boldsymbol{\beta} \|_{\frac{4}{3}}}{\beta_n}\right)^{\frac{1}{3}}\rho_n^*,$$

Next, we verify that the solution to the relaxed problem will satisfy the constraints we have relaxed. Recall that $b_n \geq a_n, \forall n \in \mathcal{N}$ and $1 \leq \omega \leq 6$ (or $e^{-72} \leq \epsilon \leq e^{-2}$). Note that

$$\frac{r_n^*}{\rho_n^*} = \frac{1}{\sqrt{\omega}}\frac{\beta_n}{\alpha_n}\left(\frac{\| \boldsymbol{\beta} \|_{\frac{4}{3}}}{\beta_n}\right)^{\frac{1}{3}} \overset{\text{(a)}}{\geq} \frac{1}{\sqrt{\omega}}\frac{\beta_n}{\alpha_n} \geq \frac{1}{\sqrt{\omega}}\sqrt{\frac{b_n}{a_n}} \geq 1,$$

where (a) follows from $\left(\frac{\|\boldsymbol{\beta}\|_{\frac{4}{3}}}{\beta_n}\right)^{\frac{1}{3}}$ being always greater than 1. The above shows that the relaxed constraint is satisfied under specific assumptions.

Appendix B

Chapter 3 Proofs

B.1 Proof of Theorem 3.4.1

Proof B.1.1 *Given the saw-tooth character of the time varying age per sensor and the randomization of the update phases, the age at a random time is given by* $A = \min_{v=1,\dots,n}[A_v]$*, where* $A_v = d + \frac{1}{r_v}U_v$ *and* U_v*, for* $v = 1, \dots, n$ *are i.i.d. Unif*$[0, 1]$ *and independent, and* $\overline{age}(\mathbf{r}) = \mathbb{E}[A]$ *,while* $age_\gamma(\mathbf{r}) = \mathbb{P}(A \geq \gamma)$*. It is easy to see that given* A_v *is a shifted and scaled uniform random variable such that*

$$P(A_v > y) = \begin{cases} 1, & \textit{if } 0 \leq y \leq d, \\ 1 - r_v(y - d), & \textit{if } d \leq y \leq d + \frac{1}{r_v}, \\ 0, & \textit{if } d + \frac{1}{r_v} < y, \end{cases}$$

and given that $P(A > y) = \prod_{v=1}^{n} P\left(A_v > y\right)$*, and further given that* $r_1 \geq r_2 \geq \dots \geq r_n$*, we have*

$$P(A \geq y) = \begin{cases} 1, & \textit{if } 0 \leq y \leq d, \\ \prod_{v=1}^{n} z_v(y), & \textit{if } d < y \leq d + \frac{1}{r_1}, \\ 0, & \textit{if } d + \frac{1}{r_1} < y, \end{cases} \tag{B.1}$$

where $z_i(y) = 1 - r_v(y - d)$ *for* $v = 1, \dots, n$*. Thus the* γ*-age violation function for* γ *such that* $d < \gamma \leq d + \frac{1}{r_1}$ *is given by,*

$$age_\gamma(\mathbf{r}) = \prod_{v=1}^{n} z_v(\gamma).$$

We can compute the average age function as follows

$$\overline{age}(\mathbf{r}) = \int_0^\infty P(A > y)dy = d + \int_d^{d+\frac{1}{r_1}} \prod_{v=1}^{n} z_v(y)\ dy.$$

The result in the theorem follows by further brute force integration polynomial function in the above integration.

B.2 Proof of Corollary 3.4.1

Proof B.2.1 *The first result follows through a straight forward calculation. Specifically since* $\mathbf{r} = (r_v : v = 1, \ldots, n)$*, where* $r_v = r$*, then,*

$$\begin{aligned}\overline{age}(\mathbf{r}) &= \mathbb{E}[\min_{v=1,\ldots,n} A_v] \\ &= d + \frac{1}{r}\mathbb{E}[\min_{v=1,\ldots,n} U_v] = d + \frac{1}{n+1}\frac{1}{r},\end{aligned}$$

where the last equality follows from U_v*'s being i.i.d. and Unif*$[0,1]$*. The result for the* γ*-age violation function* age_γ *follows directly from Theorem 1.*

The second result is a bit more subtle. Suppose that $\mathbf{r}$ *is such that* $r_1 \geq r_2 \geq \ldots \geq r_n,\ \forall v \in V$*, and let* $\bar{\mathbf{r}} = (\bar{r}_v : v = 1, \ldots, n)$ *where* $\bar{r}_v = \bar{r} = \frac{1}{n}\sum_{v=1}^{n} r_v$*, then we will show that*

$$\overline{age}(\bar{\mathbf{r}}) \geq \overline{age}(\mathbf{r}) \geq \overline{age}(n\bar{r}),$$

$$age_\gamma(\bar{\mathbf{r}}) \geq age_\gamma(\mathbf{r}) \geq age_\gamma(n\bar{r}).$$

Consider the average age function, then these inequalities can be rewritten as,

$$\begin{aligned}\mathbb{E}[\underbrace{\min_{v=1,\ldots,n} d + \frac{1}{\bar{r}}U_v}_{A}] &\geq \mathbb{E}[\underbrace{\min_{v=1,\ldots,n} d + \frac{1}{r_v}U_v}_{B}] \\ &\geq \mathbb{E}[\underbrace{d + \frac{1}{n\bar{r}}U}_{C}],\end{aligned}$$

where U *and* U_v *for* $v = 1, \ldots, n$ *are i.i.d. Unif*$[0,1]$ *and independent. We will show that in fact the random variables* A, B *and* C *are stochastically ordered, i.e., such* *that for all x we have,*

$$P(A > x) \geq P(B > x) \geq P(C > x),$$

which implies the desired result for the γ-age violation functions, and then by integrating the desired inequalities for the average age.

Let us first show $P(A > x) \geq P(B > x)$. We shall consider three cases.

Case 1: *Suppose $x \leq d$ then $P(A > x) = 1$ and thus $P(A > x) \geq P(B > x)$.*

Case 2: *Suppose $d < x \leq d + \frac{1}{r_1}$ then,*

$$P(A > x) = \prod_{v=1}^{n} P\left(U_v > \bar{r}(x-d)\right) = (1 - \bar{r}(x-d))^n = a^n$$

$$\begin{aligned} P(B > x) &= \prod_{v=1}^{n} P\left(U_v > r_v(x-d)\right) \\ &= \prod_{v=1}^{n} (1 - r_v(x-d)) = \prod_{v=1}^{n} b_v. \end{aligned}$$

Note that $\sum_{v=1}^{n} b_n = na$, and it is easy to show that the volume of a rectangle whose sides sum to na is maximized when the sides have equal length, i.e., length a, whence it follows $P(A > x) \geq P(B > x)$.

Case 3: *Suppose $d + \frac{1}{r_1} < x$ then $P(B > 0) = 0$. So clearly $P(A > x) \geq P(B > x)$.*

Let us now show $P(B > x) \geq P(C > x)$. We can write these explicitly as follows,

$$P(C > x) = \begin{cases} 1, & \textit{if } 0 \leq x \leq d, \\ 1 - n\bar{r}(x-d), & \textit{if } d \leq x \leq d + \frac{1}{n\bar{r}}, \\ 0, & \textit{if } d + \frac{1}{n\bar{r}} \leq x, \end{cases}$$

and

$$P(B > x) = \begin{cases} 1, & \textit{if } 0 \leq x \leq d, \\ \prod_{v=1}^{n} z_v(x), & \textit{if } d < x \leq d + \frac{1}{r_1}, \\ 0, & \textit{if } d + \frac{1}{r_1} < x, \end{cases}$$

where $z_v(x) = 1 - r_v(x-d)$. *The result follows if we can show that*

$$\prod_{i=v}^{n} z_v(x) \geq 1 - n\bar{r}(x-d) = 1 - (\sum_v r_v)(x-d).$$

It is easy to verify this is the case for $n = 2$, *i.e.,*

$$\begin{aligned} \prod_{i=v}^{2} z_v(x) &= (1 - r_1(x-d))(1 - r_2(x-d)) \\ &= 1 - (r_1 + r_2)(x-d) + r_1 r_2 (x-d)^2 \\ &\geq 1 - (r_1 + r_2)(x-d). \end{aligned}$$

Now multiplying both sides by $1 - r_3(x-d)$, *one obtains*

$$\begin{aligned} \prod_{v=1}^{3} z_v(x) &\geq (1 - (r_1 + r_2)(x-d))(1 - r_3(x-d)) \\ &\geq (1 - (r_1 + r_2 + r_3)(x-d)) + \\ &\qquad + (r_1 + r_2)(x-d))(r_3(x-d)) \\ &\geq 1 - (r_1 + r_2 + r_3)(x-d). \end{aligned}$$

The general result follows similarly.

