DISTRIBUTED CACHING AND LEARNING OVER WIRELESS CHANNELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Büşra Tegin
January 2020

DISTRIBUTED CACHING AND LEARNING OVER WIRELESS CHANNELS
By Büşra Tegin
January 2020

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Tolga Mete Duman (Advisor)

Sinan Gezici

Ayşe Melda Yüksel Turgut

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

DISTRIBUTED CACHING AND LEARNING OVER WIRELESS CHANNELS

Büşra Tegin
M.S. in Electrical and Electronics Engineering
Advisor: Tolga Mete Duman
January 2020

Coded caching and coded computing have drawn significant attention in recent years due to their advantages in reducing the traffic load and in distributing computational burden to edge devices. There have been many research results addressing different aspects of these problems; however, there are still various challenges that need to be addressed. In particular, their use over wireless channels is not fully understood. With this motivation, this thesis considers these two distributed systems over wireless channels taking into account realistic channel effects as well as practical implementation constraints.

In the first part of the thesis, we study coded caching over a wireless packet erasure channel where each receiver encounters packet erasures independently with the same probability. We propose two different schemes for packet erasure channels: sending the same message (SSM) and a greedy approach. Also, a simplified version of the greedy algorithm called the grouped greedy algorithm is proposed to reduce the system complexity. For the grouped greedy algorithm, an upper bound for transmission rate is derived, and it is shown that this upper bound is very close to the simulation results for small packet erasure probabilities. We then study coded caching over non-ergodic fading channels. As the multicast capacity of a broadcast channel is restricted by the user experiencing the worst channel conditions, we formulate an optimization problem to minimize the transmission time by grouping users based on their channel conditions, and transmit coded messages according to the worst channel in the group, as opposed to the worst among all. We develop two algorithms to determine the user groups: a locally optimal iterative algorithm and a numerically more efficient solution through a shortest path problem.

In the second part of the thesis, we study collaborative machine learning (ML)

systems, which is also known as federated learning, where a massive dataset is distributed across independent workers that compute their local gradient estimates based on their own datasets. Workers send their estimates through a multipath fading multiple access channel (MAC) with orthogonal frequency division multiplexing (OFDM) to mitigate the frequency selectivity of the channel. We assume that the parameter server (PS) employs multiple antennas to align the received signals with no channel state information (CSI) at the workers. To reduce the power consumption and hardware costs, we employ complex-valued low-resolution analog to digital converters (ADCs) at the receiver side and study the effects of practical low cost ADCs on the learning performance of the system. Our theoretical analysis shows that the impairments caused by a low-resolution ADC do not prevent the convergence of the learning algorithm, and fading effects vanish when a sufficient number of antennas are used at the PS. We also validate our theoretical results via simulations, and further, we show that using one-bit ADCs causes only a slight decrease in the learning accuracy.

Keywords: Coded caching, erasure broadcast channels, wireless fading channels, distributed machine learning, federated learning, stochastic gradient descent, multipath fading MAC, OFDM, low-resolution ADCs.

ÖZET

KABLOSUZ KANALLAR ÜZERİNDE DAĞITIK ÖNBELLEĞE ALMA VE MAKİNE ÖĞRENMESİ

Büşra Tegin
Elektrik Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Tolga Mete Duman
Ocak 2020

Son yıllarda, kodlanmış önbellekleme ve hesaplama, trafik yükünü azalttığı ve hesaplama yükünü uç cihazlara dağıttığı için oldukça dikkat çekti. Bu problemlerin çeşitli yönlerini ele alan birçok araştırma olsa da hala ele alınması gereken birçok zorluk bulunmaktadır. Özellikle, kablosuz kanallar üzerindeki kullanımları tam olarak anlaşılamamıştır. Bu motivasyon ile bu tez, gerçekçi kanal efektleri ve pratik uygulama kısıtlamaları dikkate alınarak bu iki dağıtılmış sistemi kablosuz kanallar üzerinden ele almaktadır.

Tezin ilk bölümünde, her alıcının paketinin birbirinden bağımsız ve aynı olasılıkla silindiği paket silme kanalı ile kodlanmış önbellekleme üzerinde çalışmaktayız. Paket silme kanalları için aynı mesajı gönderme (SSM) ve açgözlü kodlanmış önbellekleme olmak üzere iki kodlanmış önbelleğe alma şeması önermekteyiz. Ayrıca, sistem karmaşıklığını azaltmak için açgözlü algoritmanın basitleştirilmiş bir versiyonu olan gruplanmış açgözlü algoritmayı da önermekteyiz. Gruplanmış açgözlü algoritmanın iletim hızı için üst sınır elde etmekte ve bu üst sınırın küçük paket silme olasılıkları için simülasyon sonuçlarına çok yakın olduğunu göstermekteyiz. Sonrasında ise ergodik olmayan sönümleme kanalları üzerinde kodlanmış önbellekleme çalıştık. Bir yayın kanalının çok noktaya yayın kapasitesi en kötü kanal koşullarını yaşayan kullanıcı tarafından kısıtlandığı için, kullanıcıları kanal koşullarına göre gruplandırarak iletim süresini en aza indirecek kodlanmış mesajların üretimine olanak sağlayan optimizasyon problemini elde ettik. Bu sayede, her grup için oluşturulan kodlanmış mesajlar bütün kullanıcıların arasındaki en kötüye göre değil, gruptaki en kötü kullanıcının kanal koşullarına göre gönderilmektedir. Kullanıcı gruplarını belirlemek için yerel olarak en uygun yinelemeli algoritma ve en kısa yol problemiyle sayısal olarak daha verimli bir çözüm olmak üzere ki algoritma geliştirdik.

Tezin ikinci bölümünde, büyük bir veri kümesinin bağımsız olarak çalışan makinelere dağıtıldığı, ve her bir bağımsız makinenin kendi veri kümelerine göre yerel gradyan tahminlerini hesapladığı federasyon öğrenimi olarak da bilinen işbirlikçi makine öğrenme (ML) sistemlerini inceledik. Her bir makine hesaplamış olduğu gradyan tahminini kanalın frekans seçiciliğini azaltmak için dikey frekans bölmeli çoğullamalı (OFDM) çok yollu bir sönümlemeli çoklu erişim kanalı (MAC) üzerinden göndermektedir. Makinelerde kanal bilgisi yer almadığından parametre sunucusu (PS) alınan sinyalleri hizalamak için birden fazla anten kullanmaktadır. Güç tüketimini ve donanım maliyetlerini azaltmak için, alıcı tarafında karmaşık değerli düşük çözünürlüklü analog-dijital dönüştürücüler (ADC'ler) kullanmakta; pratik ve düşük maliyetli ADC'lerin sistemin öğrenme performansı üzerindeki etkilerini incelemekteyiz. Teorik analizler ile düşük çözünürlüklü ADC kullanmanın neden olduğu bozuklukların öğrenme algoritmasının yakınsamasını önlemediğini ve PS'de yeterli sayıda anten kullanıldığında sönümleme etkilerinin ortadan kalktığını göstermekteyiz. Ayrıca teorik sonuçlarımızı simülasyonlarla doğrulamakta ve bir bitlik ADC'lerin kullanılmasının öğrenme doğruluğunda çok küçük bir düşüşe sebep olduğunu göstermekteyiz.

Anahtar sözcükler: Kodllanmış önbellekleme, silme yayın kanalı, kablosuz sönümleme kanalları, dağıtılmış makine öğrenimi, federasyon öğrenimi, stokastik gradyan iniş, çok yollu sönmlemeli MAC, OFDM, düşük çözünürlüklü ADC.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Tolga M. Duman for his dedicated help, immense knowledge, motivation, and patience throughout my M.S. study. I would like to thank him for supporting and encouraging my research through insightful discussions and suggestions. Without his precious support, it would not be possible to conduct this research, and I feel very fortunate since he is an excellent advisor and mentor to me.

I would also like to thank my examiners: Prof. Sinan Gezici and Prof. Ayşe Melda Yüksel Turgut for their insightful comments.

I would like to thank all the members of the Bilkent Communication Theory and Application Research (CTAR) Lab, Talha Akyıldız, Mert Özateş, Mahdi Shakiba Herfeh, Mücahit Gümüş, and Sadra Charandabi.

This work was supported by Huawei through a graduate fellowship program, which I gratefully acknowledge this support.

Last but not least, I would like to thank my parents and sisters for their unconditional support and encouragement. I am lucky to have them.

Contents

List of Figures

List of Tables

List of Acronyms

A-DSGD Analog distributed stochastic gradient descent
ADC Analog to digital converter
CLT Central limit theorem
CP Cyclic prefix
CSI Channel state information
DAC Digital to analog converter
D-DSGD Digital distributed stochastic gradient descent
DSGD Distributed stochastic gradient descent
i.i.d. Independent and identically distributed
ICI Inter-carrier interference
MAC Multiple-access channel
MDS Maximum Distance Seperable
MIMO Multiple input multiple output
ML Machine learning
OFDM Orthogonal frequency division multiplexing
PS Parameter server
QESGD Quantized epoch stochastic gradient descent
SNR Signal-to-noise ratio
SSM Sending the same message algorithm
TU Totally unimodular

Chapter 1

Introduction

1.1 Overview

Caching is a strategy to prefetch server's contents at individual user caches during off-peak hours, i.e., when the network is not congested, and to exploit the cache contents during the delivery phase where communication is more expensive. The gain of traditional caching strategies is only due to the local memory of independent users. It has recently been shown that with a novel centralized coded caching scheme, a global caching gain can also be obtained by jointly optimizing the placement and delivery phases along with the usual local caching gain. Further, a decentralized coded caching scheme is developed outperforming the traditional caching strategies without any coordination in the placement phase.

On a different front, the rapid growth of data sensing and collection capability of computation devices facilitates the use of massive datasets enabling machine learning (ML) systems to make more intelligent decisions than ever. However, this growth makes the processing of all the data in a central processor troublesome due to energy inefficiency and privacy concerns. Recently, instead of using a central processor, performing the ML task in a distributed manner where each device connected to the central server over a finite capacity link performs the task on

its local dataset has drawn significant attention.

In this thesis, we investigate both distributed caching and distributed learning algorithms in more realistic scenarios, specifically, we take into account (wireless) channel effects and transmission constraints. For coded caching, we firstly focus on the case where the channel between the users and the server is modeled as a packet erasure channel. Secondly, we follow a coded caching model where the placement phase is performed in a decentralized manner and the delivery phase takes place over a wireless fading channel. Our objective is to study non-ergodic channels and minimize the transmission time with low complexity user grouping approaches. Finally, we study distributed learning algorithms over wireless channels taking into account the channel effects and considering the use of low-resolution analog to digital converters (ADCs) in the receive chain, and show that the convergence of the learning algorithm is guaranteed despite these practical implementation issues.

1.2 Thesis Outline

The thesis is organized into six chapters. In Chapter 2, we overview the concepts of coded caching and coded computing necessary for the rest of the thesis, and provide a detailed literature review.

In Chapter 3, we investigate coded caching over packet erasure channels and present a baseline algorithm along with newly proposed greedy and grouped greedy approaches to create multicast opportunities for erased messages. While grouped greedy coded caching gives slightly higher transmission rates than the greedy algorithm, it may be attractive due to its lower complexity. We also obtain an upper bound on the transmission rate of grouped greedy coded caching, which is tight for small erasure probabilities.

In Chapter 4, we analyze coded caching over non-ergodic fading channels, and propose a locally optimal iterative solution and a more efficient algorithm

through a shortest path problem. The basic objective of all these algorithms is to alleviate the effects of the users experiencing worse channel conditions on the multicast capacity via user grouping. The results demonstrate that user grouping for coded caching over wireless channels is highly advantageous, particularly, when the cache sizes are small.

In Chapter 5, we study distributed learning over wireless channels. Specifically, we consider practical implementation issues as well as wireless channel effects. We study and quantify the performance of a distributed learning system at the wireless edge implemented through an orthogonal frequency division multiplexing (OFDM) based transmission using low cost ADCs at the receiver side. Through analytical results, we show that the convergence of the learning algorithm is guaranteed when the number of receive antennas goes to infinity. We also argue through simulations that even a moderate number of receive antennas is sufficient to obtain a good learning performance.

Finally, in Chapter 6, we present our conclusions and provide directions for future research.

Chapter 2

Preliminaries and Literature Review

In this chapter, we provide the necessary preliminaries and a literature review required for the rest of the thesis. Firstly, coded caching is presented in detail to provide a basis for Chapters 3 and 4. Then, fundamentals of coded computing is explained which is studied in Chapter 5.

The chapter is organized as follows. In Section 2.1, centralized coded caching scheme is presented, while decentralized coded caching is covered in Section 2.2. In Section 2.3, machine learning at the wireless edge is explained. The chapter is concluded with a summary in Section 2.4.

Notation: Throughout the thesis, we will use the notation $[a\ b]$ to indicate the integer set $\{a, \ldots, b\}$ where $a \leq b$, a and b are positive integers, and simply $[b] = [1\ b]$.

2.1 Coded Caching

2.1.1 Centralized Coded Caching

Caching is a strategy to prefetch server's contents at individual user caches during off-peak hours, i.e., when the network is not congested, and to exploit the cache contents when communication is more expensive. Hence, the caching problem can be analyzed in two phases: 1) users prefetch the server's content at their caches during off-peak hours which is called the placement phase, 2) cached content is used along with the server's transmissions to satisfy the users' requests which is called the delivery phase.

Conventionally, caching is considered as a strategy to minimize the number of transmitted bits during the delivery phase by only using transmitted bits and individual cache contents of each user separately without employing any coding for both cache and transmitted contents. Hence, the gain of conventional schemes only depends on the size of local caches of each user, called the local caching gain.

In [1], Maddah-Ali and Nielsen introduced a novel centralized coded caching scheme where a server with N files (each of F bits) connected to K users each with cache capacity of M files through an error-free shared link as shown in Fig. 2.1. During the delivery phase, each user requests a file from the server. The proposed coded caching scheme provides a global caching gain by jointly optimizing the placement and delivery phases along with the usual local caching gain even if there is no cooperation among the users. This scheme aims to construct coded multicast messages to satisfy the demands of each user during the delivery phase. Thus, significantly lower transmission rates than those obtained by conventional uncoded caching are achieved.

In the following, we present an illustrative example of centralized coded caching taken from [1].

Example 1. Consider a system with $N = K = 2$, $M = 1$, where files in the

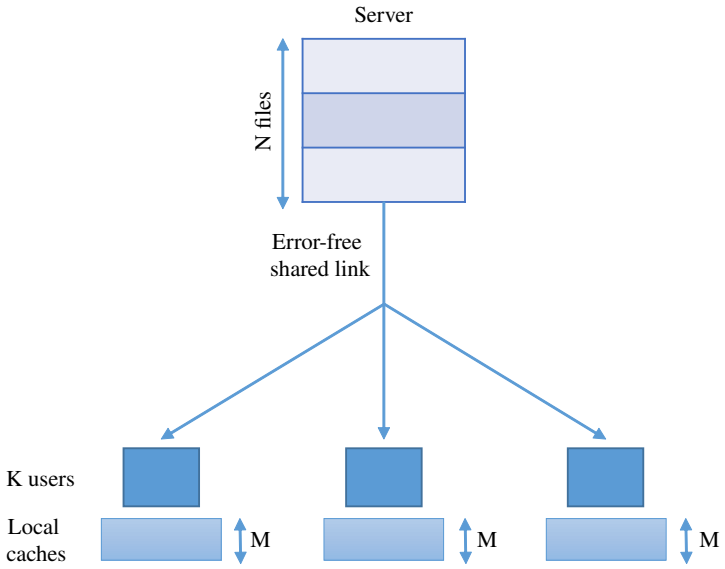

Figure 2.1: System model for centralized coded caching with K users with $M = 1$ local cache memories and a central server with N files.

server are denoted as A and B. Both files are split into equal size two subfiles, i.e., $A = (A_1, A_2)$ and $B = (B_1, B_2)$. The normalized size of each subfile is $MK/N = 1/2$. During the placement phase, user one caches $Z_1 = (A_1, B_1)$ while user two caches $Z_2 = (A_2, B_2)$ in their local caches. Thus, users store $1/2$ of each file exclusively. We can analyze the delivery phase for four different cases as shown in Fig. 2.2.

Case 1: User 1 requests file A while user 2 requests file B. User one already has A_1 in its cache, hence it only needs to receive A_2. User 2 has B_2 which means that it only needs B_1. Also note that, each user has the requested subfile of other user in their own caches. Therefore, reconstruction of the requested files is possible when the server transmits $A_2 \oplus B_1$ whose size is $F/2$ bits, where $\oplus$ represents the bit-wise XOR operation.

Case 2: User 1 requests file B while user 2 requests file A. User one already has B_1 in its cache, hence it only needs to receive B_2. User 2 has A_2, i.e., it only needs A_1. Therefore, users can reconstruct their requested subfile when the server transmits $A_1 \oplus B_2$ whose size is $F/2$ bits.

Case 3: Both users request file A. User one already has A_1 in its cache, i.e.,

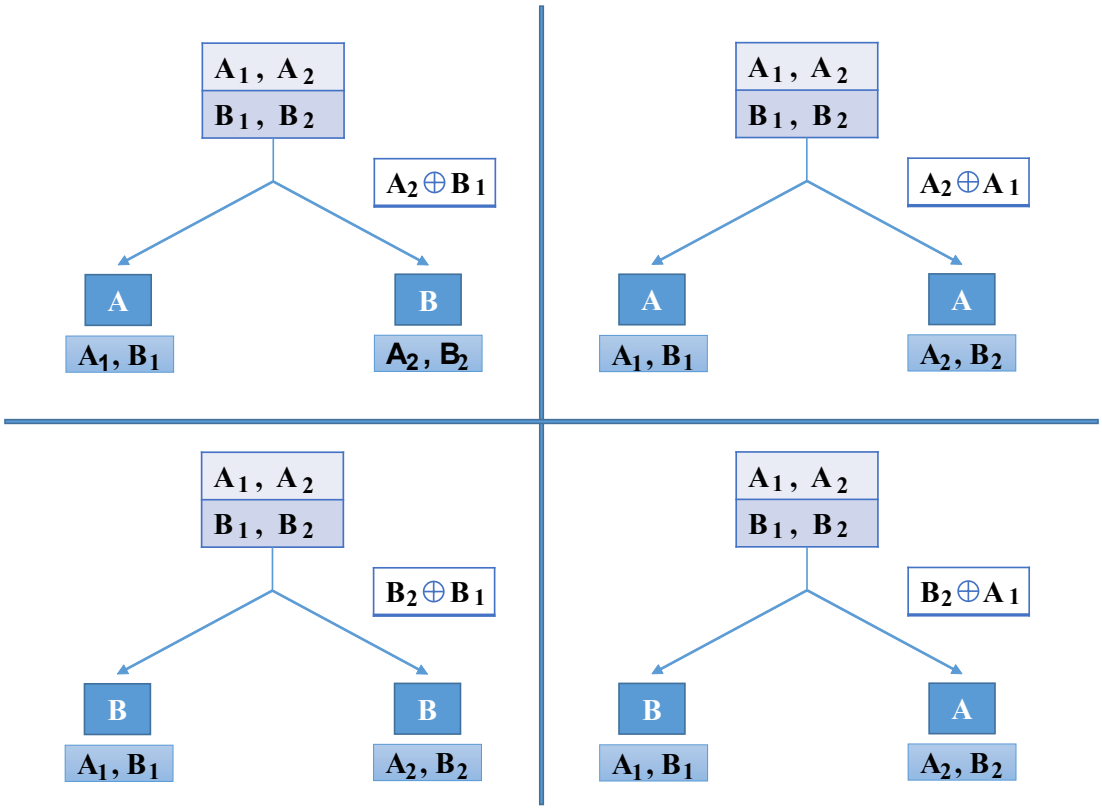

Figure 2.2: All four possible combinations of centralized coded caching configurations where $K = 2$ users with MF bit local cache memories and a central server containing $N = 2$ files [1].

it only needs A_2. User 2 has A_2, i.e., it only needs A_1. Therefore, the users can reconstruct their requested subfile when the server transmits $A_1 \oplus A_2$ whose size is $F/2$ bits.