B.3 Proof of Theorem 3.5.1

Proof B.3.1 *Suppose* $X \subset Y \subset V$. *Note if we add* $Y \setminus X$ *to the subset* X, *it follows that for any location* $x \in R$, *the number of sensors observing* x, *i.e.* n, *must either increase or stay put. Recall that the modified average age function* $b(.)$ *at location* x

is an increasing concave *function of the number of sensors n that see it, given equal rates $\mathbf{r} = (r_i = r,\ i = 1, \ldots, n)$, in particular it is given by $b(n) = \frac{n}{n+1}\frac{1}{r}$ for $n \geq 1$. Hence $u(.)$ must increase, which in turn implies that $u(X) \leq u(Y)$.*

Suppose $X \subset Y \subset V$ and $v \notin Y$ then submodularity requires that,

$$u(X \cup \{v\}) - u(X) \geq u(Y) - u(Y \cup \{v\}).$$

Let $n(x, X)$ denote the number of sensors in X which see location x and define $n(x, Y)$ in the same way for Y. Clearly $n(x, X) \leq n(x, Y)$. Also adding the sensor v can increase these by at most 1. Then using the definition of $b(.)$, for $n = 0$ we have $b(0) = 0$ and it is easy to show that for $n = 0, 1, \ldots,$ we have that $b(n+1) - b(n)$ is increasing in n. Knowing that the weights are positive, we conclude that $u(.)$ is a non-decreasing submodular set function.

B.4 Proof of Theorem 3.5.2

Proof B.4.1 *We first linearly combine both the weighted coverage and normalized utility function through a positive parameter λ as shown below,*

$$\begin{aligned} g^{\lambda}(X) &= \frac{1}{w(C(X))} \sum_{i=1}^{|\mathcal{P}^X|} w(P_i^X) b(\mathbf{r}(P_i^X)) + \lambda w(C(X)) && \text{(B.2)} \\ &= \hat{u}(X) + \lambda w(C(X)). && \text{(B.3)} \end{aligned}$$

where $\hat{u}(X) = \frac{1}{w(C(X))} u(X)$.

Theorem B.4.1 (Characterization of the weighted coverage-normalized utility function) *If $\lambda \geq 0$ then the weighted coverage-normalized utility linear combination function $g^{\lambda}(.)$ satisfies the following properties,* (***Monotonicity)*** *It is monotonically increasing, i.e., if $X \subset Y \subset V$ then $g^{\lambda}(X) \leq g^{\lambda}(Y)$.*

(***Submodularity***) *It is submodular, i.e., if $X \subset Y \subset V$ and $v \notin Y$ then,*

$$g^{\lambda}(X \cup \{v\}) - g^{\lambda}(X) \geq g^{\lambda}(Y \cup \{v\}) - g^{\lambda}(Y). \tag{B.4}$$

The proof is similar to the proof of Theorem 3.5.1.

We can now clearly state our weighted coverage-age optimization problem as follows

Problem B.4.1 (Weighted coverage-normalized utility optimization problem) *The weighted coverage-normalized utility optimization problem is a submodular optimization problem with a cardinality constraint,*

$$S^{*} \in \underset{X \subseteq V}{\arg\max}\{\ g^{\lambda}(X) \mid |X| \leq k\ \}.$$

Such combinatorial problems are NP hard, but may satisfy submodularity properties that make greedy approaches quite effective. Although this is a complex combinatorial problem, the classical greedy algorithm shown in Algorithm 8 panel requires $O(|V|k)$ function evaluations to determine a subset S_k which is $(1 - 1/e)$ constant factor of the optimal [32], i.e.,

$$g^{\lambda}(S_k) \geq \left(1 - \frac{1}{e}\right) g^{\lambda}(S^{\star}) + \frac{1}{e} g(S_0), \tag{B.5}$$

where S_0 is the initial set of selected elements with $g(S_0) = 0$ if $S_0 = \emptyset$.

Algorithm 8: Greedy submodular optimization [32]

```
Let $S_0 = \emptyset$
for $i=0,\ldots,k-1$ do
    $j \leftarrow \arg\max_j g^{\lambda}(S_i \cup \{j\}) - g^{\lambda}(S_i)$ ; $S_{i+1} \leftarrow S_i \cup \{j\}$ ;
end
```

There are computationally less costly possibly distributed versions of the algorithm leveraging random sampling. There is a growing line of work to design possibly distributed algorithms with sub-linear cost which have shown to be be similarly effective [82–88].

The choice of λ in $g^{\lambda}(.)$ is tied to the weighted coverage-normalized utility approach one desires to apply. We provide below a λ value that meets our desired target in maximizing the weighted coverage first then breaking ties by minimizing the weighted age.

Proposition B.4.1 (Characterization of λ for the maximal weighted coverage-minimal weighted average age problem) *Given a subset of sensors $X \subseteq V$ updating at the same rate r, and given that the smallest aggregated consumers' interest measure on a region of interest covered by sensors in X is $w_{min} = \min_{x \in X} w(C(x))$, then for $\lambda = \frac{1}{w_{min}} \frac{1}{6r} + \epsilon$, $\epsilon > 0$ very small, maximizing $g^{\lambda}(.)$ corresponds to first maximizing the weighted coverage of the consumers' regions of interest while minimizing their weighted average age, then to solely minimizing the weighted average age of the covered regions once maximal coverage is achieved.*

We provide a sketch of the proof of this proposition. It is easy to show that for λ satisfying the condition in the theorem, greedily maximizing $g^{\lambda}(.)$ prioritizes selecting a sensor that covers the uncovered region weighted by the smallest consumers' aggregated interest measure over a sensor that covers a previously covered region weighted by the maximal consumers' aggregated interest measure. Once maximal coverage is achieved and given the submodularity properties of the normalized utility *function $\hat{u}(X)$ [1] then the focus switches to greedily maximizing $\hat{u}(X)$ or equivalently minimizing the weighted average age.*

Our proposed Algorithm 1 is greedy in nature. It proceeds with greedily selecting the sensor that provides the largest marginal gains for both weighted coverage and the utility function. We make use of this property in the following lemma.

Lemma B.4.1 *For $\lambda = \frac{1}{w_{min}} \frac{1}{6r} + \epsilon$, $\epsilon > 0$ very small, Algorithm 1 greedily approximates the solution of Problem B.4.1.*

Lemma B.4.1 directly follows from Theorem B.4.1 and Proposition B.4.1. Using the submodular greedy maximization nature of Algorithm 1 and Lemma B.4.1, and given the submodularity property of weighted coverage [89], we lower bound $w(C(S_{k'}))$ as shown below

$$w(C(S_{k'})) \geq \left(1 - \frac{1}{e}\right) w(C(S^*_{k'})). \tag{B.6}$$

As already discussed, Algorithm 1 operates in two phases. Following Lemma B.4.1, the tightest lower bound on $g^\lambda(S_{k'})$ after greedily selecting k' sensors in phase 1 is

$$g^\lambda(S_{k'}) \geq \left(1 - \frac{1}{e}\right) g^\lambda(S^*_{k'}),$$

which by definition of $g^\lambda(X)$ in Eq.(B.2) gives

$$\begin{aligned} \hat{u}(S_{k'}) \geq& \left(1 - \frac{1}{e}\right) \hat{u}(S^*_{k'}) \\ &- \lambda \left[w(C(S_{k'})) - \left(1 - \frac{1}{e}\right) w(C(S^*_{k'})) \right]. \end{aligned} \tag{B.7}$$

[1] $\hat{u}(X)$ is submodular when maximal coverage is achieved, i.e., $w(C(X))$ is maximal and a constant.

Phase 2 of the algorithm consists of greedily maximizing û(.) over the remaining unselected sensors in V, given k' selected sensors achieving maximal weighted coverage. A lower bound on the normalized utility function by the end of the algorithm is

$$\begin{aligned}\hat{u}(S_k) &\geq \left(1-\frac{1}{e}\right)\hat{u}(S_k^*)+\frac{1}{e}\hat{u}(S_{k'})\\ &\geq \left(1-\frac{1}{e}\right)\hat{u}(S_k^*)+\frac{1}{e}\left(1-\frac{1}{e}\right)\hat{u}(S_{k'}^*)\\ &\quad -\frac{1}{e}\lambda\left[w(C(S_{k'}))-\left(1-\frac{1}{e}\right)w(C(S_{k'}^*))\right],\end{aligned}$$

where the first inequality follows from Eq. (B.5) and the second inequality follows from Eq. (B.7). Using the fact that $\hat{a}(X)=-\hat{u}(X)+d+\frac{1}{r}$, we finally derive an upper bound on the normalized weighted average age,

$$\begin{aligned}\hat{a}(S_k) \leq &\left(1-\frac{1}{e}\right)\hat{a}(S_k^*)\\ &+\frac{1}{e}\left[\left(1-\frac{1}{e}\right)\hat{a}(S_{k'}^*)+\frac{1}{e}\left(d+\frac{1}{r}\right)\right]\\ &\quad -\frac{1}{e}\lambda\left[w(C(S_{k'}))-\left(1-\frac{1}{e}\right)w(C(S_{k'}^*))\right],\end{aligned}$$

which concludes the proof of Theorem 3.5.2.

Appendix C

Chapter 5 Proofs and Supplementary Material

C.1 Statistical Model Predictor

C.1.1 Statistical model predictor based on Gated Recurrent Units

We describe in this subsection our proposed Statistical Model Predictor (*SMP*) based on a Deep Recurrent Neural Network (DRNN) architecture with its main block being the Gated Recurrent Unit (GRU), [90] which is a variant of the original Recurrent Neural Networks (RNNs). For more details on the internal operations of a GRU, we refer the reader to [57,90]. Our DRNN architecture is inspired by the DRNN proposed in [57], with two main differences consisting of adding an additional GRU layer and using a Softmax function as the output layer to generate probabilities. Our *SMP* estimates every slot τ each active user's statistical model for its viewing process.

C.1.2 DRNN architecture

We first let Ξ denote the total number of possible viewports. Our proposed DRNN architecture is presented in Figure C.1. We implement it using PyTorch [91].

Input representation. The DRNN takes as input a sequence of δ viewports. *Sequence processing.* The first three DRNN layers consist of GRU layers with δ cells of 512 memory units each per layer. $\boldsymbol{V}^{u}_{[\tau-\delta+1:\tau]}$ is processed through the first GRU

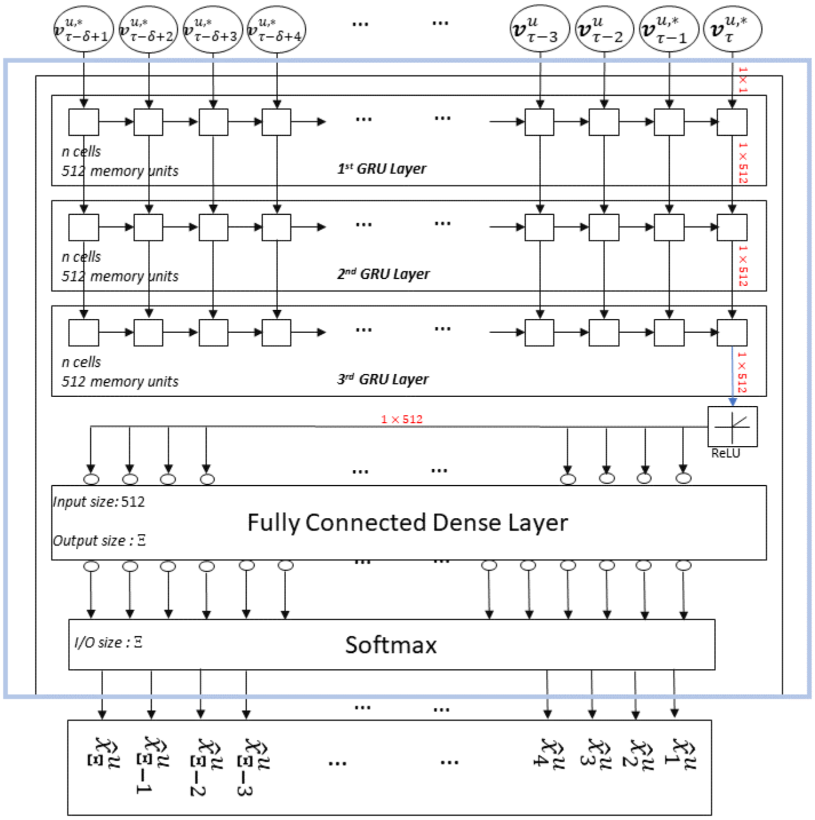


Figure C.1: Our proposed DRNN takes as input in every slot τ a sequence of δ viewports and outputs a vector $\hat{\boldsymbol{x}}^u$ which corresponds to the probability of viewing each viewport in a set of Ξ viewports.