Case 4: Both users request file B. User 1 already has B_1 in its cache, i.e., it only needs B_2. User 2 has B_2, i.e., it only needs B_1. Therefore, the users can reconstruct their requested subfile when the server transmits $B_1 \oplus B_2$ whose size is $F/2$ bits.

Thus, the centralized coded caching transmits only $F/2$ bits. In traditional uncoded caching, the server needs to transmit $(1 - M/N)$ portion of each file resulting in $R_U(M) \triangleq K \cdot (1 - M/N) \cdot \min\{1, N/K\} \cdot F = F$ bits of transmission. Hence, the centralized coded caching attains lower transmission rate than uncoded caching for all possible cases. □

In general, we can describe the coded caching algorithm as follows:

- During the placement phase, each file is split into $\binom{K}{t}$ non-overlapping equal size subfiles with $t = MK/N$. Let us denote the subfiles of W_n by $W_{n,\mathcal{S}}$ where $S \subset [K]$, $|\mathcal{S}| = t$.
- For each file in the server, subfile $W_{n,\mathcal{S}}$ is stored in the user k's cache if $k \in \mathcal{S}$. Thus, each user caches $N\binom{K-1}{t-1}\frac{F}{\binom{K}{t}} = FM$ bits in total.
- During the delivery phase, the server receives a request vector $(d_1, \cdots, d_K)$, i.e., user k wants file W_{d_k}.
- The server transmits $\oplus_{s \in \mathcal{S}} W_{d_s, \mathcal{S}\setminus\{s\}}$ for each subset $\mathcal{S} \subset [K]$ with $|S| = t+1$.

Accordingly, the achievable rate $R_C(M)$ of the centralized coded caching scheme is given in Theorem 1 of [1] as

$$R_C(M) \triangleq K \cdot (1 - M/N) \cdot \min\left(\frac{1}{1 + KM/N}, \frac{N}{K}\right). \tag{2.1}$$

The factor $(1 - M/N)$ in (2.1) is due to the local caching gain, and it is present in both uncoded caching and coded caching while the factor $\frac{1}{1+KM/N}$ is due to the global caching gain, and it is only provided by the coded caching scheme.

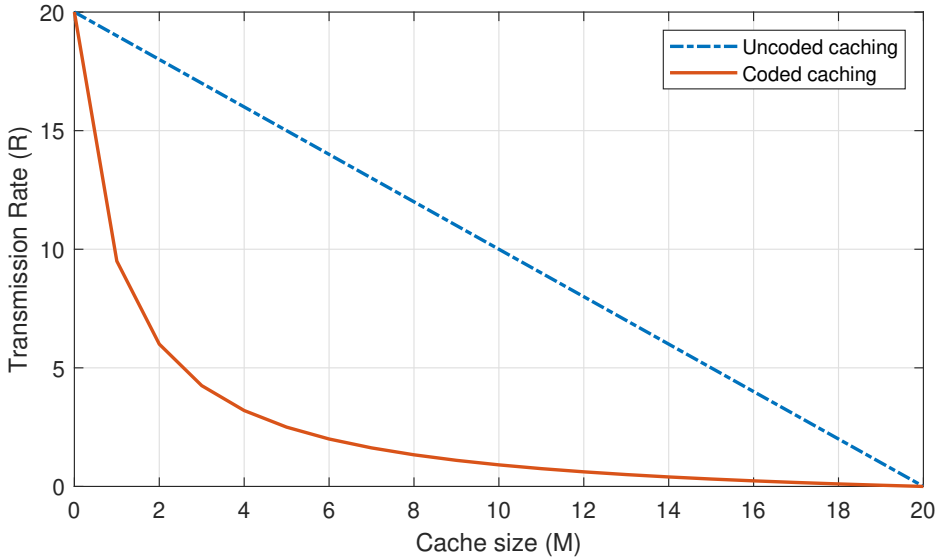

Figure 2.3: Transmission rate R required for traditional uncoded caching and coded caching with $N = K = 20$ with different cache sizes.

In Fig. 2.3, the transmission rate required for uncoded caching and coded caching with $N = K = 20$ is illustrated to emphasize the importance of global caching gain. For example, when the cache size is $M = 10$, the coded caching requires to transmit only $0.91 \cdot F$ bits while uncoded caching needs to transmit $10 \cdot F$ bits. Hence, coded caching achieves a 90.9% reduced transmission rate than uncoded caching.

2.1.2 Decentralized Coded Caching

In centralized coded caching, the placement phase is centrally coordinated, and both the number and identity of users are known to the server at the placement phase. However, this kind of coordination is not possible in real-life networks. Hence, in [2], Maddah-Ali and Nielsen propose a decentralized coded caching scheme that can provide a global caching gain even when there is no coordination.

Consider the same system with the centralized coded caching setup where K users each equipped with M caches are connected to a server containing N files

each of size F bits through an error-free shared link. Similar to the centralized coded caching, the system operates in two phases: placement and delivery phase. During the delivery phase, each user independently caches MF/N bits of each file chosen uniformly at random. Note that, unlike the centralized coded caching, the size of the cached contents for each file does not depend on the number of users K, instead it only depends on M and N. At the beginning of the delivery phase, the number and identity of the users are known to the server, and we can consider each file as a combination of 2^K exclusive subfiles. Let us use the notation $V_{k,\mathcal{S}}$ to denote the bits of the file requested by the k-th user stored by the users exclusively in $\mathcal{S}$. During the delivery phase, the server selects one of the following described procedures to minimize the transmission rate.

Algorithm 1: Delivery procedures for decentralized coded caching algorithm [2].

Procedure 1:
for $s = K, K-1, \cdots, 1$ **do**
  for $\mathcal{S} \subset [K] : |\mathcal{S}| = s$ **do**
    server trasnmits $\oplus_{k \in \mathcal{S}} V_{k,\mathcal{S}\setminus\{k\}}$
  end
end

Procedure 2:
for $n \in [N]$ **do**
  server transmits enough random linear cominations of bits of file n
  until each user can decode its requested file.
end

Procedure 1 can be explained with the following illustrative toy example:

Example 2: Consider a caching problem with $K = 2$ users each equipped with a cache of size $M = 1$, and there are $N = 2$ files in the server denoted by A and B. User one request file A, while the other one requests file B as illustrated in Fig. 2.4.

- During the placement phase, each user caches $MF/2 = F/2$ bits of each file randomly and independently.

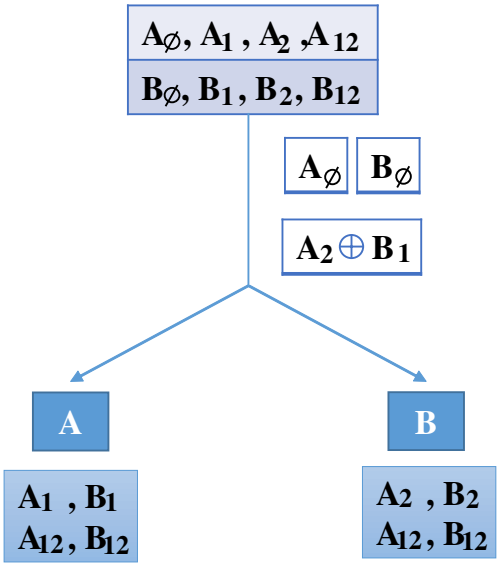

Figure 2.4: Transmission rate R required for traditional uncoded caching and coded caching with $N = K = 20$ with different cache sizes.

- At the beginning of the delivery phase, the server has access to the connected user identities and their requests.

- Each bit of a file is stored in a specific user's cache with probability $M/2 = 1/2$, and a specific bit of a file can be cached by none of the users, only by user 1, only by user 2, or by both of the users. Hence, we can consider a file as a combination of four exclusive subfiles, i.e., file A is partitioned into $A = (A_\emptyset, A_1, A_2, A_{1,2})$, and $A_\mathcal{S}$ represents the bits of file A stored in $\mathcal{S}$ where $\mathcal{S} \subset \{1, 2\}$. Using the law of large numbers, for large F, the size of each subfile can be approximately calculated as

$$|A_\mathcal{S}| \approx \left(\frac{M}{2}\right)^{|\mathcal{S}|} \left(1 - \frac{M}{2}\right)^{2-|\mathcal{S}|} F \tag{2.2}$$

  with probability one. Hence, we have $|A_\emptyset| \approx \left(1 - \frac{M}{2}\right)^2 F$, $|A_1| \approx \left(\frac{M}{2}\right)\left(1 - \frac{M}{2}\right) F$, $|A_2| \approx \left(\frac{M}{2}\right)\left(1 - \frac{M}{2}\right) F$ and $|A_{1,2}| \approx (\frac{M}{2})^2 F$.

- In Algorithm 1, when $s = 2$, we have $V_{1,2} = A_2$ and $V_{2,1} = B_1$. Thus, the server transmits $A_2 \oplus B_1$ whose size is $\approx \left(\frac{M}{2}\right)\left(1 - \frac{M}{2}\right) F$, and each user can decode their required subfile using the received signal and their cache contents.

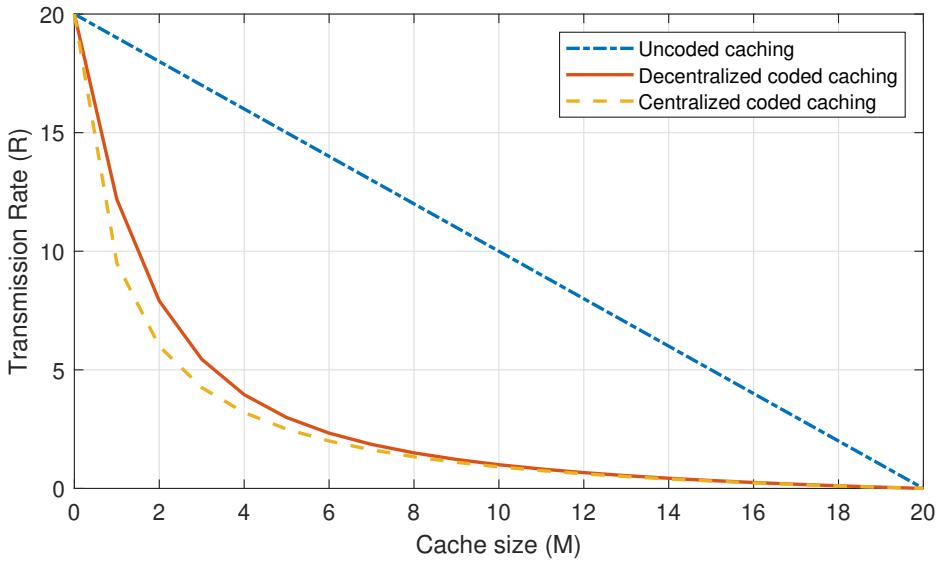

Figure 2.5: Transmission rate R required for traditional uncoded caching, decentralized coded caching, and centralized coded caching with $N = K = 20$, and different cache sizes.

- When $s = 1$, $V_{1,\emptyset} = A_\emptyset$ and $V_{2,\emptyset} = B_\emptyset$. Since none of the users have these subfiles in their local caches, no multicasting opportunities can be attained by coding. Hence, each of these subfiles are transmitted separately by the server resulting in $\approx 2\left(1 - \frac{M}{2}\right)^2 F$ bit of transmission.
- Note that $A_{1,2}$ and $B_{1,2}$ are cached by both of the users which means there is no need to transmit them.

Combining the results of the above cases, the overall transmission rate becomes $\approx \frac{3}{4}F$. □

Generalizing this illustrative example, the authors show that for N files and K users each with a cache size of M, for F large enough, Algorithm 1 gives a transmission rate arbitrarily close to

$$R_D(M) \triangleq K \cdot (1 - M/N) \cdot \min\left(\frac{N}{KM}\left(1 - (1 - M/N)^K\right), \frac{N}{K}\right). \tag{2.3}$$

As in the centralized coded caching, the term $(1 - M/N)$ is the local caching

gain, while $\frac{N}{KM}\left(1-(1-M/N)^K\right)$ is the result of global caching gain which is attained via the coded multicasting opportunities.

The performances of uncoded caching, centralized coded caching, and decentralized coded caching with $K = N = 20$ are illustrated in Fig. 2.5 which verifies the efficiency of decentralized coded caching. Uncoded caching only achieves a local caching gain, while the centralized and decentralized coded caching algorithms have a global caching gain along with the local one. Hence, both outperform the uncoded caching. However, in centralized coded caching, the server knows the identity of the users, which will be connected to the server during the delivery phase. Thus, the distribution of files is coordinated by the server, which results in a higher global caching gain. In decentralized coded caching, the server does not have any knowledge about users; hence the placement phase is performed in a random manner without coordination resulting in a slight decrease in the global caching gain.

2.1.3 Literature Review on Coded Caching

With its promised global caching gain, coded caching has drawn significant attention, and various extensions have been proposed over the last few years. In [3], the authors investigate the gap between the caching rate and the cut-set bound. They develop a coded caching strategy via network coding for both the delivery and placement phases, where the number of users is higher than the number of files in the server, and each user equipped with a small buffer. Their proposed strategy outperforms most of the existing coded caching schemes, and they show that the cut-set bound rate is achievable. In [4], a novel centralized coded caching scheme is proposed for the specific case of a cache capacity of $M = (N-1)/K$, and a lower transmission rate is achieved via the proposed scheme than the existing ones when $K \geq N \geq 3F$. In [5], the authors investigate the lower bound on the transmission rate of the centralized coded caching, and improve the bound introduced in [1, 2] for the average and the worst-case rate-memory trade-offs. Specifically, the authors compare their newly derived lower bound and the upper

bound given in [2], and show that the ratio of the upper bound to the new lower bound is decreased to 2.315 and 2.507 for the worst case and the average case, respectively.

Different from other studies, [6] considers a more flexible setup where each user decides which files to store arbitrarily. The delivery process is optimized by solving an integer linear program. The numerical results show that the proposed scheme achieves a lower bandwidth usage than the existing ones when the placement phase is uniformly random.

Ref. [7] focuses on asynchronous file requests where requests of each user arrive to the server at different times. Also, each user specifies a deadline to receive their requested files. They propose a linear programming formulation to determine the transmission schedule for asynchronous coded caching and propose a minimum cost network flow algorithm to reduce the complexity of the linear program. In [8], the authors study the trade-off between coded caching and delivery delay for delay-sensitive contents. They present three computationally efficient merging functions to combine the requests as much as possible, thereby minimizing the number of transmissions while considering the delivery-delay constraint. For large delay constraints, they can achieve the optimal performance given in [1, 2]. For strict delay constraints, the proposed approach does not achieve the optimal solution, however, it can still offer an important gain.

Ref. [9] introduces secure coded caching which uses random keys to protect users from an external eavesdropper. The goal of the paper is to minimize the information leakage to an unintended wiretapper. They obtain a memory-transmission rate trade-off for secure communication and show that security can be attained with a negligible cost. A related study, private coded caching, is performed in [10] where the authors aim at protecting the user requests and cache contents from all the other users in the system, i.e., no user can extract any information about the files it does not demand. They propose a feasible private coded caching scheme and prove the order-optimality of the proposed solution via information theoretical lower bounds.

Another interesting line of research is to study the case where the popularity distribution of the files in the server is not uniform, i.e., some files has a higher probability of being requested. For different popularity distributions, in [11], the authors optimally perform the placement phase by utilizing the distribution of the files in the server to minimize the load during the delivery phase. For a cache size equal to $M = 1$, the optimal placement algorithm is to store the most popular file in the cache. However, when $M > 1$, caching the most popular file is suboptimal. Hence, they propose a novel scheme by separating contents into groups according to their popularity distribution. During the placement phase, the same amount of cache is allocated for the files in the same group while the files in different groups may have a different amount of cache allocation. During the delivery phase, the authors only consider the coding opportunities among the same group and ignore the remaining ones. They show that their proposed solution is near optimal. In [12], the authors study online coded caching where the popularity of files in the server changes according to a Markov model during the delivery phase. They show that online coded caching achieves a very close performance to offline coded caching in terms of long-term average rates. In [13], hierarchical coded caching is investigated where the system consists of two layers of caches, and multicasting opportunities within each layer and across multiple layers are simultaneously created.

There have also been works on coded caching when the links between the users and the server during the delivery phase are non-ideal. In [14], a centralized joint encoding scheme has been proposed based on the coding scheme of [15] where the delivery phase is over a packet erasure channel. Receivers are divided into two groups as strong and weak, considering their erasure probabilities. Only weak receivers are equipped with local caches, and it is shown that even if strong receivers do not have any caches, they take advantage of the presence of weak receivers' local caches. Also, the theoretical trade-off between the achievable transmission rate and cache memory is analyzed. Reference [16] investigates decentralized coded caching over packet erasure broadcast channels by separating receivers as weak and strong with and without a secrecy constraint. The results show that communication can be secured against an external eavesdropper by a

slight increase in the transmission rate.

In [17], the authors aim to overcome the detrimental effects of weak users by designing opportunistic scheduling policies using a long-term average rate utility function. They also propose a threshold-based scheduling algorithm for asymmetric channel statistics to balance fairness among users. Both of these approaches focus on long-term averages and ignore the users whose channel gains are below a threshold, and hence, are not served. Ref. [18] exploits the pattern of coded messages by adjusting power and bandwidth allocation among submessages designated for a different subset of users to maximize the throughput, and applies both time division and frequency division modes during the delivery phase over fading channels. Ref. [19] considers a system with coded multicasting and channel coding over slow fading channels and study average delay and outage trade-off. In [20], the authors consider a coded caching system where the power allocation for the subfiles is designed according to the intended users, and they analyze the long-term average sum content delivery rate over fading channels. Furthermore, in [21, 22], the authors investigate coded caching over multiple input multiple output (MIMO) wireless networks where each user is equipped with a single antenna while the server is considered as a multi-antenna basestation. Ref. [23] applies interference management to alleviate the negative effect of link quality differences among users due to channel variations.

2.2 Coded Computing

With the rapid growth of the amount of data available, the accuracy and reliability of machine learning algorithms are enhanced, since training with more extensive training sets increases the accuracy of the learning algorithms [24]. In addition, the capability of computing devices has increased, which makes the processing of the dataset faster. However, the total amount of data is nearly incalculable, and increasing the computation speed of a single device is difficult due to the saturation of Moore's law [25]. Therefore, distributing the data to multiple devices/workers to perform parallel computing has become an inevitable

approach to speed up computations.

To reduce the computation time of linear transforms, which are the core operations performed in many machine learning algorithms, classical approaches consider the following setup: A fusion node distributes the computational task to all the connected computation nodes equally without adding redundancy. At the end of the computation process, the fusion node needs to wait for all these devices to complete and send the results of their computation [26, 27]. The basic idea is the following: consider the multiplication operation

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B}, \tag{2.4}$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are $M \times M$ full matrices. Instead of performing multiplication at one step, we decompose $\mathbf{A}$ and $\mathbf{B}$ into $\hat{\mathbf{A}}_{\mathbf{lk}} = \mathbf{A}_{\mathbf{ij}}$ and $\hat{\mathbf{B}}_{\mathbf{lk}} = \mathbf{B}_{\mathbf{ij}}$ with $\frac{1}{4}Ml \leq i \leq \frac{1}{4}M(l+1)$ and $\frac{1}{4}Mk \leq j \leq \frac{1}{4}M(k+1)$, as shown in Table 2.1.

Table 2.1: Subblock decomposition of square matrices.

$\hat{\mathbf{A}}_{\mathbf{00}}$	$\hat{\mathbf{A}}_{\mathbf{01}}$	$\hat{\mathbf{A}}_{\mathbf{02}}$	$\hat{\mathbf{A}}_{\mathbf{03}}$
$\hat{\mathbf{A}}_{\mathbf{10}}$	$\hat{\mathbf{A}}_{\mathbf{11}}$	$\hat{\mathbf{A}}_{\mathbf{12}}$	$\hat{\mathbf{A}}_{\mathbf{13}}$
$\hat{\mathbf{A}}_{\mathbf{20}}$	$\hat{\mathbf{A}}_{\mathbf{21}}$	$\hat{\mathbf{A}}_{\mathbf{22}}$	$\hat{\mathbf{A}}_{\mathbf{23}}$
$\hat{\mathbf{A}}_{\mathbf{30}}$	$\hat{\mathbf{A}}_{\mathbf{31}}$	$\hat{\mathbf{A}}_{\mathbf{32}}$	$\hat{\mathbf{A}}_{\mathbf{33}}$

$\hat{\mathbf{B}}_{\mathbf{00}}$	$\hat{\mathbf{B}}_{\mathbf{01}}$	$\hat{\mathbf{B}}_{\mathbf{02}}$	$\hat{\mathbf{B}}_{\mathbf{03}}$
$\hat{\mathbf{B}}_{\mathbf{10}}$	$\hat{\mathbf{B}}_{\mathbf{11}}$	$\hat{\mathbf{B}}_{\mathbf{12}}$	$\hat{\mathbf{B}}_{\mathbf{13}}$
$\hat{\mathbf{B}}_{\mathbf{20}}$	$\hat{\mathbf{B}}_{\mathbf{21}}$	$\hat{\mathbf{B}}_{\mathbf{22}}$	$\hat{\mathbf{B}}_{\mathbf{23}}$
$\hat{\mathbf{B}}_{\mathbf{30}}$	$\hat{\mathbf{B}}_{\mathbf{31}}$	$\hat{\mathbf{B}}_{\mathbf{32}}$	$\hat{\mathbf{B}}_{\mathbf{33}}$

By considering each submatrix as a single element, we can write

$$\hat{\mathbf{C}}_{\mathbf{lk}} = \sum_{n} \hat{\mathbf{A}}_{\mathbf{ln}} \cdot \hat{\mathbf{B}}_{\mathbf{nk}}, \tag{2.5}$$

and calculate the result of (2.4) by executing the following steps:

1. Diagonal submatrices of $\mathbf{A}$ are broadcast to all the processors in a horizontal direction, i.e., processor i receives $\hat{\mathbf{A}}_{\mathbf{ii}}$.

2. Each processor $i \in [M]$ performs (2.5) with $\hat{\mathbf{A}}_{\mathbf{ii}}$ and $\mathbf{B}$ in their hand resulting in the matrix shown in Table 2.2.

3. Submatrices of **B** are vertically rolled. The result of first roll is shown in Table 2.3.

4. Horizontal broadcasting for 'diagonal+1' submatrices of **A** is performed, e.g., the submatrices shown in Table 2.4 are broadcast.

5. Each processor multiplies the currently transmitted submatrices (dioganal+1) of **A** and rolled **B** to perform (2.5).

This steps are repeated until **B** rolled completely.

Table 2.2: Result of Step 2 in matrix multiplication.

$\hat{\mathbf{A}}_{00}\hat{\mathbf{B}}_{00}$	$\hat{\mathbf{A}}_{00}\hat{\mathbf{B}}_{01}$	$\hat{\mathbf{A}}_{00}\hat{\mathbf{B}}_{02}$	$\hat{\mathbf{A}}_{00}\hat{\mathbf{B}}_{03}$
$\hat{\mathbf{A}}_{11}\hat{\mathbf{B}}_{10}$	$\hat{\mathbf{A}}_{11}\hat{\mathbf{B}}_{11}$	$\hat{\mathbf{A}}_{11}\hat{\mathbf{B}}_{12}$	$\hat{\mathbf{A}}_{11}\hat{\mathbf{B}}_{13}$
$\hat{\mathbf{A}}_{22}\hat{\mathbf{B}}_{20}$	$\hat{\mathbf{A}}_{22}\hat{\mathbf{B}}_{21}$	$\hat{\mathbf{A}}_{22}\hat{\mathbf{B}}_{22}$	$\hat{\mathbf{A}}_{22}\hat{\mathbf{B}}_{23}$
$\hat{\mathbf{A}}_{33}\hat{\mathbf{B}}_{30}$	$\hat{\mathbf{A}}_{33}\hat{\mathbf{B}}_{31}$	$\hat{\mathbf{A}}_{33}\hat{\mathbf{B}}_{32}$	$\hat{\mathbf{A}}_{33}\hat{\mathbf{B}}_{33}$

Table 2.3: Vertical rolling of **B**.

$\hat{\mathbf{B}}_{00}$	$\hat{\mathbf{B}}_{01}$	$\hat{\mathbf{B}}_{02}$	$\hat{\mathbf{B}}_{03}$		$\hat{\mathbf{B}}_{10}$	$\hat{\mathbf{B}}_{11}$	$\hat{\mathbf{B}}_{12}$	$\hat{\mathbf{B}}_{13}$
$\hat{\mathbf{B}}_{10}$	$\hat{\mathbf{B}}_{11}$	$\hat{\mathbf{B}}_{12}$	$\hat{\mathbf{B}}_{13}$		$\hat{\mathbf{B}}_{20}$	$\hat{\mathbf{B}}_{21}$	$\hat{\mathbf{B}}_{22}$	$\hat{\mathbf{B}}_{23}$
$\hat{\mathbf{B}}_{20}$	$\hat{\mathbf{B}}_{21}$	$\hat{\mathbf{B}}_{22}$	$\hat{\mathbf{B}}_{23}$	$\Rightarrow$	$\hat{\mathbf{B}}_{00}$	$\hat{\mathbf{B}}_{01}$	$\hat{\mathbf{B}}_{02}$	$\hat{\mathbf{B}}_{03}$
$\hat{\mathbf{B}}_{30}$	$\hat{\mathbf{B}}_{31}$	$\hat{\mathbf{B}}_{32}$	$\hat{\mathbf{B}}_{33}$		$\hat{\mathbf{B}}_{00}$	$\hat{\mathbf{B}}_{01}$	$\hat{\mathbf{B}}_{02}$	$\hat{\mathbf{B}}_{03}$

Table 2.4: Horizontal broadcasting for 'diagonal+1' subbmatrices of $\mathbf{A}$.

	$\hat{\mathbf{A}}_{\mathbf{01}}$		
		$\hat{\mathbf{A}}_{\mathbf{12}}$	
			$\hat{\mathbf{A}}_{\mathbf{23}}$
$\hat{\mathbf{A}}_{\mathbf{30}}$			

Note that, the algorithm does not perform any redundant operation, and waits until all the computation devices complete their operations to obtain $\mathbf{C}$.

In most parallel computing systems, some of the computation devices, called stragglers, are slower than others and cause delays in computation. In [28], the authors categorize the reasons for outliers/stragglers into three classes as machine characteristics, network characteristics, and imbalance in work-partitioning. They present an approach called as *Mantri* which classifies outliers according to their causes and prevent slowdown of the system by the following procedures: 1) they restart the task of outliers to get rid of work imbalance, 2) the work-sharing is done according to the network characteristics, 3) the result of the task is protected by replicating the task according to the proposed cost-benefit analysis while preventing excessive task replication. Ref. [29] considers a heterogeneous system where some of the computation devices are stragglers. They take advantage of the estimated completion time of the works to obtain a robust scheduling algorithm in order to distribute the work based on finish times.

Another approach to eliminate the slowdown effects of stragglers is to introduce redundancy into the computation task. In [30], a fault-tolerant encoding algorithm for multiprocessor systems is introduced with low redundancy. In [31], the authors perform a theoretical analysis of the trade-off between response time

and resource usage in parallel computing systems. With the awareness of variability of the task execution time of each machine, they investigate replication and scheduling policies that are optimal and nearly optimal, and they analyze the conditions where and when the task replication is beneficial for the distributed systems. Furthermore, in [32], they expand their task replication analysis for multiple tasks by investigating the effects of execution time distribution of machines on the trade-off between cost and execution time, and propose new replication strategies for multiple tasks.

While the previously mentioned works focus on latency and source usage in distributed computation, Ref. [33] uses Maximum Distance Separable (MDS) codes to speed up the computation in distributed systems where some of the connected servers are stragglers. The authors analyze the trade-off between computation time and communication (shuffling) load. For a predetermined computation time, they prove a lower bound on the communication load for matrix multiplication through an information theoretic analysis.

In [34], the authors prove the superiority of coded distributed computation over uncoded ones. For matrix multiplication, they use MDS codes to reduce the destructive effects of stragglers, and prove that completion time of distributed matrix multiplication can be reduced by a factor of $\log n$ where n is the number of homogeneous workers. For data shuffling, they aim to reduce the load of communication. The authors show that coded shuffling reduces the communication load by a factor of $\left(\alpha + \frac{1}{n}\right) \delta(n)$ with respect to uncoded shuffling where n is the number of workers, α is the fraction of the matrix stored in each worker, and $\delta(n)$ is the ratio of cost of unicasting messages to n users to multicasting to n users.

A related topic evolving from coded computing is distributed machine learning where the computation load of the main server which performs the calculations during the training process are divided among edge devices as in coded computing [35, 36]. Different aspects of distributed ML are studied in the recent literature. In [37], digital and analog distributed stochastic gradient descent (D-DSGD and A-DSGD) algorithms over a Gaussian multiple-access channel (MAC) are proposed where the authors use the superposition property of the MAC to recover

the mean of local gradients computed at remote workers. In D-DSGD, workers digitally compress their locally computed gradients into a finite number bits while in A-DSGD workers use an analog compression similar to what is done in compressed sensing to obey the bandwidth limitations over wireless channels. In [38], for low latency distributed learning systems, the authors propose broadband analog aggregation scheme for a random network model with randomly distributed workers over a disk where the global model is updated at the central server using the average of locally computed models by focusing on power control and worker scheduling according to their channel state information (CSI). Ref. [39] models the channel between the workers and the parameter server (central server) as a band-limited fading MAC, and proposes analog compression schemes using both opportunistic scheduling and compressed sensing based on CSI to reduce the dimensionality of the gradient estimates. Also, they propose a worker scheduling scheme to align the received gradients based on beamforming.

In addition to the imperfections caused by fading, in [40], each worker transmits its gradient in a quantized form to effectively reduce the data exchange rate. The authors study the trade-off between the learning accuracy and precision of the transmitted gradients, and show that the convergence of the proposed approach is guaranteed. Ref [41] proposes the Quantized Epoch-SGD (QESGD) method, which compresses the updated model parameter at the parameter server by quantization, and sends the quantized version to the workers to reduce the communication load of the distributed learning system. Through numerical simulations of deep learning algorithms, the authors show that the proposed approach outperforms the other state of the art methods. In [42], the communication cost of federated learning is studied. The authors propose two approaches to reduce the uplink communication cost for poor network connections: 1) by restricting the parameter space using a structured update, 2) by compressing the local model after learning with a full model, and sending the compressed ones to the server. In [43], the secure aggregation method for federated learning systems is considered where the model is learned only by the server, and the data of participants are protected. The authors introduce two protocols, one is secure against honest adversaries with a lower communication cost while the other is against active

adversaries and comes with an extra communication load.

2.3 Thesis Contributions

In this thesis, firstly, we consider a coded caching system where the placement phase is performed in a decentralized manner, and the delivery phase takes place over a packet erasure channel where each receiver sees an independent channel with the same erasure probability. Although [14] and [16] give theoretical limits of coded caching over packet erasure channels; our proposed algorithms are practical and feasible. Firstly, we present a coding scheme called sending the same message algorithm (SSM) based on [2], and perform analytical calculations on the average transmission time for the worst-case scenario. Secondly, a greedy coded caching algorithm is proposed, and through simulation results, it is shown that it outperforms the proposed SSM algorithm. Finally, we introduce a grouped greedy coded caching algorithm which has a lower complexity than the greedy algorithm with a slight increase in the transmission rate. We also develop an upper bound on the transmission rate for the grouped greedy coded caching scheme which is tight for small erasure probabilities.

As a second contribution, we follow a coded caching model where the placement phase is performed in a decentralized manner and the delivery phase takes place over a wireless fading channel. Different from [17], which considers long-term average rates, our interest is to study non-ergodic channels and minimize the transmission time by letting some of the weak users to be in outage. With a fixed outage probability, we formulate an optimization problem to reduce the total transmission time by grouping the participating users to overcome the detrimental effects of channel fading. We also propose a locally optimal iterative algorithm to compute the signal to noise ratio (SNR) thresholds. Furthermore, we quantize the SNR thresholds and model the optimization process with the quantized thresholds as a shortest path problem for a reduced complexity solution.

Finally, we study distributed learning algorithms over wireless channels in realistic settings, also considering practical implementation issues, including the channel effects. We model the communication link as a frequency selective fading channel, and transmit the local gradients using OFDM. Furthermore, in an effort to reduce the hardware complexity and power consumption, we employ low-resolution ADCs at the receiver side, which employs multiple (even a massive number of) receive antennas. While decreasing the resolution of ADCs reduces the implementation cost and the power consumption, it also deteriorates the performance of a communication system. Our objective is to study and quantify the performance of a distributed learning system at the wireless edge implemented through OFDM based transmissions and low cost ADCs at the receiver side.

Chapter 3

Coded Caching over Packet Erasure Channels

In this chapter, we study coded caching over packet erasure channels, and propose practical and feasible algorithms to reduce the overall transmission rates. Firstly, we study sending the same message (SSM) algorithm, which simply retransmits the erased coded messages until all of the users successfully receive them. We provide analytical calculations for the average transmission rate considering distinct user demands. Secondly, we propose a greedy coded caching algorithm which gives a lower transmission rate than the SSM by exploiting the multicasting opportunities among all the erased subfiles in a greedy manner. Furthermore, we propose a grouped greedy algorithm which only considers multicasting opportunities within a range of erased messages; thus the complexity of the grouped greedy algorithm is less than that of the greedy one with a slight sacrifice in performance. Also, an upper bound of the overall transmission rate of the grouped greedy coded caching algorithm is developed, which is tight for small erasure probabilities.

The chapter is organized as follows. Section 3.1 introduces the system model. The SSM algorithm is introduced in Section 3.2, and a greedy coded caching algorithm is proposed in Section 3.3. A lower complexity solution called the grouped

greedy approach is presented in Section 3.4. Performance of the proposed algorithms are studied via simulations in Section 3.5, and the chapter is summarized in Section 3.6.

3.1 System Model

We consider a system which contains a server and K users which are connected through a packet erasure channel as shown in Fig. 3.1. There are N different files in the server where $K \leq N$ and $\boldsymbol{W} \triangleq (W_1, W_2, \cdots, W_N)$ represents the files each of size F bits in the server. Users are equipped with local caches which are able to store MF bits. During the placement phase, each user randomly caches M/N fraction of each file in their local caches in a decentralized manner as described in [2] and summarized in Chapter 2. We use the notation $W_{i,\{S\}}$ to represent the bits of the file W_i which are present in the cache of every user in S exclusively. After the decentralized placement phase, each file can be split into 2^K subfiles as $W_i = (W_{i,\{\emptyset\}}, W_{i,\{1\}}, W_{i,\{2\}}, \cdots, W_{i,\{K\}}, W_{i,\{1,2\}}, W_{i,\{1,3\}}, \cdots, W_{i,\{1,2,\cdots,K\}})$. We use d_k to denote the demand of the k-th user where $d_k \in [N]$, $\forall k \in [K]$. The aim of the server is to satisfy the demands of all the users.

Coded caching takes place in two steps. During off-peak hours, the content of the server is distributed over local user caches randomly without considering the user demands over an error-free shared link. This phase is called the placement phase. At the end of this phase, each user determines its cache content using the placement function g_k, where cache content of the k-th user is denoted by Z_k with $k \in [K]$ and defined as

$$Z_k \triangleq g_k(W_1, W_2, \cdots, W_N). \tag{3.1}$$

The second phase is the delivery phase in which the user requests are revealed to the server. In this phase, the server encodes library contents with encoding function $f_{\boldsymbol{d}}$ using the users' request vector $\boldsymbol{d} \triangleq (d_1, d_2, \cdots, d_K)$ and library

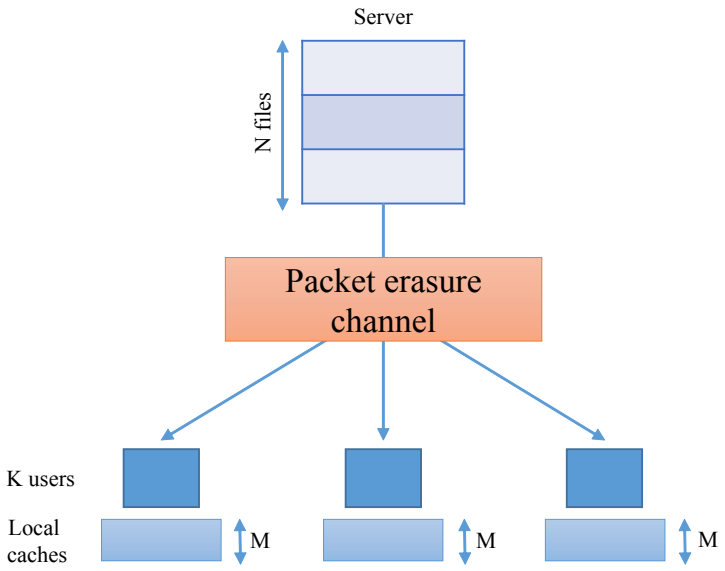

Figure 3.1: Packet erasure channel with K users with MF bit local cache memories and a central server with N content.

content to obtain a length-n codeword X^n as

$$X^n \triangleq f_{\boldsymbol{d}}(W_{d_1}, W_{d_2}, \cdots, W_{d_K}). \tag{3.2}$$

Similar to [1] and [3], the channel between the users and server is modeled as packet erasure channel during the delivery phase. The input alphabet of the packet erasure channel is $\mathcal{X} \triangleq \{0,1\}^F$ while the output alphabet is $\mathcal{Y} \triangleq \mathcal{X} \cup \Delta$ where F is the packet size and Δ represents the packet erasure symbol. Each user encounters independent packet erasures over a channel with erasure probability ϵ.

Receiver $k \in [K]$ uses the decoding function φ_k to reconstruct its demanded content W_{d_k} based on its observation Y_k^n, cache content Z_k, and demand vector $\boldsymbol{d}$ as $\hat{W}_{d_k} \triangleq \varphi_k(Y_k^n, Z_k, \boldsymbol{d})$.

3.2 Sending The Same Message Algorithm

This algorithm employs the decentralized coded caching technique introduced in [2] as the baseline algorithm. The placement phase, and the first transmission in the delivery phase are exactly the same as the baseline system. After the first transmission, when at least one of the users encounters a packet erasure, this algorithm simply retransmits the same coded messages until all of the users are able to decode their own messages successfully.

Example 1: Consider a coded caching system where $K = 3$ users with cache size $M = 1$ are connected to a server which contains $N = 3$ files denoted by (W_1, W_2, W_3) via a packet erasure channel. The demand vector of users is $\boldsymbol{d} = (W_1, W_2, W_3)$. Focusing on the coded message $W_{1,\{2,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,2\}}$, the server needs to transmit this same message even if only one of the users encounters erasure, i.e., server will retransmit $W_{1,\{2,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,2\}}$.

The average transmission rate of the SSM algorithm is analyzed in the next theorem assuming the worst-case scenario where each user has a distinct demand.

Theorem 1 *Consider the coded caching problem over a packet erasure channel with N files each of size F bits in the server and K users equipped with a cache of MF bits with $K \leq N$, and $M \in [N]$. Each user sees an independent packet erasure channel with the same erasure probability ϵ. The average transmission rate of the SSM algorithm is arbitrarily close to*

$$R_{SSM}(M, \epsilon) = \sum_{i=1}^{K} \binom{K}{i} L_i X_i, \tag{3.3}$$

where

$$X_i = \frac{1 + \sum_{k=1}^{i-1} p_{k,i} X_k}{1 - p_i}, \text{ for } i = 2, \cdots, K,$$

$$L_i = \left(\frac{M}{N}\right)^i \cdot \left(1 - \frac{M}{N}\right)^{K-i}, \text{ for } i = 0, \cdots, K,$$

$$p_{k,n} = \binom{n}{k} \cdot \epsilon^k \cdot (1-\epsilon)^{n-k}, \text{ for } k \leq n,$$

$$p_k = p_{k,k}.$$

Proof *Let R'_j denote the expected value of transmission rate to send j XOR-ed subfiles until it is received by all of the targeted users where the number of targeted users is j. Then X_j is the expected value of normalized transmission rate i.e., $X_j = R'_j/L_j$ which is normalized by the subfile size where Lj and X_j are given in* (3.4), (3.4) *respectively.*

The total transmission rate for the SSM algorithm is analyzed iteratively, as in the following.