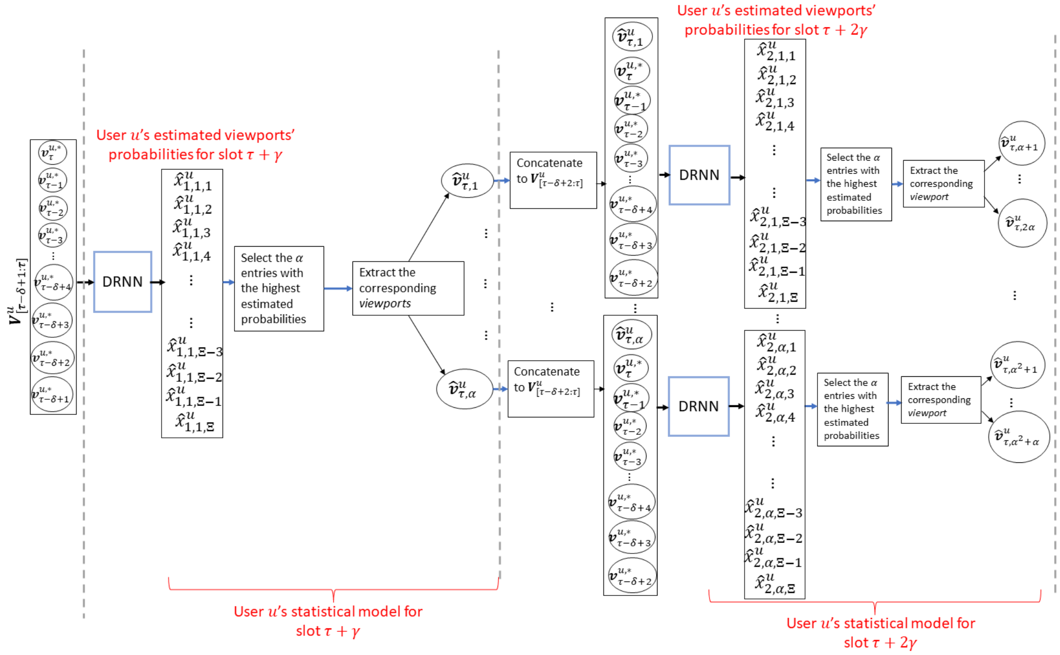

$\boldsymbol{V}^u_{[\tau-\delta+1:\tau]}$
$\boldsymbol{v}^{u,*}_{\tau}$
$\boldsymbol{v}^{u,*}_{\tau-1}$
$\boldsymbol{v}^{u,*}_{\tau-2}$
$\boldsymbol{v}^{u,*}_{\tau-3}$
$\boldsymbol{v}^{u,*}_{\tau-\delta+4}$
$\boldsymbol{v}^{u,*}_{\tau-\delta+3}$
$\boldsymbol{v}^{u,*}_{\tau-\delta+2}$
$\boldsymbol{v}^{u,*}_{\tau-\delta+1}$
DRNN
User u's estimated viewports' probabilities for slot $\tau + \gamma$
$\hat{x}^u_{1,1,1}$
$\hat{x}^u_{1,1,2}$
$\hat{x}^u_{1,1,3}$
$\hat{x}^u_{1,1,4}$
$\hat{x}^u_{1,1,\Xi-3}$
$\hat{x}^u_{1,1,\Xi-2}$
$\hat{x}^u_{1,1,\Xi-1}$
$\hat{x}^u_{1,1,\Xi}$
Select the α entries with the highest estimated probabilities
Extract the corresponding *viewports*
$\hat{\boldsymbol{v}}^u_{\tau,1}$
$\hat{\boldsymbol{v}}^u_{\tau,\alpha}$
User u's statistical model for slot $\tau + \gamma$
Concatenate to $\boldsymbol{V}^u_{[\tau-\delta+2:\tau]}$
$\hat{\boldsymbol{v}}^u_{\tau,1}$
$\hat{\boldsymbol{v}}^u_{\tau,\alpha}$
$\boldsymbol{v}^{u,*}_{\tau}$
$\boldsymbol{v}^{u,*}_{\tau-1}$
$\boldsymbol{v}^{u,*}_{\tau-2}$
$\boldsymbol{v}^{u,*}_{\tau-3}$
$\boldsymbol{v}^{u,*}_{\tau-\delta+4}$
$\boldsymbol{v}^{u,*}_{\tau-\delta+3}$
$\boldsymbol{v}^{u,*}_{\tau-\delta+2}$
DRNN
User u's estimated viewports' probabilities for slot $\tau + 2\gamma$
$\hat{x}^u_{2,1,1}$
$\hat{x}^u_{2,1,2}$
$\hat{x}^u_{2,1,3}$
$\hat{x}^u_{2,1,4}$
$\hat{x}^u_{2,1,\Xi-3}$
$\hat{x}^u_{2,1,\Xi-2}$
$\hat{x}^u_{2,1,\Xi-1}$
$\hat{x}^u_{2,1,\Xi}$
$\hat{x}^u_{2,\alpha,1}$
$\hat{x}^u_{2,\alpha,2}$
$\hat{x}^u_{2,\alpha,3}$
$\hat{x}^u_{2,\alpha,4}$
$\hat{x}^u_{2,\alpha,\Xi-3}$
$\hat{x}^u_{2,\alpha,\Xi-2}$
$\hat{x}^u_{2,\alpha,\Xi-1}$
$\hat{x}^u_{2,\alpha,\Xi}$
Select the α entries with the highest estimated probabilities
Extract the corresponding *viewport*
$\hat{\boldsymbol{v}}^u_{\tau,\alpha+1}$
$\hat{\boldsymbol{v}}^u_{\tau,2\alpha}$
$\hat{\boldsymbol{v}}^u_{\tau,\alpha^2+1}$
$\hat{\boldsymbol{v}}^u_{\tau,\alpha^2+\alpha}$
User u's statistical model for slot $\tau + 2\gamma$

Figure C.2: *SMP* architecture with $\gamma = 5$ and $\omega = 2$ based on the DRNN architecture presented in Figure C.1.

layer which then feeds its output into the 2^{nd} GRU layer whose output is finally fed into the 3^{rd} GRU layer. The output of the 3^{rd} layer goes through a Rectified Linear Unit (ReLu) which only outputs positive values and sets negative values to 0. The output then goes through a Fully Connected Dense layer which outputs Ξ values.

Output representation. The output of the Fully Connected Dense layer is fed to a Softmax layer which outputs a vector $\hat{\boldsymbol{x}}^u := (\hat{x}^u_\xi : \xi \in \{1, \dots, \Xi\})$ of length Ξ where $\hat{x}^u_\xi$ is the estimated probability that the viewport in the ξ's entry is viewed by user u. We note that all entries of $\hat{\boldsymbol{x}}^u$ sum to 1.

C.1.3 SMP architecture

The *SMP* architecture presented in Figure C.2 estimates in a given slot τ a user's statistical model for its viewing process for slots $\tau+5$ and $\tau+10$, i.e., for $\gamma = 5$ and $\omega = 2$ with at most α viewports branching out of each predicted viewport.

We let $\boldsymbol{V}^{u,*}_{[\tau-\delta+1,\tau]} = (\boldsymbol{v}^{u,*}_{\tau-\delta+1}, \boldsymbol{v}^{u,*}_{\tau-\delta+2}, \ldots, \boldsymbol{v}^{u,*}_{\tau})$ be the input to our *SMP*, which consists of the sequence of δ viewports observed by user u between slots $\tau-\delta+1$ and τ. With a slight abuse of notation, we let $\hat{\boldsymbol{x}}^{u}_{1,1} := (\hat{x}^{u}_{1,1,\xi} : \xi \in \{1, \ldots, \Xi\})$ denote the output vector of the first DRNN which corresponds to the set of probabilities estimated for viewports with deadline at $\tau + 5$. Next, the α entries in $\hat{\boldsymbol{x}}^{u}_{1,1}$ with the highest estimated probabilities are selected and the corresponding viewports are denoted as $\hat{\boldsymbol{v}}^{u}_{\tau,1}, \ldots, \hat{\boldsymbol{v}}^{u}_{\tau,\alpha}$. Each of the predicted viewports is concatenated to the most recent $\delta-1$ entries in $\boldsymbol{V}^{u}_{[\tau-\delta+1:\tau]}$ and the resulting vectors are then fed into new DRNNs. Hence, the probability vectors estimated for slot $\tau + 10$ are $\hat{\boldsymbol{x}}^{u}_{2,1}, \ldots, \hat{\boldsymbol{x}}^{u}_{2,\alpha}$. The above process is repeated ω times which results in user u's desired estimated statistical model for its viewing process.

C.1.4 SMP training

We train our proposed *SMP* in a supervised manner. The ground-truth label $\boldsymbol{y}^{u}_{w} := (y^{u}_{w,\xi} : \xi \in \{1, \ldots, \Xi\})$ for slot $\tau + w\gamma$ is such that $y^{u}_{w,\xi}$ takes a value of 1 if the viewport in the ξ^{th} entry of $\boldsymbol{y}^{u}_{w}$ is viewed by user u in slot $\tau + w\gamma$, and 0 otherwise. It follows that a single entry in $\boldsymbol{y}^{u}_{w}$ for $w \in \{1, \ldots, \omega\}$ takes value 1. Our proposed loss function is based on the cross-entropy loss available in the PyTorch library and is given as follows,

$$\text{Loss} = \frac{\alpha - 1}{\alpha^{\omega} - 1} \sum_{w=1}^{\omega} \sum_{i=1}^{\alpha^{(w-1)}} \sum_{\xi=1}^{\Xi} \frac{\exp(\hat{x}^u_{w,i,\xi})}{\sum_{\xi'=1}^{\Xi} \exp(\hat{x}^u_{w,i,\xi'})} y^u_{w,\xi}.$$

Adam algorithm is used to optimize the gradients. Adam parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with no decay, and the learning rate is set to 0.001. The gradient backpropagation is performed over data batches of size 512 and during 350 training epochs.