Transmission rate for the messages with one targeted user:
Consider a subfile which is not stored by any of the users, and assume that only one user requests this subfile. Therefore, it can be recovered only when it is sent separately. The algorithm starts with sending this message without coding, which gives L_1 in (3.4)*, where L_1 is the size of the subfile. Then if this message is successfully transmitted (no erasure) which occurs with probability $p_{0,1}$, we do not need to transmit anything ($0 \cdot p_{0,1}$ in* (3.4)*) where $p_{k,n}$ is defined in* (3.4)*. When there is an erasure which occurs with probability $p_{1,1}$, this event will be equal to the beginning of the transmission, this leads $p_{1,1} \cdot X_1$ in* (3.4)*.*

$$\begin{aligned} R'_1 &= L_1 \cdot \left(1 + p_{0,1} \cdot 0 + p_{1,1} \cdot X_1\right) \\ &= L_1 \cdot X_1, \end{aligned} \tag{3.4}$$

Here, $p_{1,1}$ means that targeted user encounters erasure. When such an event occurs, the remaining expected transmission rate is equal to the one at the beginning of the transmission (when no message is sent), i.e., the right-hand side of (3.4) *is equal to $L_1 \cdot X_1$.*
Hence, X_1 is obtained as:

$$X_1 = \frac{1}{1 - p_{1,1}}. \tag{3.5}$$

Since there are $\binom{K}{1}$ such messages, the total transmission rate for one targeted user can be calculated as:

$$R_1 = \binom{K}{1} L_1 X_1. \tag{3.6}$$

Transmission rate for the messages with two targeted users, e.g. $W_{1,\{2\}} \oplus W_{2,\{1\}}$:

Without loss of generality, assume user one requests subfile $W_{1,\{2\}}$ and has subfile $W_{2,\{1\}}$ in its own cache while user two requests subfile $W_{2,\{1\}}$ and has subfile $W_{1,\{2\}}$. Hence, these subfiles can be recovered when they are XOR-ed according to the baseline algorithm.

At the beginning of the transmission, we need to send the XOR-ed message ($L_2 \cdot 1$ in (3.7)). Then, if this message is successfully transmitted (no erasure) which occurs with probability $p_{0,2}$, we do not need to transmit anything ($L_2 \cdot p_{0,2} \cdot 0$ in (3.7)). When there is an erasure for one of the users, which is with probability $p_{1,2}$, this will be the same as the previous case (when one user is targeted) and leads to $p_{1,2} \cdot X_1$ in (3.7). Hence, while both of the users encounter erasure, it is the beginning scenario of the transmission, which results in $L_2 \cdot X_2$ in (3.7).

$$\begin{aligned} R_2' &= L_2 \cdot \left(1 + p_{0,2} \cdot 0 + p_{1,2} \cdot X_1 + p_{2,2} \cdot X_2\right) \\ &= L_2 \cdot X_2. \end{aligned} \tag{3.7}$$

Hence, X_2 is obtained as:

$$X_2 = \frac{1 + p_{1,2} X_1}{1 - p_{2,2}}. \tag{3.8}$$

There are $\binom{K}{2}$ such messages when 2 users are targeted, hence the total transmission rate can be calculated as:

$$R_2 = \binom{K}{2} L_2 X_2. \tag{3.9}$$

Similar calculations can be performed for all possible number of targeted users. By induction, we can obtain general formulas for X_i and R_i as:

$$X_i = \frac{1 + \sum_{k=1}^{i-1} p_{k,i} X_k}{1 - p_i}, \tag{3.10}$$

where $i = 2, \cdots, K$, and

$$R_i = \binom{K}{i} L_i X_i. \tag{3.11}$$

Note that, $p_{k,k} = p_k$, and $X_1 = \dfrac{1}{1 - p_1}$.

Then, the total transmission rate is found as

$$R_{SSM}(M, \epsilon) = \sum_{i=1}^{K} R_i = \sum_{i=1}^{K} \binom{K}{i} L_i X_i, \tag{3.12}$$

where $\binom{K}{i}$ *is the number of messages with* i *subfiles.*

3.3 Greedy Coded Caching Algorithm

The SSM algorithm simply retransmits the same coded messages over the packet erasure channel when at least one of the users encounters erasure. However, when there is at least one successful transmission of the coded message, new multicasting opportunities may be available among the erased packets, which would potentially result in a lower transmission rate than sending the same message algorithm again over the channel without considering the successfully received ones. Here, we propose a greedy coded caching algorithm that aims to send a multicast stream benefiting the maximum number of users at each transmission. This algorithm is utilized recursively until all of the users decode their required contents.

The first step of the transmission is to construct the usual coded caching messages for the desired contents. After the first transmission, coded messages are constructed using brute force search to find decodable coded messages in order to benefit the maximum number of users, e.g., if there are $K = 8$ users, the greedy algorithm's initial purpose is to reconstruct coded messages from 8 subfiles whose targeted users are distinct. A coded message X is decodable if and only if all the users can extract their requested subfile from X along with the cache contents. For instance, assume that the requests of user 1 and 2 are W_1 and W_2 while their caches contain $\{W_{2,\{1,3\}}\}$ and $\{W_{1,\{2\}}\}$, respectively. Since both users can reconstruct their desired subfile from the message $W_{1,\{2\}} \oplus W_{2,\{1,3\}}$ using both the message and their local cache content, the message $W_{1,\{2\}} \oplus W_{2,\{1,3\}}$ is a decodable message. Note that, if the number of bits of XOR-ed messages is not the same, the smaller one is zero-padded.

The above greedy algorithm has $\mathcal{O}(\epsilon^K \cdot 2^{K^2})$ average-case complexity at each iteration which prevents real-time processing. Hence, we offer a grouped greedy algorithm, which has a lower complexity in the next section.

3.4 Grouped Greedy Coded Caching

Since the computational cost of the greedy algorithm is high, we propose another approach which constructs similar coded messages to [2], but at the same time tries to take advantage of new multicasting opportunities in a greedy manner when there is an erasure.

The following definitions are used to describe the proposed approach.

Definition 1 *Companion subfiles are the subfiles which construct the same coded message according to the decentralized coded caching algorithm.*

Definition 2 *Successive subfile of $W_{i,\mathcal{S}}$ is the subfile $W_{i,\mathcal{U}}$ where $\mathcal{U}$ is the nonempty proper (or strict) subset of $\mathcal{S}$.*

In the following, we present an example of companion subfile and successive subfiles.

Example 2: Consider the same scenario with Example 1 in Section 3.2. For the coded message $W_{1,\{2,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,2\}}$, subfiles $W_{1,\{2,3\}}$, $W_{2,\{1,3\}}$, and $W_{3,\{1,2\}}$ are companions of each other, since they construct a single coded message. Focusing on the subfile $W_{1,\{2,3\}}$, $\mathcal{S} = \{2,3\}$. Then $W_{1,\{2\}}$ and $W_{1,\{3\}}$ are the successive subfiles of $W_{1,\{2,3\}}$. $\square$

Similar to the greedy algorithm, grouped greedy coded caching initially transmits the same set of coded messages with [2]. After the first transmission of all the messages, each erased subfile's companion and successive subfiles are checked to determine whether new multicasting opportunities can be attained. In the SSM algorithm, these new multicast coded messages are disregarded, and multiple coded messages retransmitted even if it is not necessary. Therefore this method is expected to achieve a lower transmission rate then SSM. In the greedy coded caching, while we greedily explore new multicasting opportunities, its complexity is high, which makes the greedy approach undesirable. However, grouped

Table 3.1: Outputs of the First Message

Targeted users	**Coded message:** $W_{1,\{2,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,2\}}$
User 1	Δ
User 2	$W_{1,\{2,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,2\}}$
User 3	$W_{1,\{2,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,2\}}$

greedy coded caching requires to check only companion and successive subfiles. Accordingly, our grouped greedy coded caching algorithm attains a lower transmission rate than the SSM algorithm, and it has a lower complexity than the greedy coded caching approach.

In the following, we present an illustrative example to highlight the critical points of the proposed algorithm.

Example 3: Consider the same scenario with Example 1 in Section 3.2. In Table 3.1, a sample output over packet erasure channel for each targeted user is given for the first coded message $W_{1,\{2,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,2\}}$, while in Table 3.2 it is given for the second coded message $W_{1,\{2\}} \oplus W_{2,\{1\}}$. Note that, coded messages which encounter erasure are represented by Δ.

The subfile $W_{1,\{2,3\}}$ of the first message is transmitted for user 1, and it is received as Δ by the first user while other users successfully receive companion subfiles in the first message. Before constructing a new message, the server needs to check successive subfiles of $W_{1,\{2,3\}}$ which are $W_{1,\{2\}}$ and $W_{1,\{3\}}$. In the second coded message, $W_{1,\{2\}}$ is targeted for user 1 and received successfully by user 1, while user 2's input is Δ ($W_{1,\{2,3\}}$ is necessary for user 2, but it encounters erasure). Grouped greedy coded caching uses $W_{1,\{2,3\}}$ and its successive subfile $W_{2,\{1\}}$ and sends a single coded message $W_{1,\{2,3\}} \oplus W_{2,\{1\}}$. Since the shorter subfile is zero-padded, the required transmission rate will be the size of the subfile with the maximum length. On the other hand, the SSM algorithm retransmits first and second coded messages separately in such a scenario, whose transmission rate is the summation of the sizes of two messages. $\square$

Table 3.2: Outputs of the Second Message

Targeted users	**Coded message:** $W_{1,\{2\}} \oplus W_{2,\{1\}}$
User 1	$W_{1,\{2\}} \oplus W_{2,\{1\}}$
User 2	Δ

An upper bound for the average transmission rate of grouped greedy coded caching algorithm is given in the next theorem.

Theorem 2 *Consider the coded caching problem over a packet erasure channel with N files each of size F bits in the library and K users equipped with a cache of MF bits with $K \leq N$, and $M \in [N]$. Assume that each user sees an independent packet erasure channel with the same erasure probability ϵ. An upper bound for the average transmission rate of grouped greedy coded caching algorithm is*

$$R_{greedy}(M,\epsilon) \leq R_{SSM}(M,\epsilon) - R_{gain}(M,\epsilon), \tag{3.13}$$

where $R_{SSM}(M,\epsilon)$ is given in (3.3), and $R_{gain}(M,\epsilon)$, $E(k,j,m)$, and $A(K,j,m)$ are given by

$$R_{gain}(M,\epsilon) = \sum_{j=2}^{K} \sum_{m=1}^{j-1} \sum_{k=1}^{j-m-1} E(k,j,m) \cdot \Big(X_j L_j + X_{j-1} L_{j-1} - A(k,j,m) \Big), \tag{3.14}$$

$$E(k,j,m) = \binom{K}{j} \cdot \binom{j}{m} \cdot \binom{j-m-1}{k} \cdot (j-m) \cdot q_{k+m,2j-1}, \tag{3.15}$$

$$A(k,j,m) = \frac{\max\{L_{j-1}, L_j\} \Big(1 + \sum_{u=1}^{k+m-1} q_{u,k+m} \sum_{t=0}^{m} \binom{m}{m-t} \cdot \binom{k}{u-m+t} \Big)}{1 - q_{k+m,k+m}}, \tag{3.16}$$

with L_j and X_j being defined in the previous section.

Proof *Let us assume that there are j subfiles in the first coded message, and m of these subfiles are erasured after the first transmission. There are k erasures in the second coded message, which contains successive subfiles for the erasured subfiles, and these subfiles are successfully transmitted, i.e., $j-k-1$ successful transmission in the second message.*

There will be $\binom{K}{j}$ *such messages in the first transmission, and* m *erasures may occur in* $\binom{j}{m}$ *different ways. There will be* $j-m$ *messages in the second transmission which can be coded with the erased subfiles of the first message, and in the second message, erasures can occur in* $\binom{j-m-1}{k}$ *ways.*

The probability of having such an erasure pattern is $q_{k+m,2j-1} = \epsilon^{k+m} \cdot (1-\epsilon)^{2j-k-m-1}$ *where* $q_{k,n} = \epsilon^k \cdot (1-\epsilon)^{n-k}$ *for* $k < n$*. Then expected value of this scenario is represented by* $E(k,j,m)$ *as analyzed in (3.15).*

Let us define the new transmission rate as $A(k,j,m)$*, where the first message has* j *subfiles, and* m *of the subfiles are erased. There are* k *erasures in the second message which has* $j-1$ *subfiles in total. When these two successive groups construct a new coded message, this new message will have* $k+m$ *subfiles. Hence,* $k+m$ *users are targeted by the new message. Thus, we have* $A(k,j,m)$ *for* $1 \le k \le j-m-1$*,* $1 \le m \le j-1$*, and* $2 \le j \le K$ *as shown in* (3.17)*.*

$$
\begin{aligned}
A(k,j,m) = \max\{L_{j-1}, L_j\} &+ q_{0,k+m}\binom{m}{0}\binom{k}{0} \cdot 0 \\
&+ q_{1,k+m}\left[\binom{m}{1}\binom{k}{0}L_j + \binom{m}{0}\binom{k}{1}L_{j-1}\right] \\
&+ \cdots \\
&+ q_{u,k+m}\left[\binom{m}{u}\binom{k}{0}L_j + \sum_{l=1}^{u-1}\binom{m}{l}\binom{k}{u-l}\max\{L_{j-1}, L_j\}\right. \\
&\qquad\qquad \left. + \binom{m}{0}\binom{k}{u}L_{j-1}\right] \\
&+ \cdots \\
&+ q_{k+m,k+m}A(k,j,m). \qquad (3.17)
\end{aligned}
$$

The first term $\max\{L_{j-1}, L_j\}$ *in* (3.17) *is due to the first XOR-ed transmission of newly coded messaged where smaller subfiles are zero-padded, hence we need to use maximum length as transmission rate.*

The term $q_{0,k+m}\binom{m}{0}\cdot\binom{k}{0}\cdot 0$ *is when all of users receives message successfully which leads zero transmission rate. The term* $q_{1,k+m}\left[\binom{m}{1}\cdot\binom{k}{0}\cdot L_j+\binom{m}{0}\cdot\binom{k}{1}\cdot L_{j-1}\right]$ *is when one of* $k+m$ *users encounters an erasure. If erasured subfile is from the first message, its required transmission rate will be* L_j*. If the erasured subfile is from the second message, its required transmission rate will be* L_{j-1}*. Number of such an erasure pattern is determined by the binomial terms. The term* $q_{2,k+m}\Big[\binom{m}{2}\binom{k}{0}L_j+\binom{m}{1}\binom{k}{1}\max\{L_{j-1},L_j\}+\binom{m}{0}\binom{k}{2}L_{j-1}\Big]$ *is when two targeted users encounter erasure. When all of the erasures are in the first message, required transmission rate is* L_j*, otherwise* L_{j-1}*. In other cases, we need to use* $\max\{L_j,L_{j-1}\}$ *since there is an erasure in both messages. Other erasure patterns can be determined in a similar way.*

When all of the users encounter erasures (corresponding to the term $q_{k+m,k+m}A(k,j,m)$ *in* (3.17)*), this event will be equal to the beginning of the transmission. Hence a general formulation for* $A(k,j,m)$ *is obtained as* (3.16) *where* $\binom{n}{m}=0$ *for* $n<m$*,* $n<0$ *and* $m<0$*. Since upper bound is analyzed,* $\max\{L_{j-1},L_j\}$ *is used as size of newly coded messages in all cases. Thus, a gain on transmission rate* $R_{gain}(M,\epsilon)$ *is obtained as* (3.14)*.*

Finally, the upper bound for total transmission rate $R_{greedy}(M,\epsilon)$ *is calculated by subtracting* $R_{gain}(M,\epsilon)$ *from* $R_{SSM}(M,\epsilon)$*.* □

In Theorem 2, we derive the result on the average transmission rate of the grouped greedy coded caching algorithm by considering only certain coding opportunities among the erased subfiles. Thus, some new multicasting opportunities are ignored, and the result is an upper bound on the transmission rate. Also, for smaller erasure probabilities, the expected number of erased subfiles after the first transmission is lower, hence the amount of additional coding opportunities is limited. Hence, the derived upper bound is expected to be tight for smaller erasure probabilities and can be used for the expected transmission rate analysis of the grouped greedy coded caching.

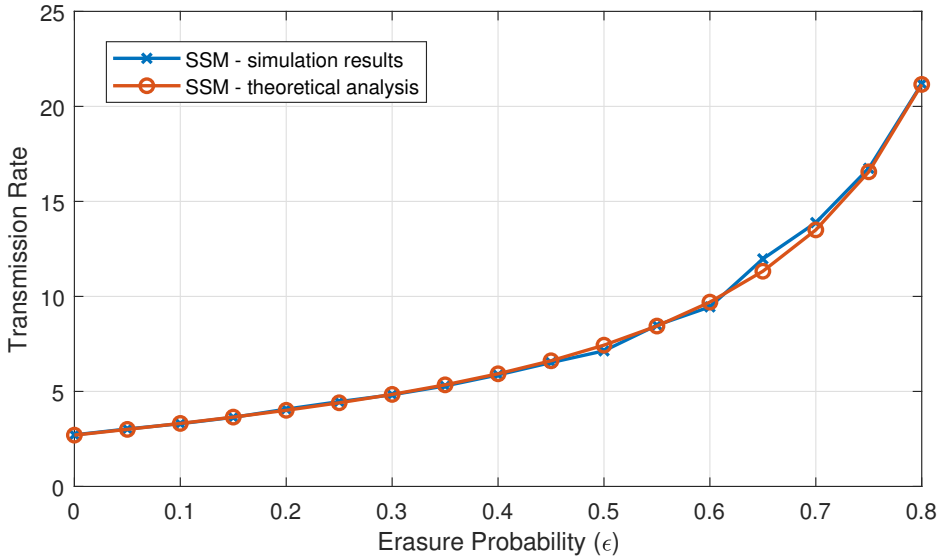

Figure 3.2: Theoretical analysis and simulation results of the transmission rate for $N = K = 8$, and $M = 2$ with different erasure probabilities with the SSM algorithm.

3.5 Numerical Examples

In this section, we numerically evaluate the performance of the proposed algorithms.

Consider a system where $K = 8$ users are connected to a central server through a packet erasure channel. The number of files in the server is $N = 8$, and each user randomly picks and stores $M/N = 1/4$ fraction of each file during the placement phase.

Fig. 3.2 illustrates the transmission rate of the SSM algorithm both via simulation and analysis results as a function of the erasure probability ϵ. It is observed that the theoretical result given by (3.3) matches the simulation results well. However, beyond a certain erasure probability, the average transmission rate of the greedy approach increases significantly which suggests a need for new approaches which can exploit additional multicasting opportunities among the erased subfiles.

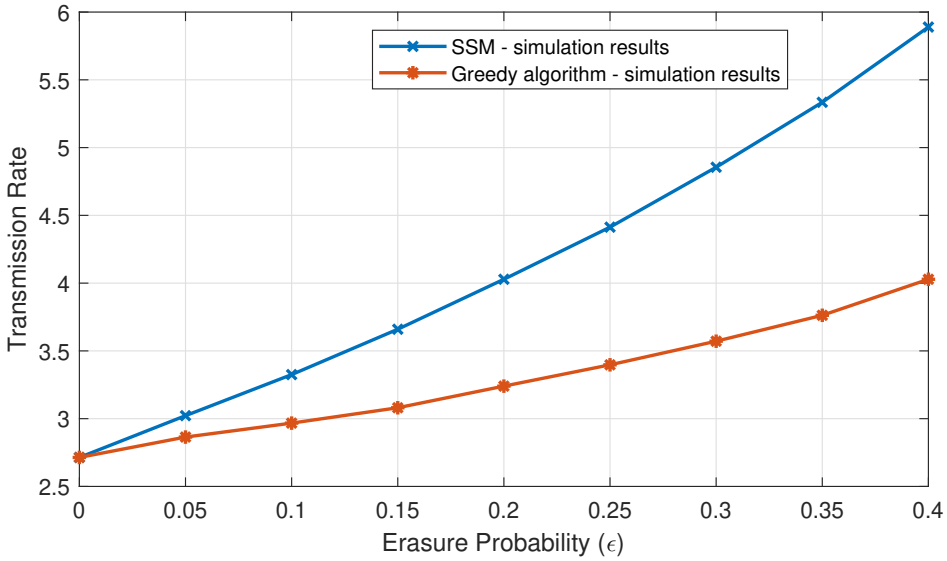

Figure 3.3: Simulation results of the transmission rate for $N = K = 8$, and $M = 2$ with different erasure probabilities with the SSM and greedy algorithm.