C.2 Expected marginal increase in a user's QoE

We first derive the expression for the expected marginal increase in user u's QoE over slot $\tau + \gamma$ if tile $t \in \hat{\mathcal{T}}^u_{\tau}$ is scheduled in slot τ under policy π for transmission to user u. The expression is given in Equation (C.1). The crude approximation in Equation (C.1) is a result of the Taylor expansion of the function $q\left(\frac{1}{|h^u_{\tau+\gamma}(X^u_{\tau+\gamma})|} \sum_{f \in h^u_{\tau+\gamma}(X^u_{\tau+\gamma})} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_{\tau} \bigcup \{t\}\}}\right)$ around $\frac{1}{|h^u_{\tau+\gamma}(X^u_{\tau+\gamma})|} \sum_{f \in h^u_{\tau+\gamma}(X^u_{\tau+\gamma})} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_{\tau}\}}$, under the assumption that $\frac{\mathbb{1}_{\{g(h^u(X^u_{\tau+\gamma}, \tau+\gamma)) \cap \{t\} \neq \emptyset\}}}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|}$ is small, and $q'(\cdot)$ is the first derivative of $q(\cdot)$.

Let $\hat{\nabla}^{u,\pi}_{\tau,t}(Q^{u,\pi}_{\tau})$ denote the expected marginal increase in user u's QoE over sampled slots $\tau + \gamma, \ldots, \tau + \omega\gamma$, if tile $t \in \hat{\mathcal{T}}^u_{\tau}$ is scheduled in slot τ under policy π for transmission to user u, and given the set of HD tiles cached in u's device buffer

$$
\begin{aligned}
&\hat{\mathbb{E}}_\tau \left[q \left(\frac{1}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|} \sum_{f \in h^u(X^u_{\tau+\gamma}, \tau+\gamma)} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_\tau \bigcup \{t\}\}} \right) \right] \\
&- \hat{\mathbb{E}}_\tau \left[q \left(\frac{1}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|} \sum_{f \in h^u(X^u_{\tau+\gamma}, \tau+\gamma)} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_\tau\}} \right) \right] \\
&\approx \hat{\mathbb{E}}_\tau \left[q \left(\frac{1}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|} \sum_{f \in h^u(X^u_{\tau+\gamma}, \tau+\gamma)} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_\tau\}} \right) \right] \\
&\qquad + \hat{\mathbb{E}}_\tau \left[\frac{\mathbb{1}_{\{g(h^u(X^u_{\tau+\gamma}, \tau+\gamma)) \cap \{t\} \neq \emptyset\}}}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|} \cdot q' \left(\frac{1}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|} \sum_{f \in h^u(X^u_{\tau+\gamma}, \tau+\gamma)} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_\tau\}} \right) \right] \\
&\qquad\quad - \hat{\mathbb{E}}_\tau \left[q \left(\frac{1}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|} \sum_{f \in h^u(X^u_{\tau+\gamma}, \tau+\gamma)} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_\tau\}} \right) \right] \\
&= \hat{\mathbb{E}}_\tau \left[\frac{\mathbb{1}_{\{g(h^u(X^u_{\tau+\gamma}, \tau+\gamma)) \cap \{t\} \neq \emptyset\}}}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|} \cdot q' \left(\frac{1}{|h^u(X^u_{\tau+\gamma}, \tau+\gamma)|} \sum_{f \in h^u(X^u_{\tau+\gamma}, \tau+\gamma)} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_\tau\}} \right) \right] \\
&= \sum_{i : n^u_{\tau,i} \in \hat{\mathcal{V}}^u_\tau} p^u_{\tau,i} \mathbb{1}_{\{k^u_{\tau,i} = \tau+\gamma\}} \frac{\mathbb{1}_{\{g(h^u(\hat{\boldsymbol{v}}^u_{\tau,i}, k^u_{\tau,i})) \cap \{t\} \neq \emptyset\}}}{|h^u(\hat{\boldsymbol{v}}^u_{\tau,i}, k^u_{\tau,i})|} \cdot q' \left(\frac{1}{|h^u(\hat{\boldsymbol{v}}^u_{\tau,i}, k^u_{\tau,i})|} \sum_{f \in h^u(\hat{\boldsymbol{v}}^u_{\tau,i}, k^u_{\tau,i})} \mathbb{1}_{\{g(f) \in Q^{u,\pi}_\tau\}} \right).
\end{aligned}
\tag{C.1}
$$

in slot τ, $Q_{\tau}^{u,\pi}$. Then,

$$\begin{aligned}\hat{\nabla}_{\tau,t}^{u,\pi}(Q_{\tau}^{u,\pi}) &= \sum_{\alpha=1}^{\omega} \sum_{i:n_{\tau,i}^{u} \in \hat{\mathcal{V}}_{\tau}^{u}} p_{\tau,i}^{u} \mathbb{1}_{\{k_{\tau,i}^{u}=\tau+\alpha\gamma\}} \frac{\mathbb{1}_{\{g(h^u(\hat{\boldsymbol{v}}_{\tau,i}^{u},k_{\tau,i}^{u}))\cap\{t\}\neq\emptyset\}}}{|h^u(\hat{\boldsymbol{v}}_{\tau,i}^{u}, k_{\tau,i}^{u})|} \\ &\quad \cdot q'\left(\frac{1}{|h^u(\hat{\boldsymbol{v}}_{\tau,i}^{u}, k_{\tau,i}^{u})|} \sum_{f \in h^u(\hat{\boldsymbol{v}}_{\tau,i}^{u},k_{\tau,i}^{u})} \mathbb{1}_{\{g(f)\in Q_{\tau}^{u,\pi}\}}\right) \\ &= \sum_{i:n_{\tau,i}^{u} \in \hat{\mathcal{V}}_{\tau}^{u}} p_{\tau,i}^{u} \frac{\mathbb{1}_{\{g(h^u(\hat{\boldsymbol{v}}_{\tau,i}^{u},k_{\tau,i}^{u}))\cap\{t\}\neq\emptyset\}}}{|h^u(\hat{\boldsymbol{v}}_{\tau,i}^{u}, k_{\tau,i}^{u})|} \\ &\quad \cdot q'\left(\frac{1}{|h^u(\hat{\boldsymbol{v}}_{\tau,i}^{u}, k_{\tau,i}^{u})|} \sum_{f \in h^u(\hat{\boldsymbol{v}}_{\tau,i}^{u},k_{\tau,i}^{u})} \mathbb{1}_{\{g(f)\in Q_{\tau}^{u,\pi}\}}\right). \qquad \text{(C.2)}\end{aligned}$$

Note that if $q(\cdot)$ is a linear function such that $q(x) = x$, then $\hat{\nabla}_{\tau,t}^{u,\pi}(Q_{\tau}^{u,\pi}) = \lambda_{\tau,t}^{u}$, i.e., to the weight of HD tile t evaluated in slot τ.

C.3 Expected QoE of user u

We let $\hat{r}_{i}^{u,\pi}$ for $i \geq \tau + 1$, be the estimate QoE of user u in slot i evaluated based on $\hat{r}_{\tau}^{u,\pi} = r_{\tau}^{u,\pi}$ as well as on both the set of HD tile frames whose tiles are cached in u's device buffer in slot τ and on their weights that account for their probability of being viewed. For every active user $u \in U_{\tau}$ and every predicted HD tile t that has neither been scheduled for transmission yet nor is cached in u's device's buffer, $\hat{r}_{e_{\tau,t}^{u}-1}^{u,\pi}$ is evaluated in Line 5 of Algorithm 7 through iterative computations described as follows. $\hat{r}_{i}^{u,\pi}$ is computed as a weighted average between $\hat{r}_{i-1}^{u,\pi}$ and the concave function $q(\cdot)$ of the sum of the weights of the HD tile frames that are cached in u's device's buffer and with deadlines in slot i.