In Fig. 3.3, we present simulation results for the proposed greedy coded caching and the SSM algorithm. As expected, the proposed algorithm performs better than the SSM approach since it takes advantage of new multicasting opportunities. It is worth noting that, as the erasure probability gets higher, the difference between the greedy and the SSM algorithms becomes more apparent, since a higher erasure probability induces more multicasting opportunities involving more number of users.

In Fig. 3.4, we compare the performance of the greedy and grouped greedy coded caching. Since the grouped greedy approach only evaluates the multicasting opportunities among companion and successive subfiles, it may miss some of the multicasting opportunities involving other subfiles. Hence, it results in a higher transmission rate than that of the greedy approach. However, it may still be preferable because of its lower complexity. Also, as shown in Fig. 3.4, for smaller erasure probabilities, the upper bound on the transmission rate of the grouped greedy algorithm is very close to the simulation results, particularly, for small erasure probabilities.

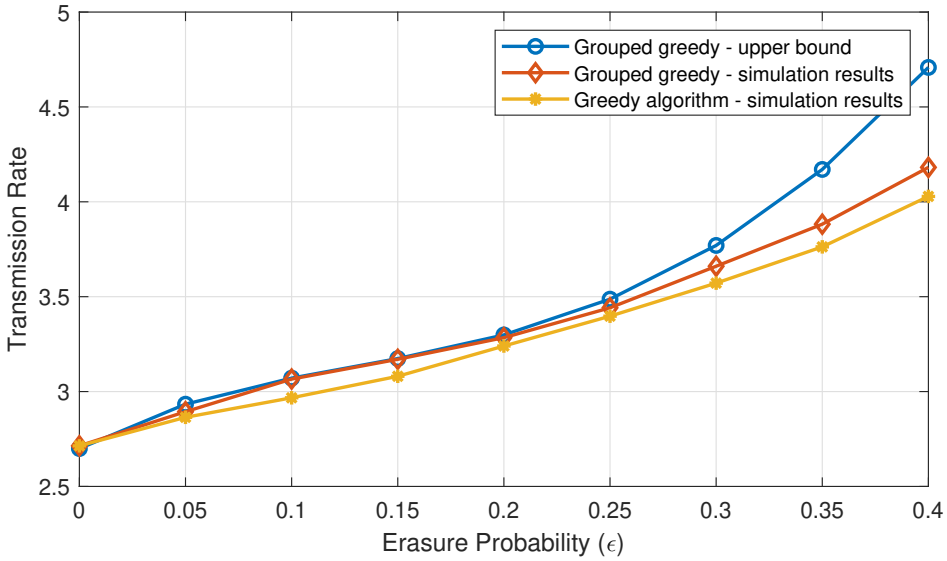

Figure 3.4: Upper bound of the grouped greedy coded caching and simulation results for $N = K = 8$ with different erasure probabilities for the greedy and the grouped greedy coded caching algorithms.

3.6 Chapter Summary

In this chapter, we have studied coded caching over packet erasure channels where each user sees an independent channel. We propose three algorithms for the delivery phase: the SSM algorithm, a greedy algorithm, and a grouped greedy algorithm for packet erasure channels. For the SSM algorithm, we have shown that our analytical calculations match well the simulation results. While grouped greedy coded caching gives a slightly higher transmission rate than the greedy algorithm, it has a lower complexity. We have also obtained an upper bound on the transmission rate of the grouped greedy coded caching algorithm, which is tight for small erasure probabilities.

Chapter 4

Coded Caching with User Grouping over Wireless Channels

In this chapter, we follow a coded caching model where the placement phase is performed in a decentralized manner and the delivery phase takes place over a wireless fading channel. Different from [17], which considers long-term average rates, our interest is to study non-ergodic channels and minimize the transmission time by letting some of the weak users to be in outage. With a fixed outage probability, we formulate an optimization problem to reduce the total transmission time by grouping the participating users to overcome the detrimental effects of channel fading. We also develop a locally optimal iterative algorithm to compute the signal to noise ratio (SNR) thresholds. Furthermore, we quantize the SNR thresholds, and model the optimization process with the quantized thresholds as a shortest path problem, and obtain a reduced complexity solution.

The chapter is organized as follows. Section 4.1 introduces the system model and preliminaries. The optimization problem for grouping users is studied in Section 4.2, and an iterative algorithm to determine the SNR thresholds is proposed. The simplified approach to the problem by quantizing the possible threshold values and using the shortest path model is presented in Section 4.3. Performance of the proposed algorithms are studied via simulations in Section 4.4, and the

chapter is summarized in Section 4.5.

4.1 System Model and Preliminaries

We consider a system which contains a server with N files each of size F bits connected through a fading channel to K users. Users are equipped with local caches which are able to store MF bits. The normalized cache size for each user is defined as $m = M/N$ which is the ratio of cache size to the total number of files in the server. We consider the decentralized coded caching framework introduced in [2], where the placement phase is performed during the off-peak hours over an error-free shared link. However, the delivery phase takes place over a wireless (fading) channel.

We model the (wireless) channel between the server and the users during the delivery phase as a non-ergodic (quasi-static) fading channel. We examine the coded caching system where the server only knows the channel statistics to determine the SNR thresholds of user groups. During the delivery phase, it receives limited feedback from the users indicating their groups with a low overhead (a few bits of feedback). Since the channels are non-ergodic, the server chooses not to serve the users with low SNRs, and puts them in outage. If user $k \in [K]$ is not in outage, it receives $\mathbf{y}_k = \sqrt{\rho_k} h_k \mathbf{X} + \mathbf{n}$ where the coded message $\mathbf{X}$ is constructed according to [2], components of $\mathbf{n}$ are independent and identically distributed (i.i.d.) zero mean circularly symmetric complex Gaussian random variables, i.e., each follows $\sim \mathcal{CN}(0, 1)$. h_k's are fading coefficients which are independent complex random variables. For instance, if h_k's are zero mean circularly symmetric complex Gaussian random variables with variance $1/2$ per dimension, then the channels are Rayleigh fading with ρ_k denoting the average SNR for user k.

Receiver $k \in [K]$ reconstructs its demanded content based on $\mathbf{y}_k$, cache content, and the demand vector, and an error occurs for the user when the reconstructed content is different than the requested one.

4.2 Grouping Users Using Channel Statistics

In this section, we examine the optimal user grouping problem for the case where the server only has access the channel statistics of users to form the user groups. First, we note that for multicast transmission to a group of users, the capacity of the channel is restricted by the user with the worst channel condition [21], and conditioned on the channel gains, it is given by $R(\mathbf{\Lambda}_{n\in[N_g]}) = \log_2(1+\min_{n\in[N_g]} \Lambda_n)$ where N_g is the number of users in consideration, and Λ_n is the instantaneous SNR of user n. Therefore, the rate $R(\mathbf{\Lambda}_{n\in[N_g]})$ dictates the reliable transmission limit. Obeying this limitation, the corresponding transmission takes $T_{\text{req}} = \frac{T(m,N_g)}{R(\mathbf{\Lambda}_{n\in[N_g]})}$ units of time [17] where the normalized length of the coded message to serve N_g users each equipped with a normalized cache size of m is $T(m, N_g) = \frac{1-m}{m}(1 - (1-m)^{N_g})$ [21].

By taking into consideration the limitation due to the user with the worst channel condition, creating a single coded message for all the users which are to be served may increase the total transmission time dramatically. We argue that this can be alleviated by grouping users and creating specific coded messages to the different groups of users experiencing instantaneous SNRs close to each other, and based on this intiution, we formulate an optimization problem to minimize the total required transmission time. Also, since the server does not have access to the instantaneous SNR values as it only receives limited feedback from users indicating their group, we use a single transmission rate for every user in the same group, and transmit the coded messages accordingly.

4.2.1 Optimization Problem for Threshold Determination

Since the individual links between the server and the users are modeled as non-ergodic channels, Shannon type capacity is zero, hence we adopt an outage capacity formulation. According to this model, for a given rate R, the outage probability for user $k \in [K]$ is $P_{\text{out}} = P(C(\lambda_k) < R)$ where $C(\lambda_k) = \log_2(1 + \lambda_k)$ where $\lambda_k = |h_k|^2\rho_k$ is the effective SNR of user k with cumulative distribution

function (CDF) $F_k(\cdot)$.

Let us denote the SNR thresholds for user groups by x_j with $j = 0, 1, \ldots, t-1$ where t is the number of groups and x_0 is the SNR threshold that determines the users in outage, i.e., users with a lower instantaneous SNR than x_0 are not served. The expected number of users that are not served and number of users in each group can be determined using the following propositions, whose proofs are straightforward.

Proposition 1 *For independent fading links, the outage SNR threshold x_0 with a given ratio of the expected number of users that are not served (P_{out}) can be obtained by solving*

$$P_{out} = \frac{1}{K}\left(K - \sum_{k=1}^{K} 1 - F_k(x_0)\right), \tag{4.1}$$

where $F_k(\cdot)$ is the CDF of the effective SNR for user $k \in [K]$, and K is the total number of users.

Proposition 2 *For independent fading links, the expected number of users in a group formed by users with SNR $\in [x_{j-1}, x_j)$ (denoted by K_j) is given by*

$$K_j = \sum_{k=1}^{K} \left(F_k(x_j) - F_k(x_{j-1})\right), \tag{4.2}$$

where $j = 0, 1, \ldots, t$, $x_{-1} = 0$, $x_t = \infty$, and K_0 is the expected number of users in outage.

Note that, if $h_k \sim \mathcal{CN}(0,\, 1)$, we have Rayleigh fading channels, which result in $P_{\text{out}} = \frac{1}{K}\left(K - \sum_{k=1}^{K} e^{-\frac{x_0}{\rho_k}}\right)$, and $K_j = \sum_{k=1}^{K}\left(e^{-\frac{x_{j-1}}{\rho_k}} - e^{-\frac{x_j}{\rho_k}}\right)$.

As stated before, each group's multicast capacity is limited by the worst user in that group. Hence, to calculate the transmission time to serve the requests of group j, we focus on the coded message constructed for that group with a normalized length of $T(m, K_j)$, and the worst user's capacity in the group $\log_2(1 + x_{j-1})$.

We emphasize that the server does not have access to the instantaneous SNR values as each user only sends $\lceil \log_2(t) \rceil$ bits of feedback indicating their group along with their demands at the beginning of the delivery phase to keep the communication overhead low. This limited feedback enables the server to create coded messages for different user groups. Hence, it is clear that the total required time to satisfy all the requests except for those in outage can be written as

$$T_{\text{req}} = \sum_{i=1}^{t} \frac{T(m, K_i)}{\log_2 (1 + x_{i-1})}. \tag{4.3}$$

Given the number of groups t, we are interested in minimizing the required transmission time T_{req} over the SNR threshold vector $\boldsymbol{x} = [x_1 \ \ldots \ x_{t-1}]$, i.e., we need to solve the following optimization problem.

$$\begin{aligned} &\underset{x_1 \cdots x_{t-1}}{\text{minimize}} && \sum_{i=1}^{t} \frac{T(m, K_i)}{\log_2 (1 + x_{i-1})} \\ &\text{subject to} && x_{i-1} \leq x_i, \ i = 1, \ldots, t-1. \end{aligned} \tag{$\mathbf{P}_1$}$$

Since $\mathbf{P}_1$ does not have any appealing structure, it requires a brute force search to determine the continuous valued parameters $x_1, x_2, \ldots, x_{t-1}$, which is infeasible for nontrivial values of t. To reduce the computational complexity, we propose an iterative algorithm to find the threshold values. The proposed approach is practical, and it is guaranteed to converge to a locally optimal solution.

4.2.2 An Efficient Locally Optimal Algorithm for Threshold Determination

We notice that each parameter to be optimized in $\mathbf{P}_1$ is only present in two different terms of the objective function. Let us now focus on the terms that involve a given threshold value x_j of the summation. That is, for the SNR threshold x_j, we consider

$$\bar{T}_{x_j} = \frac{T(m, K_j)}{\log_2 (1 + x_{j-1})} + \frac{T(m, K_{j+1})}{\log_2 (1 + x_j)}. \tag{4.4}$$

Recall that K_j's can be calculated using (4.2). It is clear that if j is even the only other dependence is to the adjacent odd indexes and vice versa. Hence, to simplify the optimization process, we can exploit this structure to obtain sub-problems which only depend on a single parameter and solve the original problem by using an iterative approach. Our proposed solution proceeds as follows: we first select arbitrary initial values for x_j's. We minimize (4.4) over the odd indexed thresholds by fixing the even indexed ones. And then, we apply same procedure by reversing the roles of odd and even indices, i.e., we fix the odd indexed thresholds, and determine optimal values for the even indexed ones. We continue these iterations for a predetermined number of times to find the final set of (optimal) SNR thresholds.

The proposed approach divides the initial problem into two sub-problems each containing $t/2$ minimization problems with a single unknown parameter in each step, hence simplifying the search considerably. We also note that, the overall cost function reduces at each iteration, and since it is bounded from below, the algorithm is guaranteed to converge by the Monotone Convergence Theorem [44]. However, since the problem is not convex, the algorithm is only guaranteed to be locally optimal.

To obtain the best solution, we repeat the above described steps for different number of thresholds, and the solution which gives the minimum transmission time is selected as the final set of SNR thresholds to be employed.

4.3 A Reduced Complexity User Grouping Approach

In this section, we propose a reduced complexity approach for user grouping by reformulating the non-convex optimization problem $\mathbf{P}_1$ using quantized threshold values and solving a corresponding integer program.

We first quantize the SNR threshold values using a large number of possible

groups, denoted by q. Although different quantizers can be used, we utilize a uniform quantizer where τ_i is the i^{th} quantization level with $i = 1, \ldots, q-1$. With this discretization, the problem $\mathbf{P}_1$ can be converted into the following integer program which minimizes the total transmission time over the set of SNR thresholds $\boldsymbol{x}$.

$$\begin{aligned} \underset{x_1 \cdots x_{q-1}}{\text{minimize}} \quad & \sum_{i=1}^{q} \frac{T(m, K_i)}{\log_2(1 + x_{i-1})} \\ \text{subject to} \quad & x_{i-1} \leq x_i, \ i = 1, \ldots, q-1, \\ & x_j \in \{\tau_1, \ldots, \tau_{q-1}\}, j = 1, \ldots, q-1, \end{aligned} \tag{$\mathbf{P}_2$}$$

where x_0 and K_i's are determined using (4.1) and (4.2), respectively. Note that, even though we start with a large number of possible groups q, the solution determines the optimal number of groups which minimizes the overall transmission time.

Although formulating $\mathbf{P}_2$ as an integer program and utilizing quantized thresholds decreases the complexity of the brute force search solution, the problem still does not have any appealing structure to be exploited. Therefore, to interpret the optimization problem further, we construct a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ which comprises of a set $\mathbf{V}$ of vertices and a set $\mathbf{E}$ of edges. In this model, each vertex represents a quantization level for the SNR thresholds and each edge carries the corresponding transmission time by choosing the thresholds according to its incident vertices. With this, the minimization problem given in $\mathbf{P}_2$ becomes an instance of a well-known graph theory problem, namely, the shortest path problem which can be solved in polynomial time if there are no negative dicycles [45]. In this way, the worst case complexity can be reduced to $O(q^2)$ using existing approaches in the literature. In the shortest path problem, the aim is to minimize the length of the path (transmission time) between the starting vertex and the terminating one. It is clear that determination of shortest path automatically finds the optimal number of user groups, and hence there is no need to try different number of groups as needed for the case of the iterative approach proposed in Section III.

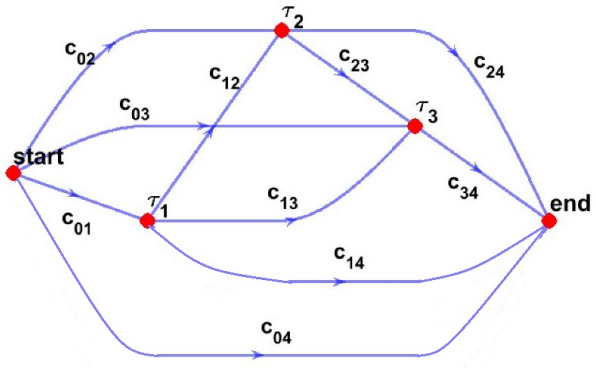

Figure 4.1: Sample of a directed graph with 3 quantization levels and edge costs c_{ij}.

To compose the corresponding graph and determine the cost of each edge, the expected number of users whose SNRs are between τ_i and τ_j with $\tau_i \leq \tau_j$ can be found as

$$K_{ij} = \sum_{k=1}^{K} \left(F_k(\tau_j) - F_k(\tau_i) \right), \tag{4.5}$$

where τ_i's are the possible quantization levels with $i = 1, \ldots, q-1$, and $F_k(\cdot)$ is the CDF of k-th user's SNR. And, the cost (required transmission time) of each edge (for the users whose SNRs are between τ_i and τ_j) can be calculated as

$$c_{ij} = \frac{T(m, K_{ij})}{\log_2(1+\tau_i)}. \tag{4.6}$$

As illustrated in Fig. 4.1, c_{ij}'s represent the cost of each edge. If the optimal solution contains only the starting and terminating vertices, there will only be a single edge in the solution. Hence, we need to create a single coded message for all the users using the algorithm introduced in [2]. If the optimal solution contains vertices other than the starting and terminating ones, the users will be placed in multiple groups and separate coded messages will be created.

4.4 Numerical Examples

We now provide several examples of coded caching over wireless channels that illustrate the effectiveness of the developed solutions. We consider Rayleigh fading

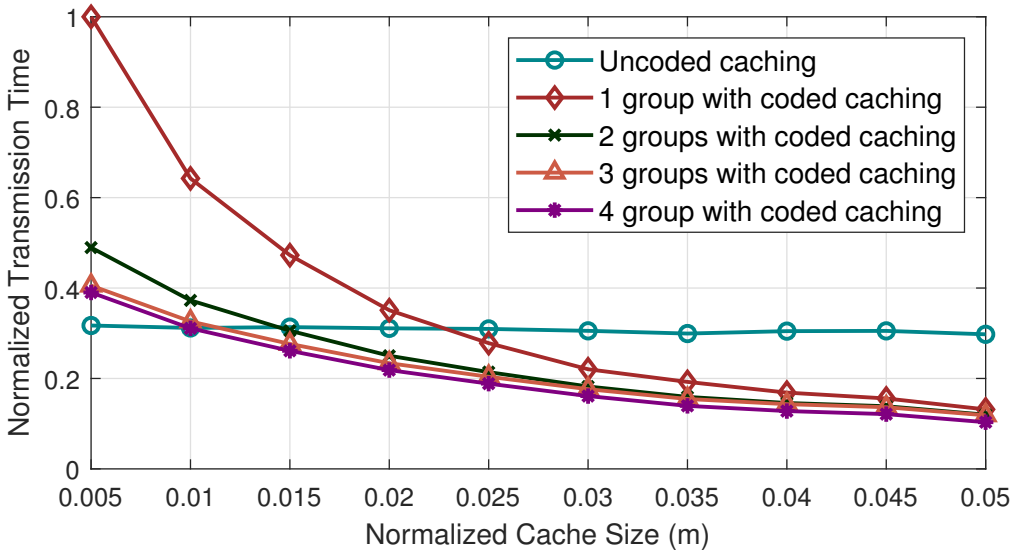

Figure 4.2: Simulation results with uncoded caching, coded caching with $t = 1, 2, 3$ and 4 groups.

for all the examples, i.e., h_k's are independent zero mean circularly symmetric complex Gaussian random variables with variance $1/2$ per dimension.

In the first example, the total number of users is $K = 400$, and the mean values of the user SNRs are taken as -3, 0, 3, and 6 dB each for a quarter of them. The server contains $N = 1000$ files, and the outage probability is set to $P_{\text{out}} = 0.05$. The normalized required transmission time obtained by solving $\mathbf{P}_1$ with brute force search for different cache capacities is depicted in Fig. 4.2. The results show the performance of the uncoded (traditional) caching, coded caching with no grouping, and coded caching with $2, 3$ and 4 user groups. It is observed that even with small cache sizes coded caching can be effective, and allowing for a greater number of user groups results in lower normalized transmission times, especially for smaller cache sizes.

Next, we consider a network consisting $K = N = 1000$ users. User SNRs are exponentially distributed with means -3, 0, 3, and 6 dB, each for a quarter of users, and $P_{\text{out}} = 0.05$. For the iterative algorithm, the maximum number of thresholds t_{max} is set to 8. The number of quantization levels in the shortest path algorithm is determined by quantizing the SNR range from the outage SNR

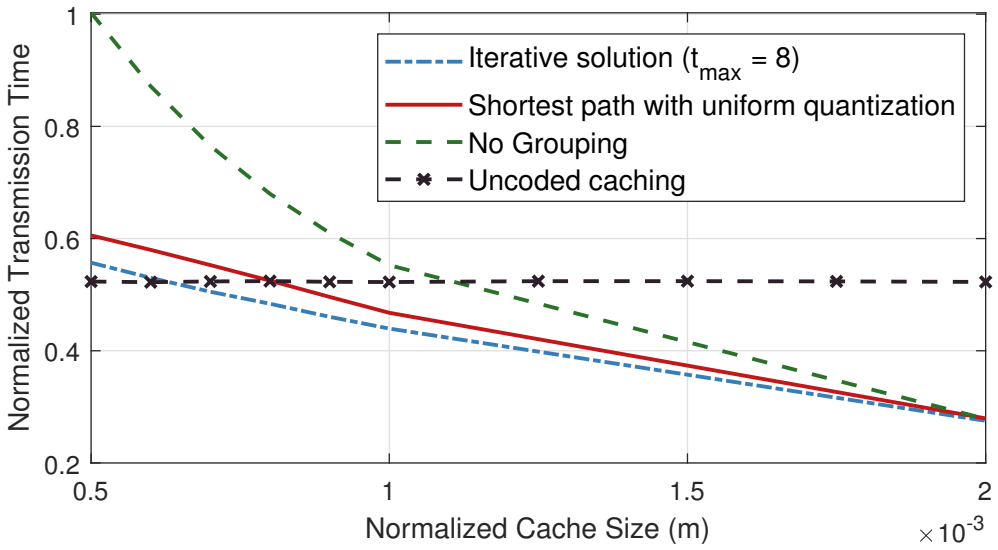

Figure 4.3: Effect of normalized cache size (m) with $K = 1000$.

threshold x_0 to 30 dB uniformly with a step size of 0.3 dB. Fig. 4.3 illustrates the normalized transmission times as a function of the normalized cache size for different cases. As expected, both of the proposed algorithms outperform the coded caching solution without any user grouping, and are very effective (again, particularly, for smaller cache sizes). We also note that while the performance of the shortest path algorithm is slightly inferior to the one using the iterative approach to find the optimal SNR thresholds, it may still be preferable due to its reduced complexity (as it avoids the many line searches needed for determination of continuous threshold values).

As another example, we consider a network whose parameters are the same as the one above, except that we set $N = K = 5000$, and the user SNRs are exponentially distributed with means -6, 0, 6, and 12 dB, each for $K/4$ users. The normalized transmission times are depicted as a function of the normalized cache size in Fig. 4.4. We observe that the performance with only the channel statistics is slightly inferior to the one with the exact SNR knowledge as shown in Fig. 4.4.

In Fig. 4.5, the number of vertices in the optimal solution of the shortest path problem (i.e., the number of selected thresholds) is given, which shows that, for

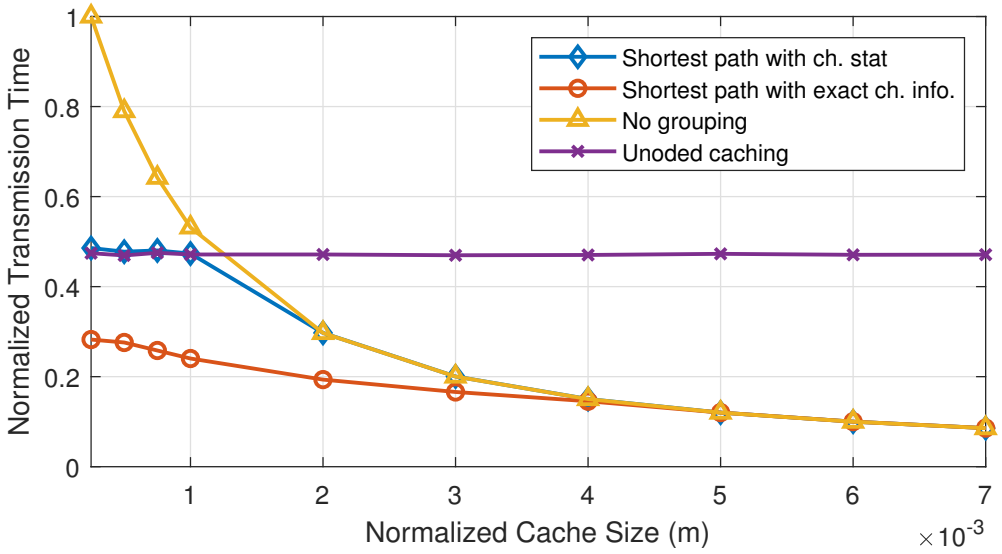

Figure 4.4: Effect of normalized cache size (m) with $K = 5000$ on the normalized transmission time.

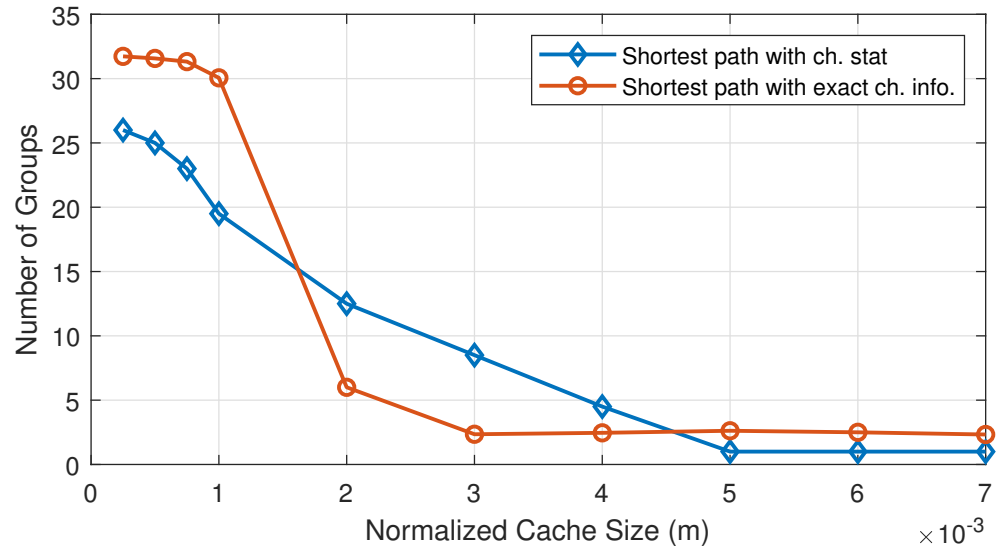

Figure 4.5: Effect of normalized cache size (m) with $K = 5000$ on the number of groups.

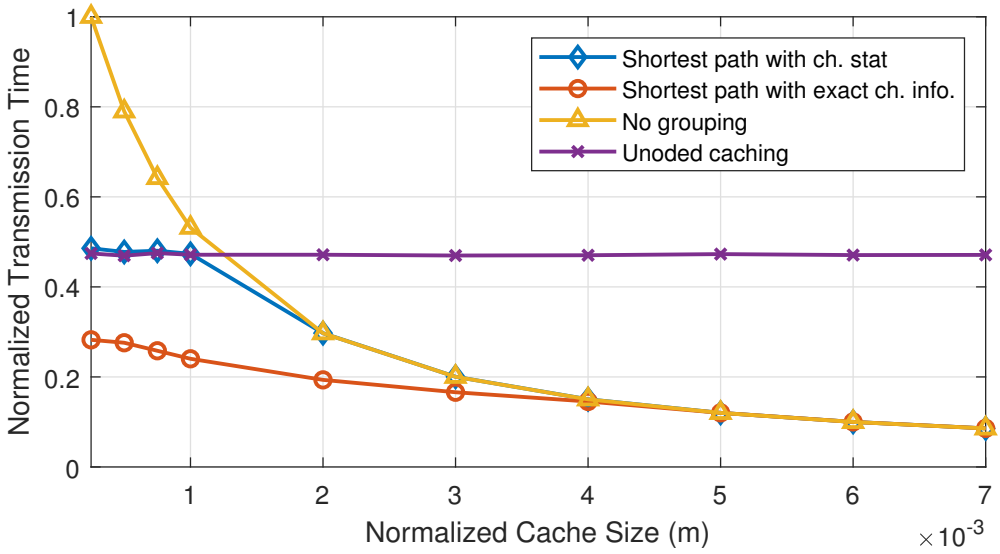

Figure 4.6: Effect of quantization level (q) with $K = 5000$ on the normalized transmission time.

relatively low normalized cache sizes, grouping gain obtained by serving the weak users separately from the strong ones is higher.

Next, without changing the other parameters of the network, we set the normalized cache size to $m = 0.0005$, and consider the number of quantization levels as a variable by changing the quantization step size. Fig. 4.6 shows the performance results for both cases of exact SNR knowledge and using channel statistics only. As in the previous examples, the performance of knowing only the channel statistics is slightly worse than the one with the exact SNR knowledge. Also, it is obvious that, increasing the number of quantization levels decreases the normalized transmission times as it allows for further grouping opportunities for the users.

Finally, in Fig. 4.7, we consider a network whose parameters are the same as the one above, except all the user SNRs are exponentially distributed with mean -6 dB. We remark that, even when all of the users have same channel statistics, as the instantaneous SNRs are different due to channel fading, the proposed grouping approach attains lower transmission time than no grouping and uncoded caching. For instance, when $m = 10^{-3}$, the transmission time of the

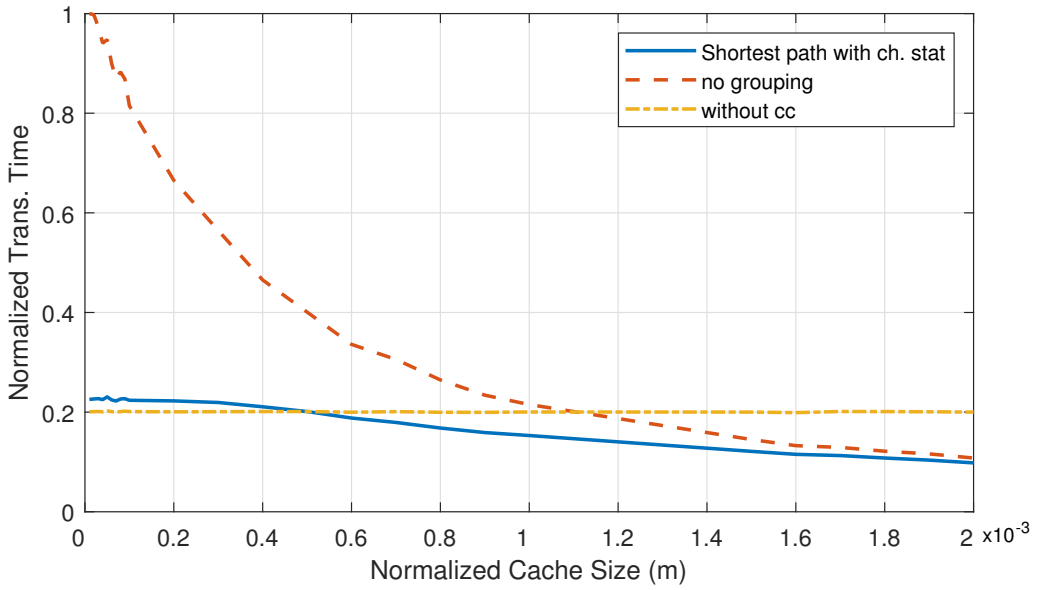

Figure 4.7: Effect of normalized cache size (m) on shortest path solution with same user channel statistics.

shortest path algorithm is 28.9% and 24.6% less than no grouping and uncoded caching, respectively.

4.5 Chapter Summary

In this chapter, we propose grouping of users for coded caching over non-ergodic fading channels to minimize the total transmission time by utilizing only the channel statistics users for threshold determination, and relying on a few bits of feedback from each user indicating the group it belongs to. The first proposed approach for the threshold determination is a locally optimal iterative solution, while in the second one, we quantize the possible threshold values and convert the original problem into a shortest path problem enabling highly efficient solutions with a slight sacrifice in performance. The results demonstrate that user grouping for coded caching over wireless channels is highly advantageous, particularly, when the cache sizes are small.

Chapter 5

Machine Learning at the Wireless Edge with Low-Resolution Analog to Digital Converters

In this chapter, our main objective is to study distributed learning algorithms over wireless channels in more realistic settings considering practical implementation issues, including the channel effects, e.g., the effect of frequency selective channels and low-resolution receive chains. Hence, we model the communication link as a frequency selective fading channel, and transmit the local gradients using orthogonal frequency division multiplexing (OFDM). Furthermore, in an effort to reduce the hardware complexity and power consumption, we employ low-resolution analog to digital converters (ADCs) at the receiver side, which employs multiple (even a massive number of) receive antennas. While decreasing the resolution of ADCs reduces the implementation cost and the power consumption, it also deteriorates the performance of a communication system. From a communication theory perspective, the effects of ADCs at the base station with a massive number of antennas and orthogonal frequency division multiplexing (OFDM) is studied in [46]. The authors analyze the uplink performance of the massive multiple user MIMO systems for wideband communication. They focus on the trade-off between the quantization precision and performance of channel

estimation and data detection, show that even with coarse quantization, they achieve almost no performance loss compared to infinite resolution case. In [47], single carrier and OFDM transmission for massive MIMO with one-bit ADCs are analyzed, and achievable rate for a wideband system with a higher number of channel taps are derived. They show that the quantization causes two types of error in symbol detection: the circularly symmetric error and amplitude distortion. As the number of antennas increases, the circularly symmetric error follows a Gaussian distribution while the amplitude distortion reduces as the number of channel taps increases. Thus, they show that using one-bit ADC is effective with wideband massive MIMO. Ref. [48] investigates the quantization, clipping, and thermal noise in OFDM systems due to finite resolution ADCs by jointly analyzing quantization type, sampling rate, and filter type. In [49],low-resolution ADCs are utilized at the receiver for mmWave massive MIMO systems. They study difficulties regarding channel estimation, feedback, precoding, and signal detection with ADCs.

In [50], extreme the extreme case of one-bit ADC is analyzed. They obtain achievable throughput for a multi-user MIMO system where a large number of low-resolution ADCs is placed in the base station and model the system for the case of no CSI at both receiver and transmitter, hence CSI can be learned via coarse observations obtained by pilot transmissions. To overcome the limitation caused by finite resolution observations, they propose a channel estimation method based on Bussgang Theorem [51] and show that one-bit quantization can attain almost the same achievable rate with infinite resolution. A similar approach is used in [52], where the authors propose a channel estimation approach for frequency selective and flat fading channels based on Bussgang Therem [51] by formulation nonlinear quantization operation as a linear function. For flat fading channels, the achievable rate for a large number of users and low SNR is obtained, and significant design aspects are investigated, e.g., energy and spectral efficiency, optimal resource allocation, and the number of antennas to be used.

The chapter is organized as follows. Section 5.1 introduces the system model and preliminaries. DSGD with low-resolution ADCs is analyzed in Section 5.2

while the results are specialized to the case of 1-bit ADCs in Section 5.3. The performance of distributed machine learning systems over fading multipath channels with OFDM and low-resolution ADCs at the receiver is studied via simulations in Section 5.4, and the chapter is concluded in Section 5.5.

Notation: Throughout this chapter, the real and imaginary parts of $x \in \mathbb{C}$ are represented by x^R and x^I, respectively. We denote l_2 norm of a vector $\mathbf{x}$ by $||\mathbf{x}||_2$.

5.1 System Model

We consider a distributed ML system where each worker calculates its gradient estimate and sends it to a central PS through a multipath fading MAC with OFDM as illustrated in Fig. 5.1. At the receiver side, analog to digital conversion, OFDM demodulation, signal combining and global model parameter update are performed, and the global parameter is broadcast to the workers over an error-free link. We assume that there is no transmit side CSI, and that the PS employs multiple antennas to recover the average of the workers' gradients. With the use of many antennas, a significant amount of power at the receiver is consumed by the ADCs [53]. As the power consumption of ADCs increases linearly, and their hardware cost increases exponentially with the number of quantization bits [54], we consider a distributed learning system where receiver side ADCs have a low-resolution to keep the implementation cost and power consumption small.

In the distributed learning at the wireless edge setup, we jointly train a learning model by using iterative SGD to minimize a loss function $f(\cdot)$. During the t-th iteration, worker $m \in [M]$ calculates the gradient estimate $\mathbf{g}_m^t \in \mathbb{R}^d$ by processing its local dataset $\mathcal{B}_m$ according to $\frac{1}{|\mathcal{B}_m|} \sum_{u \in \mathcal{B}_m} \nabla f(\boldsymbol{\theta}_t, u)$ where $\boldsymbol{\theta}_t \in \mathbb{R}^d$ is the vector of model parameters, d is the number of model parameters, and $g_m^t[n]$ represents the n-th entry of the gradient estimate vector.

Since the local gradient vector has real components, we obtain the frequency

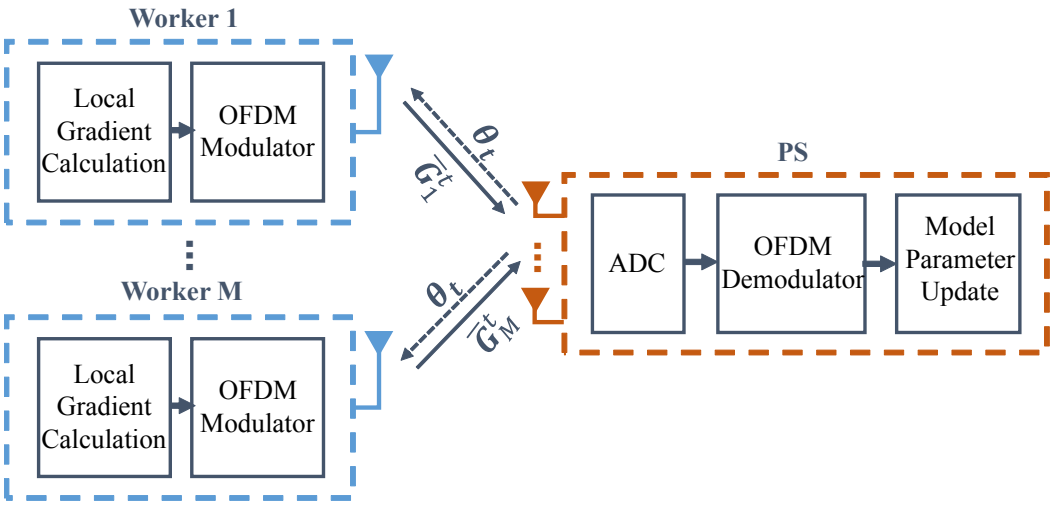

Figure 5.1: System model for distributed machine learning at the wireless edge.

domain representation of the gradients as

$$\hat{\mathbf{g}}_m^t = \left[g_m^t[1] + jg_m^t[e+1], g_m^t[2] + jg_m^t[e+2], \cdots, g_m^t[e] + jg_m^t[2e]\right], \tag{5.1}$$

where $e = \lceil d/2 \rceil$, $\hat{\mathbf{g}}_m^t \in \mathbb{R}^e$, and $g_m^t[2e]$ is assigned as zero if $d \equiv 1 \pmod 2$.

After obtaining the the frequency domain representation of the local gradient estimates, the first step is to form the OFDM signal by taking an N-point inverse discrete Fourier Transform (IDFT) of the gradient vector as

$$G_m^t[u] = \frac{1}{N} \sum_{n=1}^{N} \hat{g}_m^t[n] e^{j2\pi nu/N}, \tag{5.2}$$

for $u \in [N]$. If $e < N$, $\hat{g}_m^t[n] = 0$ for $n > e$, i.e., $\hat{\mathbf{g}}_m^t$ is zero padded if $e < N$.