C.4 Proof of Theorem 5.6.1

Proof C.4.1 *We introduce c_i^u, for $i \in \{0, \cdots, n^u\}$, and we let $c_i^u = i \cdot b^u$. We let $f^{u,\pi^G}(c_i^u) = \sum_{j=0}^{i} q_j^u$, with $q_0^u = 0$, be the function that evaluates the sum of the weights of tiles scheduled under policy π^G and capacity value c_i^u, for $i \in \{0, \cdots, n^u\}$. It should be clear that $f^{u,\pi^G}(c_i^u)$ is increasing in c_i^u, with $f^{u,\pi^G}(0) = 0$ and $f^{u,\pi^G}(n^u b^u) = 1$.*

Given that $q_1^u \geq q_2^u \cdots \geq q_{n^u}^u$, the marginal gain of $f^{u,\pi^G}(c_i^u)$ decreases as c_i^u increases. Therefore, $f^{u,\pi^G}(c_i^u)$ is a discrete concave function of c_i^u.

User u's mean QoE in Eq. (5.5) can be re-written as,

$$\begin{aligned}
\overline{\mathrm{QoE}}^{u,\pi^G} &= \mathbb{E}\left[\mathbb{1}_{\{X_\tau^u \in Q_\tau^{u,\pi^G}\}}\right] \\
&= \sum_{i=0}^{n^u} f^{u,\pi^G}(c_i) P(K \in [c_i^u, c_{i+1}^u)) \\
&= \boldsymbol{f}^{u,\pi^G \intercal} \boldsymbol{x}^u,
\end{aligned}$$

where $\boldsymbol{f}_i^{u,\pi^G} := (f^{u,\pi^G}(c_i^u) : i \in \{0, \cdots, n^u\})$ and $\boldsymbol{x}^u := (P(K \in [c_i^u, c_{i+1}^u)) : i \in \{0, \cdots, n^u\})$.

We formulate the following linear program to solve for the capacity allocation that maximizes $\overline{\mathrm{QoE}}^{u,\pi^G}$.

Problem C.4.1

$$\begin{aligned}
&\max_{\boldsymbol{x}^u, \boldsymbol{c}^u} \boldsymbol{f}^{u,\pi^G \intercal} \boldsymbol{x}^u \\
\text{s.t.} \quad & \boldsymbol{c}^{u\intercal} \boldsymbol{x}^u = \kappa, \\
& \mathbb{1}^\intercal \boldsymbol{x}^u = 1,
\end{aligned}$$

where $\boldsymbol{c}^u := (c_i^u : i \in \{0, \dots, n^u\})$. The first constraint corresponds to $\mathbb{E}[K] = \kappa$ and the second constraint guarantees that $\sum_{i=0}^{n^u} P(K \in [c_i^u, c_{i+1}^u)) = 1$.

Given that $m^* b^u \leq \kappa < (m^*+1)b^u$, for $m^* \in \{1, \cdots, n^u\}$ and that $f^{u,\pi^G}(c_i^u)$ is discrete concave in c_i^u for $i \in \{0, \cdots, n^u\}$, it follows from standard convex optimization tools that setting $\boldsymbol{x}^u = \left[\frac{(m^*+1)b^u - \kappa}{b^u}, 1 - \frac{(m^*+1)b^u - \kappa}{b^u}\right]^\intercal$ and $\boldsymbol{c}^u = [m^* b^u, (m^*+1)b^u]^\intercal$ satisfies the set of constraints and maximizes $\boldsymbol{f}^{u,\pi^G \intercal} \boldsymbol{x}^u$ and therefore maximizes $\overline{\mathrm{QoE}}^{u,\pi^G}$, which concludes the proof.

C.5 Proof of Theorem 5.6.2

The proof of this theorem comprises of three main parts.

Part 1. We first prove that $\overline{\mathrm{QoE}}_j^{u_1,\pi^G} \geq \overline{\mathrm{QoE}}_j^{u_2,\pi^G}$ for $j \in \{1, 2\}$.

We introduce c_i, for $i \in \{0, \cdots, n\}$, and we let $c_i = i \cdot b$. We let $f^{u_1,\pi^G}(c_i) = \sum_{j=0}^{i} q_j^{u_1}$ and $f^{u_2,\pi^G}(c_i) = \sum_{j=0}^{i} q_j^{u_2}$ as the functions that evaluate the sum of weights of tiles scheduled under π^G and capacity value c_i, for $i \in \{0, \cdots, n\}$, for users u_1 and u_2 respectively. It should be clear that $f^{u_l,\pi^G}(c_i)$, for $l \in \{1, 2\}$, is increasing in c_i, with $f^{u_l,\pi^G}(0) = 0$ and $f^{u_l,\pi^G}(nb) = 1$. It follows that $f^{u_l,\pi^G}(c_i) = \sum_{j=0}^{i} q_j^{u_l}$, for $l \in \{1, 2\}$, with $q_0^{u_l} = 0$.

Given that $q_1^{u_l} \geq q_2^{u_l} \cdots \geq q_n^{u_l}$, for $l \in \{1, 2\}$, it follows that the marginal gain of $f^{u_l,\pi^G}(c_i)$ decreases as c_i increases. Therefore, $f^{u_l,\pi^G}(c_i)$, for $l \in \{1, 2\}$, are discrete concave functions of c_i.

Since $\boldsymbol{q}_1 \succ \boldsymbol{q}_2$, it follows that $f^{u_1,\pi^G}(c_i) \geq f^{u_2,\pi^G}(c_i)$, for $i \in \{0, \cdots, n\}$. The mean

QoE of users u_1 and u_2 in Eq. (5.5) are written as follows,

$$\begin{aligned}\overline{\mathrm{QoE}}_j^{u_l,\pi^G} &= \mathbb{E}\left[\mathbb{1}_{\{X_\tau^{u_l}\in Q_\tau^{u_l,\pi^G}\}}\right] \\ &= \sum_{i=0}^{n} f^{u_l,\pi^G}(c_i)P(K_j \in [c_i, c_{i+1})) \\ &= \sum_{i=0}^{n} f^{u_l,\pi^G}(c_i)P(a(K_j) = c_i) \\ &= \mathbb{E}[f^{u_l,\pi^G}(a(K_j))],\end{aligned}$$

where $l, j \in \{1,2\}$ and $a(x) = c_i$ if $x \in [c_i, c_{i+1})$.

By linearity of the expectation, it follows that $\mathbb{E}[f^{u_1\pi^G}(a(K_j))] \geq \mathbb{E}[f^{u_2,\pi^G}(a(K_j))]$, which results in $\overline{\mathrm{QoE}}_j^{u_1,\pi^G} \geq \overline{\mathrm{QoE}}_j^{u_2,\pi^G}$, for $j \in \{1,2\}$. This concludes **Part 1**.

Part 2. *We now prove that for user u_l, for $l \in \{1,2\}$, it holds that $\overline{\mathrm{QoE}}_1^{u_l,\pi^G} \geq \overline{\mathrm{QoE}}_2^{u_l,\pi^G}$ under networks with capacity distributions K_1 and K_2 respectively. We use the same notation as in* **Part 1**.

Since $K_2 \geq^{icx} K_1$ and following from the definition of increasing convex ordering of random variables (See Theorem 3.A.1 in [92]) and the discrete concavity of $f^{u_l,\pi}(K_j)$ for $l, j \in \{1,2\}$, we have that $\mathbb{E}[f^{u_l,\pi^G}(a(K_1))] \geq \mathbb{E}[f^{u_l,\pi^G}(a(K_2))]$, for $l \in \{1,2\}$, which leads to $\overline{\mathrm{QoE}}_1^{u_l,\pi^G} \geq \overline{\mathrm{QoE}}_2^{u_l,\pi^G}$ for $l \in \{1,2\}$. This concludes **Part 2**.