To mitigate the ISI caused by the multipath, cyclic prefix (CP) addition is performed by

$$\bar{\mathbf{G}}_m^t = \left[G_m^t[N - N_{cp} + 1] \ldots G_m^t[N]\ G_m^t[1] \ldots G_m^t[N]\right], \tag{5.3}$$

where $\bar{\mathbf{G}}_m^t \in \mathbb{C}^{N+N_{cp}}$ is the OFDM word to be transmitted by the m-th worker. The resulting OFDM words are transmitted to the PS which are equipped with K receive antennas. The corresponding receive chains are equipped with a complex-valued low-resolution ADC which performs elementwise complex-valued mapping.

The PS uses the received signal to update the model and sends it back to all the receivers over an error-free link.

The channel between worker m and the k-th antenna of the PS is modeled as a wireless multipath MAC. We assume that the channel does not change during the transmission of one OFDM word, while it may be different for different OFDM words. The impulse response of the channel is

$$h_{mk}^t[n] = \sum_{l=1}^{L} h_{mkl}^t \delta[n - \tau_{mkl}], \tag{5.4}$$

where $n \in [N + N_{cp}]$, L is the number of channel taps, τ_{mkl} is the time delay and $h_{mkl}^t \in \mathbb{C}$ is the gain of the l-th channel tap from the m-th worker to the k-th antenna of the PS. We assume that h_{mkl}^t is a zero-mean complex Gaussian variable with $\mathbb{E}\left[(h_{mkl}^t) \cdot (h_{m'k'l'}^t)^*\right] = 0$ for $(m, k, l) \neq (m', k', l')$, and $\mathbb{E}\left[|h_{mkl}^t|^2\right] = \sigma_{h,l}^2$, i.e., all the channel taps experience Rayleigh fading.

At the k-th receive chain, after removing CPs, the n-th entry of the received vector at the output of the ADC during iteration t is written as

$$Y_k^t[n] = \sum_{m=1}^{M} \sum_{l=1}^{L} h_{mkl}^t G_m^t[n - \tau_{mkl}] + z_k^t[n], \tag{5.5}$$

where the additive noise terms $z_k^t[n] \in \mathbb{C}$ are independent and identically distributed (i.i.d.) circularly symmetric zero mean complex Gaussian random variables, i.e., $z_k^t[n] \sim \mathcal{CN}(0, \sigma_z^2)$ for $k \in [K]$. Estimate of the gradient vector is obtained by processing the quantized input signal to recover the average of the gradient estimates by $\frac{1}{M}\sum_{m=1}^{M} \mathbf{g}_m^t$. At the PS, the model parameter is updated according to $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mu_t \frac{1}{M}\sum_{m=1}^{M} \mathbf{g}_m^t$, and it is shared with the workers over an error-free link.

5.2 DSGD with b-bit Low-Resolution ADCs

In this section, we analyze the effect of multipath fading and low-resolution ADCs at the receiver on the convergence of the distributed ML solutions. For ease of notation, we drop the subscripts referring to iteration count t.

At each receive chain, after removing CPs of the received OFDM word, a complex-valued b-bit low-resolution ADC performs quantization with the quantizer output denoted by $Q(\cdot)$. A complex-valued ADC consists of two parallel real-valued ADCs with quantization function $Q_b(\cdot)$ that independently quantizes the real and imaginary parts into $\beta = 2^b$ reconstruction levels. The reconstruction levels are denoted by $\hat{\mathbf{a}} = [\hat{a}_1 \ \hat{a}_2 \cdots \hat{a}_\beta] \in \mathbb{R}^\beta$ while the boundaries of the quantization regions are denoted by $\hat{\mathbf{x}} = [\hat{x}_1 \ \hat{x}_2 \cdots \hat{x}_{\beta+1}] \in \mathbb{R}^{\beta+1}$ where $\hat{x}_1 = -\infty$ and $\hat{x}_{\beta+1} = +\infty$ for convenience. Also, we have, $\hat{a}_i < \hat{a}_j$, if $1 \leq i < j \leq \beta$, $\hat{x}_i < \hat{x}_j$ if $1 \leq i < j \leq \beta + 1$, and $\hat{x}_i \leq \hat{a}_j < \hat{x}_k$ if $1 \leq i \leq j < k \leq \beta + 1$. The corresponding real valued quantizer is $Q_b(z) = \hat{a}_i$ for $\hat{x}_i \leq z < \hat{x}_{i+1}$, $i \in [\beta]$, $z \in \mathbb{R}$. The complex-valued ADC operation can be expressed as

$$Q(x) = Q_b(x^R) + jQ_b(x^I). \tag{5.6}$$

Hence, at the k-th receive chain, the received signal at time n is given by

$$R_k[n] = Q_b\left(\sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl}G_m[n - \tau_{mkl}] + z_k[n]\right). \tag{5.7}$$

In [55], it is shown that, if the input data which forms the OFDM word is i.i.d. and bounded, the convex envelope of the OFDM word weakly converges to a Gaussian random process as the number of subcarriers goes to infinity through an application of central limit theorem (CLT). Similarly, if we assume that the elements of the gradient vector in the learning process are i.i.d. and bounded, then the real and imaginary parts of the baseband OFDM word obtained from the gradient vector can be modeled as independent zero-mean stationary Gaussian processes. Therefore, we model the OFDM words as Gaussian processes. As a verification, in Fig. 5.2 and Fig. 5.3, we provide an exemplary histogram of the OFDM word samples obtained through the 10-th iteration of a certain learning task with our setup, demonstrating the received samples are approximately Gaussian.

Modeling the real and imaginary parts of the received OFDM words with identical autocorrelation functions $R_G(n_1, n_2)$ where $R_G(n, n) = \sigma_G^2$, and using Bussgang's theorem [51], we can decompose the quantized signal into two parts

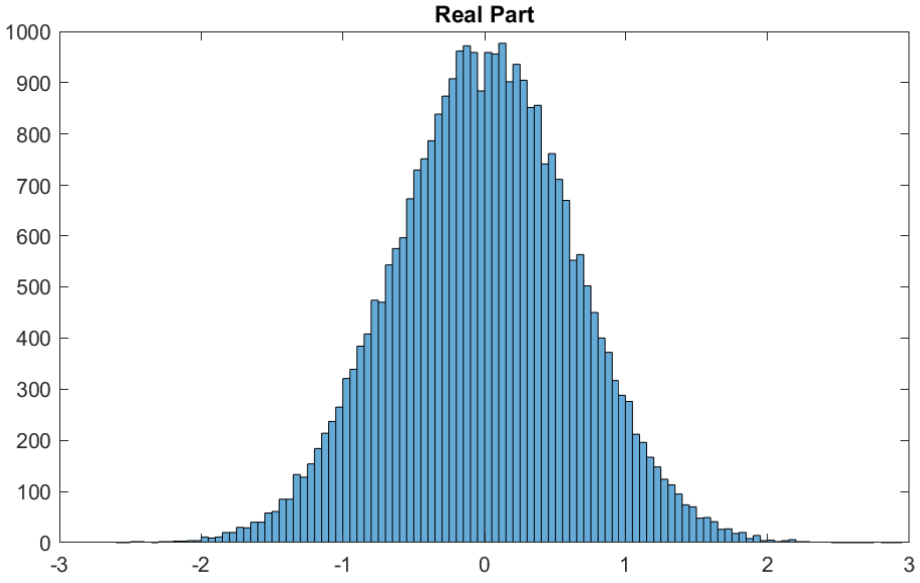

Figure 5.2: Histogram of the real part of the received OFDM word.

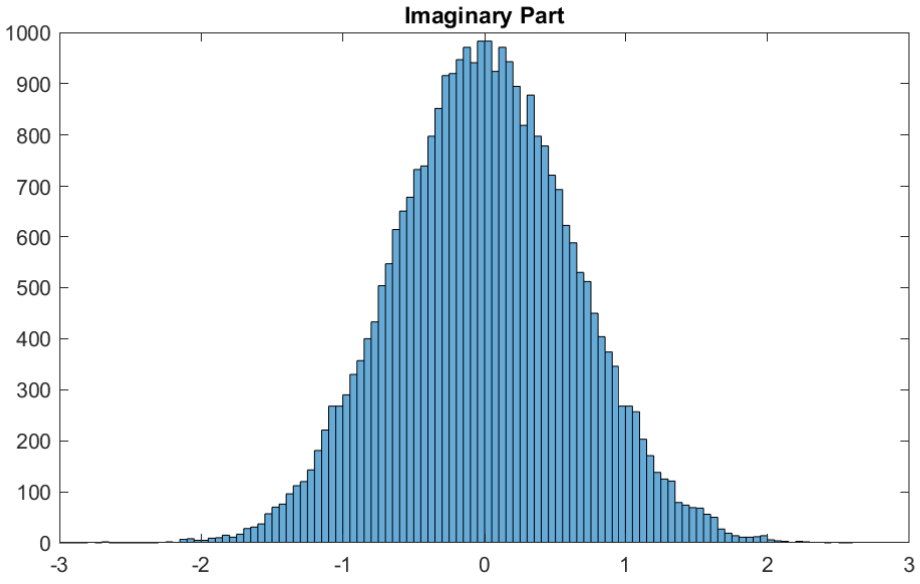

Figure 5.3: Histogram of the imaginary part of the received OFDM word.

as the desired signal component and quantization distortion which is independent of the desired signal. Analytically, we can write the quantized signal as

$$R_k[n] = (1-\eta_k)\left(\sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl}G_m[n-\tau_{mkl}] + z_k[n]\right) + w_q[n], \tag{5.8}$$

where η_k is the distortion factor, and $w_q[n]$ is a non-Gaussian distortion noise whose variance is $\sigma^2_{w_q}$. The detailed calculations for the distortion factor and quantization noise are given in the following subsection. If we define the total effective noise caused by the channel and quantization as

$$w_k[n] = (1-\eta_k)z_k[n] + w_q[n], \tag{5.9}$$

the output of the complex ADC can be written as

$$R_k[n] = (1-\eta_k)\sum_{m=1}^{M}\sum_{l=1}^{L} h_{mkl}G_m[n-\tau_{mkl}] + w_k[n], \tag{5.10}$$

where $w_k[n]$ is non-Gaussian total noise with variance $\sigma^2_{w_k} = (1-\eta_k)^2\sigma_z^2 + \sigma^2_{w_q}$. To perform OFDM demodulation, we take the discrete Fourier Transform (DFT) of (5.10) which gives

$$r_k[i] = (1-\eta_k)\sum_{m=1}^{M} H_{mk}[i]g_m[i] + W_k[i], \tag{5.11}$$

where $H_{mk}[i]$'s are the channel gains from the m-th worker to the k-th receive chain for the i-th subcarrier, given by

$$H_{mk}[i] = \sum_{n=0}^{N-1} h_{mk}[n]e^{-j2\pi in/N} \tag{5.12a}$$

$$= \sum_{n=0}^{N-1}\left(\sum_{l=1}^{L} h_{mkl}\delta[n-\tau_{mkl}]\right)e^{-j2\pi in/N} \tag{5.12b}$$

$$= \sum_{l=1}^{L} h_{mkl}e^{-j2\pi i\tau_{mkl}/N}. \tag{5.12c}$$

Since the channel taps are Rayleigh fading, $H_{mk}[i]$'s are zero-mean Gaussian random variables with variance $\sigma_H^2 = \sum_{l=1}^{L}\sigma_{h,l}^2$.

Taking DFT of the effective noise, $W_k[i]$ is evaluated as

$$W_k[i] = \sum_{n=0}^{N-1} w_k[n] e^{-j2\pi in/N}. \tag{5.13}$$

We know that the channel noise is i.i.d., and we assume that the distortion noise is m-dependent to decorrelate fast enough, i.e., $m \ll N$. Hence, $W_k[i]$ converges absolutely to a Gaussian random variable by an application of CLT [56], i.e., $W_k[n] \sim \mathcal{CN}(0,\, \sigma^2_{W_k})$ where $\sigma^2_{W_k} = N\sigma^2_{w_k}$.

Assuming that the CSI is available at the PS as proposed in [57], the received signals from the K antennas can be combined to align the gradient vectors as

$$y[i] = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{1-\eta_k} \Bigg(\sum_{m=1}^{M} (H_{mk}[i])^* \Bigg) r_k[i]. \tag{5.14}$$

By substituting (5.11) into (5.14), we obtain

$$y[i] = \underbrace{\frac{1}{K} \sum_{k=1}^{K} \sum_{m=1}^{M} |H_{mk}[i]|^2 g_m[i]}_{\text{signal term}} \tag{5.15a}$$

$$+ \underbrace{\frac{1}{K} \sum_{k=1}^{K} \sum_{m=1}^{M} \sum_{m'=1, m' \neq m}^{M} (H_{mk}[i])^* H_{m'k}[i] g_{m'}[i]}_{\text{interference term}} \tag{5.15b}$$

$$+ \underbrace{\frac{1}{K} \sum_{k=1}^{K} \frac{1}{1-\eta_k} \Bigg(\sum_{m=1}^{M} (H_{mk}[i])^* \Bigg) W_k[i]}_{\text{noise term}}. \tag{5.15c}$$

There are three different terms in (5.15): the signal component, interference and noise. Using the law of large numbers, as the number of antennas at the PS $K \to \infty$, the signal term approaches

$$y_{\text{sig}}[i] = \sigma^2_H \sum_{m=1}^{M} g_m[i]. \tag{5.16}$$

Thus, the PS can recover the i-th entry of the desired signal

$$\frac{1}{M} \sum_{m=1}^{M} g_m[i] = \frac{y_{\text{sig}}[i]}{M\sigma^2_H}. \tag{5.17}$$

To analyze the interference term (5.15b), we define

$$\kappa[i] = \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{m'=1,m'\neq m}^{M}(H_{mk}[i])^{*}H_{m'k}[i], \tag{5.18}$$

where $i \in [N]$. Since $H_{mk}[i]$ and $H_{m'k}[i]$ are independent for $m' \neq m$, the mean and variance of $\kappa[i]$ are calculated as

$$\mathbb{E}\left[\kappa[i]\right] = 0, \tag{5.19a}$$

$$\mathbb{E}\left[|\kappa[i]|^2\right] = \frac{M(M-1)\sigma_H^4}{K}. \tag{5.19b}$$

Accordingly, for fixed gradient values, the interference term (5.15b) has zero mean and its variance scales with M^2/K. Thus, similar to the ideal case (where the receive chains have infinite resolution as considered in [57]), the interference term approaches zero as $K \to \infty$. Therefore, using a sufficiently large number of antennas at the PS diminishes the destructive effects of the interference on the learning process, and the estimate for the gradient vector can be determined by

$$\frac{1}{M}\sum_{m=1}^{M} g_m[i] = \begin{cases} \frac{y^R[i]}{M\sigma_H^2}, & \text{if } 1 \leq i \leq e, \\ \frac{y^I[i-e]}{M\sigma_H^2}, & \text{if } e < \ i \leq 2e, \end{cases} \tag{5.20}$$

for $i \in [d]$ from the noisy version of the received local gradients. This result clearly shows that the convergence of the learning process is guaranteed even if we employ low cost low-resolution ADCs at the receiver.

5.2.1 Distortion Factor (η_k) and Noise Variance Calculations for b-bit ADCs

Focusing on the real part of $Y_k[n]$ defined in (5.5), we have

$$Y_k^R[n] = \Bigg[\sum_{m=1}^{M}\sum_{l=1}^{L}\Big(h_{mkl}^R G_m^R[n-\tau_{mkl}] \tag{5.21a}$$

$$- h_{mkl}^I G_m^I[n-\tau_{mkl}]\Big)\Bigg] + z_k^R[n], \tag{5.21b}$$

whose variance is evaluated as follows:

$$\sigma^2_{Y_k^R} = \sum_{m=1}^{M}\sum_{l=1}^{L} |h_{mkl}|^2\sigma_G^2 \tag{5.22a}$$

$$+ \Bigg[\sum_{m=1}^{M}\sum_{l=1}^{L} \sum_{\substack{m'=1 \\ \forall m',l' \setminus \{m'=m,l'=l\}}}^{M}\sum_{l'=1}^{L} \big(h^R_{mkl}h^R_{m'kl'} \tag{5.22b}$$

$$+ h^I_{mkl}h^I_{m'kl'}\big)\cdot R_G(mkl,m'kl')\Bigg] + \sigma^2_{z^R}. \tag{5.22c}$$

Thus, $Y_k^R[n] \sim \mathcal{N}(0,\, \sigma^2_{Y_k^R})$.

For b-bit low-resolution ADCs, the distortion factor for the real part of the ADC input can be calculated by dividing the MSE by the signal power as:

$$\eta_{k,\text{real}} = \frac{\mathbb{E}\left[\big(Q(Y_k^R[n]) - Y_k^R[n]\big)^2\right]}{\mathbb{E}\left[\big(Y_k^R[n]\big)^2\right]}. \tag{5.23}$$

The MSE is obtained as

$$\mathbb{E}\big[\big(Q(Y_k^R[n]) - Y_k^R[n]\big)^2\big] \tag{5.24a}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2_{Y_k^R}}}\Bigg(\sum_{i=1}^{\beta}\int_{\hat{x}_i}^{\hat{x}_{i+1}} (r-\hat{a}_i)^2 e^{-\frac{r^2}{2\sigma^2_{Y_k^R}}}\, dr\Bigg), \tag{5.24b}$$

and $\mathbb{E}\left[\big(Y_k^R[n]\big)^2\right] = \sigma^2_{Y_k^R}$ which gives

$$\eta_{k,\text{real}} = \frac{\frac{1}{\sqrt{2\pi\sigma^2_{Y_k^R}}}\Big(\sum_{i=1}^{\beta}\int_{\hat{x}_i}^{\hat{x}_{i+1}} (r-\hat{a}_i)^2 e^{-\frac{r^2}{2\sigma^2_{Y_k^R}}}\, dr\Big)}{\sigma^2_{Y_k^R}}. \tag{5.25}$$

Similar to [58], the variance of the real part of the noise term $w_k[n]$ which is denoted by $\sigma^2_{w_k^R}$ is calculated as

$$\sigma^2_{w_k^R} = (1-\eta_{k,\text{real}})^2\sigma^2_{z^R} + \eta_{k,\text{real}}(1-\eta_{k,\text{real}})\sigma^2_{Y_k^R}. \tag{5.26}$$

It is assumed that the total effective noise $w_k[n]$ is a non-Gaussian m-dependent random variable whose variance is $\sigma^2_{w_k} = 2\sigma^2_{w_k^R}$. The PS takes N-point DFT of

the ADC output. Hence, $W_k[i] = \sum_{n=0}^{N} w_k[n] e^{-j\frac{2\pi}{N}in}$, which converges absolutely to a Gaussian random variable by the CLT [56] with zero mean and variance is calculated as

$$\sigma^2_{W_k} = \sum_{n=0}^{N-1} \sigma^2_{w_k} = N\sigma^2_{w_k}. \tag{5.27}$$

5.3 One-bit ADCs

When large number of receive chains are used at the PS, it is one of the most important bottlenecks in terms of hardware cost and power consumption. Hence, using one-bit ADCs significantly reduces the cost for practical energy efficient PS system design. With this motivation, we now specialize the results of the previous section to this case.

A one-bit ADC is used to map non-negative inputs to a positive reconstruction level a, and the negative inputs to $-a$. The corresponding quantization function $Q_1(\cdot)$ is simply

$$Q_1(x) = \begin{cases} a, & \text{if } x \geq 0 \\ -a, & \text{otherwise.} \end{cases} \tag{5.28}$$

The mean squared error (MSE) of the quantized signal is calculated as

$$\mathbb{E}\big[|Q(Y_k^R[n]) - Y_k^R[n]|^2\big] = \frac{1}{\sqrt{2\pi\sigma^2_{Y_k^R}}} \left(\int_{-\infty}^{\infty} |Q(r) - r|^2 e^{-\frac{r^2}{2\sigma^2_{Y_k^R}}} dr \right) \tag{5.29a}$$

$$= a^2 + \sigma^2_{Y_k^R} - \frac{4a\sigma_{Y_k^R}}{\sqrt{2\pi}}. \tag{5.29b}$$

The distortion factor for the real part of the ADC input can be calculated by

dividing the MSE by the signal power as:

$$\eta_{k,\text{real}} = \frac{\mathbb{E}\left[\left(Q(Y_k^R[n]) - Y_k^R[n]\right)^2\right]}{\mathbb{E}\left[\left(Y_k^R[n]\right)^2\right]} \tag{5.30a}$$

$$= \frac{a^2 + \sigma_{Y_k^R}^2 - \frac{4a\sigma_{Y_k^R}}{\sqrt{2\pi}}}{\sigma_{Y_k^R}^2}. \tag{5.30b}$$

Thus, the PS can use the same approach to combine the received signal from different antennas to align them by simply changing the η_k value in (5.14) with (5.30).