Part 3 *follows from both* **Parts 1** *and* **2** *where*

$$\begin{aligned}\overline{\mathrm{QoE}}_1^{u_1,\pi^G} &= \mathbb{E}[f^{u_1,\pi^G}(a(K_1))] \\ &\geq \mathbb{E}[f^{u_2,\pi^G}(a(K_1))] \geq \mathbb{E}[f^{u_2,\pi^G}(a(K_2))] = \overline{\mathrm{QoE}}_2^{u_2,\pi^G},\end{aligned}$$

where the first inequality follows from **Part 1** *and the second inequality follows from* **Part 2**. *This concludes the proof.*

Bibliography

[1] J. Abou Rahal, G. De Veciana, T. Shimizu, and H. Lu, “Optimizing networked situational awareness,” in *2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. IEEE, 2019, pp. 1–8.

[2] J. Abou Rahal, G. de Veciana, T. Shimizu, and H. Lu, “Optimizing timely coverage in communication constrained collaborative sensing systems,” in *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. IEEE, 2020, pp. 1–8.

[3] ——, “Optimizing timely coverage in communication constrained collaborative sensing systems,” *IEEE Transactions on Control of Network Systems*, 2022.

[4] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, “Vehicle-to-everything (V2X) services supported by LTE-Based systems and 5G,” *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.

[5] N. P. Antzela Kosta and V. Angelakis, *Age of Information: A New Concept, Metric, and Tool.* Foundations and Trends in Networking, 2017, vol. 12.

[6] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *Sensor, Mesh and Ad Hoc Communications and*

Networks (SECON), 8th Annual IEEE Communications Society Conference on. IEEE, 2011, pp. 350–358.

[7] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[8] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, “Minimizing the age of information in broadcast wireless networks,” in *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sept 2016, pp. 844–851.

[9] S. K. Kaul, R. D. Yates, and M. Gruteser, “Status updates through queues,” in *46th Annual Conference on Information Sciences and Systems (CISS)*, March 2012, pp. 1–6.

[10] R. D. Yates and S. K. Kaul, “The age of information: Real-time status updating by multiple sources,” *IEEE Transactions on Information Theory*, pp. 1–1, 2018.

[11] Y. Sun, Y. Polyanskiy, and E. Uysal-Biyikoglu, “Remote estimation of the wiener process over a channel with random delay,” in *IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 321–325.

[12] D. P. Palomar and M. Chiang, “A tutorial on decomposition methods for network utility maximization,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[13] J. Widmer, R. Denda, and M. Mauve, “A survey on TCP-friendly congestion control,” *IEEE Network*, vol. 15, no. 3, pp. 28–37, 2001.

[14] R. Srikant and L. Ying, *Communication networks: an optimization, control, and stochastic networks perspective.* Cambridge University Press, 2013.

[15] K. Nar and T. Başar, “Sampling multidimensional wiener processes,” in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 3426–3431.

[16] P. Kall, S. W. Wallace, and P. Kall, *Stochastic programming.* Springer, 1994.

[17] A. Rauch, F. Klanner, and K. Dietmayer, “Analysis of V2X communication parameters for the development of a fusion architecture for cooperative perception systems,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE.* IEEE, 2011, pp. 685–690.

[18] H. Li and F. Nashashibi, “Multi-vehicle cooperative perception and augmented reality for driver assistance: A possibility to ‘see’through front vehicle,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.* IEEE, 2011, pp. 242–247.

[19] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, “Car2x-based perception in a high-level fusion architecture for cooperative perception systems,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE.* IEEE, 2012, pp. 270–275.

[20] S.-W. Kim, Z. J. Chong, B. Qin, X. Shen, Z. Cheng, W. Liu, and M. H. Ang, “Cooperative perception for autonomous vehicle control on the road: Motiva-

tion and experimental results," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* IEEE, 2013, pp. 5059–5066.

[21] X. Zhao, K. Mu, F. Hui, and C. Prehofer, "A cooperative vehicle-infrastructure based urban driving environment perception method using a D-S theory-based credibility map," *Optik-International Journal for Light and Electron Optics*, vol. 138, pp. 407–415, 2017.

[22] Y. Wang, G. de Veciana, T. Shimizu, and H. Lu, "Performance and scaling of collaborative sensing and networking for automated driving applications," in *Proc. IEEE ICC Workshop 5G and Cooperative Autonomous Driving*, May 2018, pp. 1–6.

[23] Y. Wang, G. de Veciana, T. Shimizu, and H. Lu, "Performance and Scaling of Collaborative Sensing and Networking for Automated Driving Applications," Dec 2018.

[24] A. Kosta, N. Pappas, V. Angelakis *et al.*, "Age of information: A new concept, metric, and tool," *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017.

[25] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM.* IEEE, 2012, pp. 2731–2735.

[26] I. Kadota, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Minimizing the age of information in broadcast wireless networks," in *2016 54th Annual Allerton*

Conference on Communication, Control, and Computing (Allerton). IEEE, 2016, pp. 844–851.

[27] R. D. Yates and S. K. Kaul, “The age of information: Real-time status updating by multiple sources,” *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1807–1827, 2018.

[28] R. Talak, S. Karaman, and E. Modiano, “Optimizing information freshness in wireless networks under general interference constraints,” in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing.* ACM, 2018, pp. 61–70.

[29] H. Tang, P. Ciblat, J. Wang, M. Wigger, and R. Yates, “Age of information aware cache updating with file-and age-dependent update durations,” in *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT).* IEEE, 2020, pp. 1–6.

[30] J. Rahal, G. de Veciana, T. Shimizu, and H. Lu, “Optimizing networked situational awareness,” in *Proc. IEEE WIOPT RAWNET Workshop*, June 2019, pp. 1–8.

[31] A. Prasad, S. Jegelka, and D. Batra, “Submodular meets Structured: Finding Diverse Subsets in Exponentially-Large Structured Item Sets,” *arXiv e-prints*, Nov 2014.

[32] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions—i,” *Mathematical programming*,

vol. 14, no. 1, pp. 265–294, 1978.

[33] M. Jaggi, “Revisiting frank-wolfe: Projection-free sparse convex optimization.” 2013, pp. 427–435.

[34] S. Lacoste-Julien, “Convergence rate of frank-wolfe for non-convex objectives,” *arXiv preprint arXiv:1607.00345*, 2016.

[35] A. A. Bian, B. Mirzasoleiman, J. M. Buhmann, and A. Krause, “Guaranteed non-convex optimization: Submodular maximization over continuous domains,” 2016.

[36] S. S. Panwar, D. Towsley, and J. K. Wolf, “Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service,” *Journal of the ACM (JACM)*, vol. 35, no. 4, pp. 832–844, 1988.

[37] L.-O. Raviv and A. Leshem, “Maximizing service reward for queues with deadlines,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2296–2308, 2018.

[38] J. Liebeherr, D. E. Wrege, and D. Ferrari, “Exact admission control for networks with a bounded delay service,” *IEEE/ACM transactions on networking*, vol. 4, no. 6, pp. 885–901, 1996.

[39] H. Hoang, M. Jonsson, U. Hagstrom, and A. Kallerdahl, “Switched real-time ethernet with earliest deadline first scheduling protocols and traffic handling,” in *Proceedings 16th International Parallel and Distributed Processing Symposium.* IEEE, 2002, pp. 6–pp.

[40] X. Qin, Y. Xia, H. Li, Z. Feng, and P. Zhang, “Distributed data collection in age-aware vehicular participatory sensing networks,” *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14 501–14 513, 2021.

[41] M. Liu, D. Li, H. Wu, F. Lyu, and X. S. Shen, “Real-time search-driven caching for sensing data in vehicular networks,” *IEEE Internet of Things Journal*, 2021.

[42] B. Yao, H. Gao, Y. Zhang, J. Wang, and J. Li, “Maximum aoi minimization for target monitoring in battery-free wireless sensor networks,” *IEEE Transactions on Mobile Computing*, 2022.

[43] C. Li, S. Li, Y. Chen, Y. T. Hou, and W. Lou, “Aoi scheduling with maximum thresholds,” in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 436–445.

[44] C. Li, Q. Liu, S. Li, Y. Chen, Y. T. Hou, and W. Lou, “On scheduling with aoi violation tolerance,” in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–9.

[45] P. Zou, J. Zhang, X. Wei, and S. Subramaniam, “Overage and staleness metrics for status update systems,” *arXiv preprint arXiv:2109.14062*, 2021.

[46] C. Li, Q. Liu, S. Li, Y. Chen, Y. T. Hou, W. Lou, and S. Kompella, “Scheduling with age of information guarantee,” *IEEE/ACM Transactions on Networking*, 2022.

[47] K. Saurav and R. Vaze, “Online energy minimization under a peak age of information constraint,” in *2021 19th International Symposium on Modeling*

and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt). IEEE, 2021, pp. 1–8.

[48] N. Rajaraman, R. Vaze, and G. Reddy, "Not just age but age and quality of information," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1325–1338, 2021.

[49] V. Tripathi, I. Kadota, E. Tal, M. S. Rahman, A. Warren, S. Karaman, and E. Modiano, "Wiswarm: Age-of-information-based wireless networking for collaborative teams of uavs," *arXiv preprint arXiv:2212.03298*, 2022.

[50] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.

[51] C.-L. Fan, W.-C. Lo, Y.-T. Pai, and C.-H. Hsu, "A survey on 360 video streaming: Acquisition, transmission, and display," *Acm Computing Surveys (Csur)*, vol. 52, no. 4, pp. 1–36, 2019.

[52] L. Liu, R. Zhong, W. Zhang, Y. Liu, J. Zhang, L. Zhang, and M. Gruteser, "Cutting the cord: Designing a high-quality untethered vr system with low latency remote rendering," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 68–80.

[53] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Enabling high-quality untethered virtual reality." in *NSDI*, 2017, pp. 531–544.

[54] C. Li, W. Zhang, Y. Liu, and Y. Wang, "Very long term field of view prediction for 360-degree video streaming," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).* IEEE, 2019, pp. 297–302.

[55] L. Xie, X. Zhang, and Z. Guo, "Cls: A cross-user learning based system for improving qoe in 360-degree video adaptive streaming," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 564–572.

[56] X. Hou and S. Dey, "Motion prediction and pre-rendering at the edge to enable ultra-low latency mobile 6dof experiences," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1674–1690, 2020.

[57] C. Perfecto, M. S. Elbamby, J. Del Ser, and M. Bennis, "Taming the latency in multi-user vr 360°: A qoe-aware deep learning-aided multicast framework," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2491–2508, 2020.

[58] S. Zhang, M. Tao, and Z. Chen, "Exploiting caching and prediction to promote user experience for a real-time wireless vr service," in *2019 IEEE Global Communications Conference (GLOBECOM).* IEEE, 2019, pp. 1–6.

[59] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "Vr is on the edge: How to deliver 360 videos in mobile networks," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 30–35.

[60] B. Yu, X. Zhang, and I. You, "Collaborative cache allocation and transmission

scheduling for multi-user in edge computing," *IEEE Access*, vol. 8, pp. 163 953–163 961, 2020.

[61] J. Chakareski, "Vr/ar immersive communication: Caching, edge computing, and transmission trade-offs," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 36–41.

[62] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Edge computing meets millimeter-wave enabled vr: Paving the way to cutting the cord," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[63] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communication, computing and caching for mobile vr delivery: Modeling and trade-off," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[64] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 99–114.

[65] H. Ahmadi, O. Eltobgy, and M. Hefeeda, "Adaptive multicast streaming of virtual reality content to mobile users," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017, pp. 170–178.

[66] X. Liu, Q. Xiao, V. Gopalakrishnan, B. Han, F. Qian, and M. Varvello, "360 innovations for panoramic video streaming," in *Proceedings of the 16th ACM*

Workshop on Hot Topics in Networks, 2017, pp. 50–56.

[67] M. Palash, V. Popescu, A. Sheoran, and S. Fahmy, “Robust 360° video streaming via non-linear sampling,” in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[68] X. Wei, C. Yang, and S. Han, “Prediction, communication, and computing duration optimization for vr video streaming,” *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1947–1959, 2020.

[69] J. Chen, X. Qin, G. Zhu, B. Ji, and B. Li, “Motion-prediction-based wireless scheduling for multi-user panoramic video streaming,” in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[70] J. Chen, F. Qian, and B. Li, “Enhancing quality of experience for collaborative virtual reality with commodity mobile devices,” in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 1018–1028.

[71] ISO. (2019) Dynamic adaptive streaming over http (dash). [Online]. Available: https://www.iso.org/standard/75485.html

[72] I. Koprinska and S. Carrato, “Temporal video segmentation: A survey,” *Signal processing: Image communication*, vol. 16, no. 5, pp. 477–500, 2001.

[73] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, “Adaptive 360-degree video streaming using layered video coding,” in *2017 IEEE Virtual Re-*

ality (VR). IEEE, 2017, pp. 347–348.

[74] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *IEEE Transactions on circuits and systems for video technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[75] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.

[76] D. Lecompte and F. Gabin, "Evolved multimedia broadcast/multicast service (embms) in lte-advanced: Overview and rel-11 enhancements," *IEEE Communications Magazine*, vol. 50, no. 11, pp. 68–74, 2012.

[77] A. Ghosh, J. Zhang, J. G. Andrews, and R. Muhamed, *Fundamentals of LTE.* Pearson Education, 2010.

[78] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[79] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360 video viewing dataset in head-mounted virtual reality," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 211–216.

[80] M. PROJECT, "Mahimahi wireless network traces," https://github.com/ravinet/mahimahi/tree/ma 2019.

[81] K. Winstein, A. Sivaraman, H. Balakrishnan *et al.*, “Stochastic forecasts achieve high throughput and low delay over cellular networks.” in *NSDI*, vol. 1, no. 1, 2013, pp. 2–3.

[82] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, “Distributed submodular maximization: Identifying representative elements in massive data,” in *Advances in Neural Information Processing Systems*, 2013, pp. 2049–2057.

[83] E. Balkanski and Y. Singer, “The adaptive complexity of maximizing a submodular function,” in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing.* ACM, 2018, pp. 1138–1151.

[84] A. Ene and H. L. Nguyen, “Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time,” in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM, 2019, pp. 274–282.

[85] M. Fahrbach, V. Mirrokni, and M. Zadimoghaddam, “Submodular maximization with optimal approximation, adaptivity and query complexity,” *arXiv preprint arXiv:1807.07889*, 2018.

[86] M. Minoux, “Accelerated greedy algorithms for maximizing submodular set functions,” in *Optimization techniques.* Springer, 1978, pp. 234–243.

[87] A. Badanidiyuru and J. Vondrák, “Fast algorithms for maximizing submodular functions,” in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms.* SIAM, 2014, pp. 1497–1514.

[88] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrak, and A. Krause, "Lazier than lazy greedy," in *AAAI Conference on Artificial Intelligence*. AAAI, 2015.

[89] A. Krause and D. Golovin, "Submodular function maximization." 2014.

[90] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[91] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library."

[92] M. Shaked and J. G. Shanthikumar, *Stochastic orders*. Springer, 2007.

Vita

Jean Abou Rahal earned his Scientific Baccalaureate from the Collège des Frères Maristes Champville in Lebanon in June 2013. He then enrolled at the American University of Beirut (AUB) in August 2013 from where he graduated in June 2017 with a Bachelor's degree in Computer and Communications Engineering with high-distinction and a minor in Mathematics. In August 2017, he enrolled at the University of Texas at Austin as a Master's student. He joined Professor Gustavo de Veciana's team as Ph.D. student in January 2018. He earned his Master's degree in May 2020. During his graduate studies, he performed three internships with Qualcomm Inc. focusing on tracking algorithms and designing deep learning algorithms for depth and motions estimation. His research interests include applied probability, designing algorithms for safety-critical applications, optimization and computer vision.

Permanent address: **jeanabourahal@utexas.edu**

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.