Remark 1 *Since the real and imaginary parts of (5.5) have same statistics and independent of each other, the calculations for these two cases will be the same. Hence,* $\eta_k = \eta_{k,imag} = \eta_{k,real}$.

Remark 2 *Note that (5.29c) is a convex function of* a*, hence* $\frac{d\left(\mathbb{E}\left[\left(Q(Y_k^R[n]) - Y_k^R[n]\right)^2\right]\right)}{da} = 0$ *whose solution* $a = \frac{2\sigma_{Y_k^R}}{\sqrt{2\pi}}$ *results in the minimum distortion.*

5.4 Numerical Examples

In this section, we evaluate the performance of the distributed learning algorithms at the wireless edge with realistic channel effects and hardware limitations. We use the MNIST dataset [59] with 60000 training and 10000 test samples to train a single layer neural network using the ADAM optimization algorithm. We perform the training for $T = 400$ iterations. At the beginning of the training process, each worker caches $B = 1000$ training samples randomly. The number of parameters is $d = 7850$.

Our system consists of $M = 20$ workers connected to a PS through a multipath fading channel with $L = 3$ taps and $\sigma_{h,l}^2 = 1/L$, hence we have a normalized

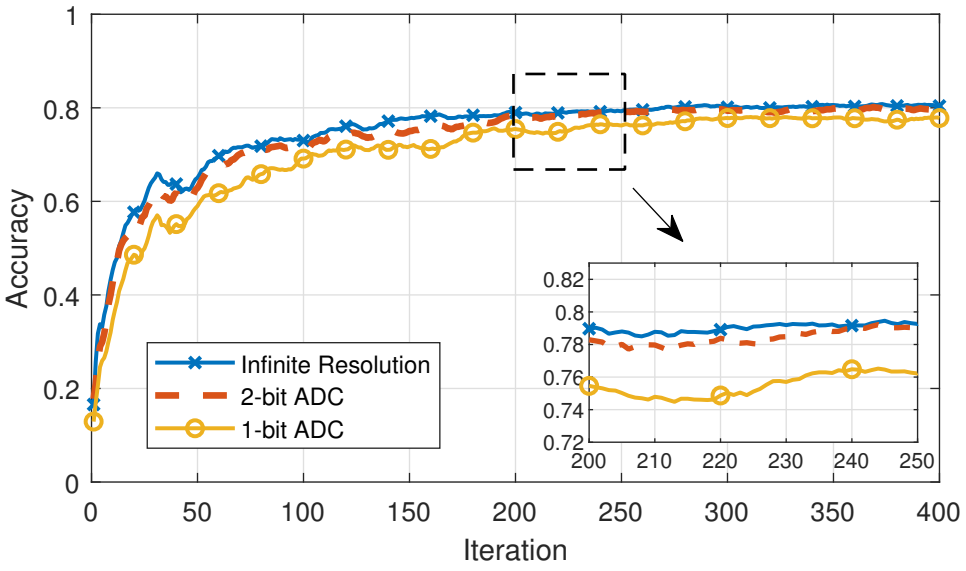

Figure 5.4: Test accuracy of the system with $K = 5$, $\sigma_z^2 = 4 \times 10^{-3}$ for the cases 1) infinite resolution, 2) two-bit ADC, 3) one-bit ADC.

uniform multipath delay profile where each tap experiences Rayleigh fading. The number of subcarriers is taken as $N = 4096$. We assume that first tab has no delay and coherence time corresponds to 1000 indexes of the OFDM word. Also, time delays are uniformly spaced, i.e., $\tau_{mk1} = 1$, $\tau_{mk2} = 501$, $\tau_{mk3} = 1001$ for $\forall m, k$. The cyclic prefix length is set to $N_{\text{cp}} = 1024$, which is enough to remove the ISI effects caused by the multipath. The average transmit power of the OFDM word transmitted by the m-th worker is calculated as $P_T = \frac{1}{T}\sum_{t=1}^{T} \left|\left|\bar{\mathbf{G}}_m^t\right|\right|_2^2$, which gives $P_T = 1.3267 \times 10^{-4}$ for this setup.

In Fig. 5.4, we compare the test accuracy for the above setup by fixing the number of PS antennas as $K = 5$, and the channel noise variance as $\sigma_z^2 = 4 \times 10^{-3}$ for different ADC resolutions. We observe that the system with two-bit ADCs experiences almost no degradation while there is a slight accuracy loss when one-bit ADCs are used compared to the ideal case. Clearly, the DSGD algorithm with OFDM transmission is highly robust against the distortion caused by low-resolution ADCs at the receiver side.

In Fig. 5.5 and 5.6, the test accuracy for different number of antennas $K \in \{1,\ 5,\ M,\ 2M^2\}$ each equipped with one-bit ADC is illustrated for a system with

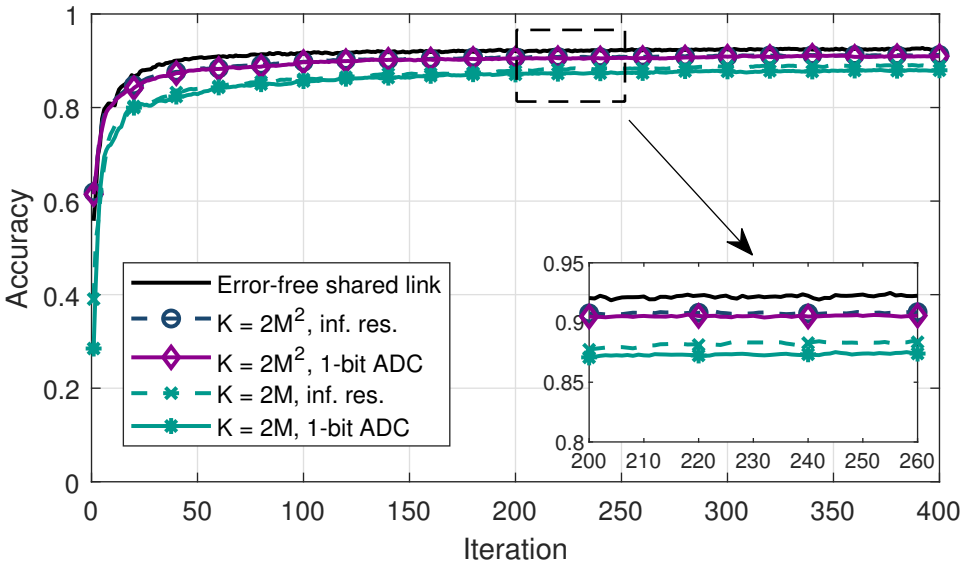

Figure 5.5: Test accuracy of the system with infinite resolution and one-bit ADC with channel noise variance $\sigma_z^2 = 8 \times 10^{-4}$, and $K = 2M, 2M^2$.

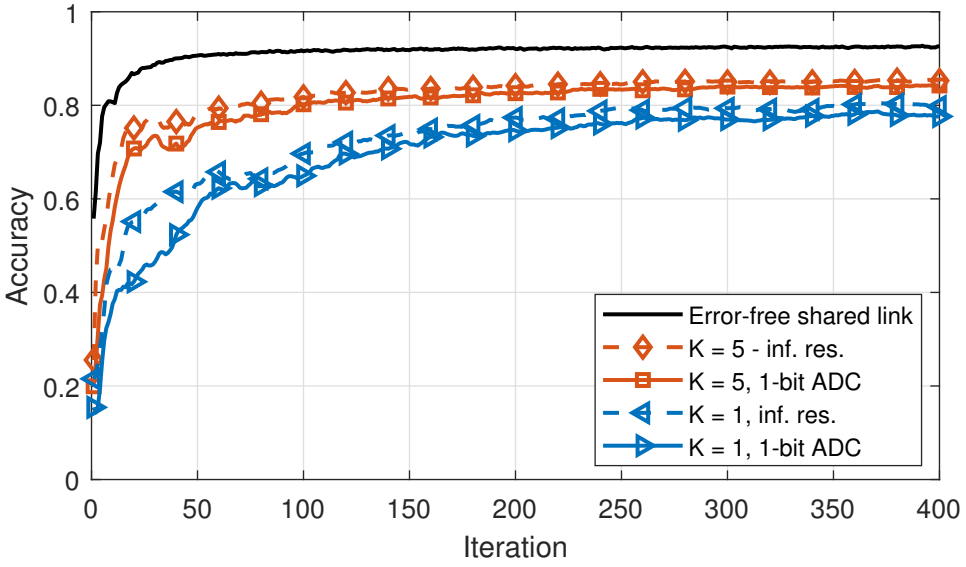

Figure 5.6: Test accuracy of the system with infinite resolution and one-bit ADC with channel noise variance $\sigma_z^2 = 8 \times 10^{-4}$, and $K = 1, 5$.

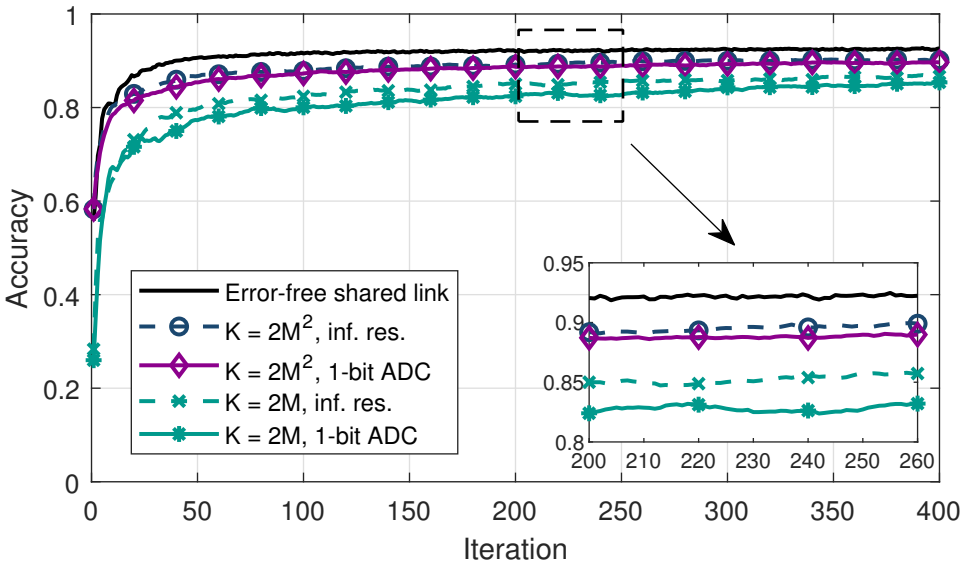

Figure 5.7: Test accuracy of the system with infinite resolution and one-bit ADC with channel noise variance $\sigma_z^2 = 4 \times 10^{-3}$, and $K = 2M, 2M^2$.

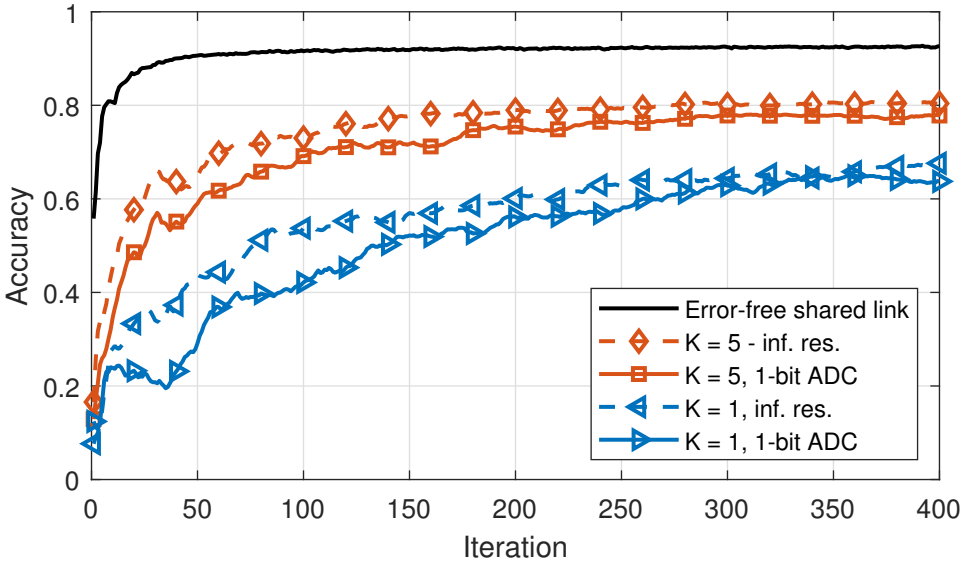

Figure 5.8: Test accuracy of the system with infinite resolution and one-bit ADC with channel noise variance $\sigma_z^2 = 4 \times 10^{-3}$, and $K = 1, 5$.

$\sigma_z^2 = 8 \times 10^{-4}$, and compared with the error-free shared link case. As expected, using higher number of receive antennas results in an improved learning accuracy. Indeed the results are very close to those of the case of error-free shared link. For instance, after the 10-th iteration, using one-bit ADCs causes only 4.69%, 2.44%, 0.89%, and 0.28% accuracy loss on average compared to infinite resolution case for $K = 1$, $K = 5$, $K = 2M$, and $K = 2M^2$, respectively. These excellent results are due to the fact that increasing the number of antennas reduces the interference dramatically which makes the combined signal a very good estimate of the gradient vector, even with low-resolution ADCs.

Without changing any other parameters of the setup described above, we increase the noise variance to $\sigma_z^2 = 4 \times 10^{-3}$ in Fig. 5.7 and 5.8. As in the previous case, for both infinite resolution and one-bit ADC case, the performance of the proposed scheme is very close to the error-free case for large number of receive antennas. When the number of antennas is decreased, with the detrimental effects of the channel noise and interference caused by shared multipath fading channels, the accuracy decreases. However, even for this high level of channel noise, using one-bit ADCs causes only 10.65%, 6.28%, 2.69%, and 0.91% accuracy loss on average compared to infinite resolution case for $K = 1$, $K = 5$, $K = 2M$, and $K = 2M^2$, respectively. Hence, with a slight sacrifice on the accuracy rate of the learning algorithm, power and hardware efficient systems can be designed and implemented for distributed learning at the wireless edge for realistic channel scenarios.

5.5 Chapter Summary

In this chapter, we investigate a distributed learning system at the wireless edge with OFDM based transmission and low-resolution ADCs at the receiver side for practical and inexpensive PS design. Our analytical results illustrate that even with the use of one-bit ADCs at the PS, the convergence of the learning algorithm is guaranteed when the number of receive antennas goes to infinity as in the ideal case of infinite resolution ADCs. Through extensive numerical examples, it is also

observed that using a moderate number of antennas, e.g., using 5 PS antennas, significantly improves the accuracy of the learning algorithm. It is also observed that, in case of low channel noise, the learning performance is decreased only slightly even with one-bit ADCs while the system with two-bit ADCs achieves almost the same accuracy as the ideal case.

Chapter 6

Conclusions and Future Work

In the first part of the thesis, we study the coded caching over packet erasure channels where each user in the system encounters independent packet erasures. We first construct a baseline scheme called sending the same message (SSM) algorithm where the erased submessages are retransmitted until they are received successfully by all the targeted users. Observing that the transmission rate of the SSM algorithm increases significantly beyond a certain erasure probability due to the ignored additional multicasting opportunities among the erased subfiles, we propose a greedy coded caching algorithm which exploits new multicasting opportunities and outperforms the SSM algorithm. Then, we propose a grouped greedy coded caching algorithm, which reduces the complexity of the greedy one with a slight increase in the transmission rate.

We extend our study on coded caching to the case of non-ergodic fading channels where (conditioned on the channel gains) the multicast capacity of the broadcast channel is restricted by the user with the worst SNR. To overcome this limitation, we propose an optimization problem to minimize the transmission time by grouping the users based on their channel conditions, and transmitting the coded messages according to the worst user in the corresponding group, as opposed to the worst user among all. We consider solutions obtained by channel statistics as opposed to their instantaneous SNR knowledge. We derive a locally optimal

iterative solution to find signal to noise ratio thresholds for user grouping. We also develop a simplified approach through a corresponding shortest path problem enabling a numerically efficient solution. We demonstrate that the proposed user grouping is particularly advantageous when the cache sizes are small.

Finally, we study distributed machine learning at the wireless edge with low-resolution analog to digital converters at the receive chains. We assume that the workers independently perform their computation and send the gradient estimates to the PS through a multipath fading MAC via OFDM. At the receiver side, the PS employs multiple antennas to eliminate the fading effect caused by the lack of CSI at the transmitters and uses low-resolution ADCs to reduce the hardware cost and power consumption. We show that the undesired interference term due to the lack of CSI and impairments caused by low-resolution ADCs do not prevent the convergence of the learning algorithm.

As further work, there are several directions to be followed for a complete understanding of distributed caching and distributed learning over wireless channels, especially, considering practical limitations and impairments.

One potential direction for coded caching is to study delay-sensitive contents as considered in [8] where each user has a delay constraint, and the coded transmissions are performed by obeying these constraints. Extending the approach in [8] to the case considering the particular channel characteristics would be interesting.

Another interesting study on coded caching is its use over wireless networks where there is more than one main server which may store the same files in their libraries. Hence, an optimization problem can be formulated to determine the best user assignment protocol based on the quality of the links and the server contents. Also, similar to our study with user grouping over wireless channels, joint user grouping and user assignment algorithms can be developed.

Furthermore, in video streaming servers, the quality of the bits of the files can be separated as the bits which carry the coarse/essential information and the ones which carry the fine information about the file. Hence, the users may

define distortion constraints for the reconstruction of the files where the coarse information is always required while the fine information is only partially required resulting in a lossy reconstruction. In [60], a cut-set bound on the achievable rates with heterogenous distortion factors is derived by considering outer bounds for the capacity region of the system. Together with this cut-set bound, for $K = 2$, the authors also derive a tight lower bound on the transmission rate for specific scenarios. However, for the general case, it is shown that the gap between the existing schemes and the cut-set bound is still significant when the number of users is more than two. Hence, an exciting line of research is to study practical and feasible coded caching schemes which are closer to the cut-set bound compared to the existing schemes when the number of users is more than two.

For distributed learning, an interesting line of work is to consider the case where there is mobility in the system due to relative motion between the workers and the PS, which results in inter-carrier interference (ICI) for OFDM systems by destroying the orthogonality of the subcarriers. Hence, the ICI in distributed learning systems can be studied in terms of its effects on the learning accuracy and convergence of the algorithms.

In addition, one can also study a federated learning system where the workers employ low-resolution digital to analog converters (DACs) at their transmit antennas to reduce the transmit power and hardware costs. Employing DAC at the workers leads to nonlinear distortions in the transmitted gradients, which could result in errors in the global model parameter update. Hence, the effects of this additional source of impairments on the convergence of learning algorithms and on the learning accuracy should be examined.

Finally, in our studies, we use a single layer network during the training process. However, when the number of hidden layers is higher, the amount of computation and communicational cost becomes higher. Also, the complexity of the computations could vary among different layers of the network. Hence, different network architectures can be considered where the required computations are distributed among the workers optimally by also taking the effects of the wireless channel.

Bibliography

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[2] ——, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 4, pp. 1029–1040, 2015.

[3] Z. Chen, P. Fan, and K. B. Letaief, "Fundamental limits of caching: Improved bounds for users with small buffers," *IET Communications*, vol. 10, no. 17, pp. 2315–2318, 2016.

[4] M. M. Amiri and D. Gündüz, "Fundamental limits of coded caching: Improved delivery rate-cache capacity tradeoff," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 806–815, 2016.

[5] C.-Y. Wang, S. S. Bidokhti, and M. Wigger, "Improved converses and gap results for coded caching," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7051–7062, 2018.

[6] S. M. Asghari, Y. Ouyang, A. Nayyar, and A. S. Avestimehr, "Optimal coded multicast in cache networks with arbitrary content placement," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, May 2018.

[7] H. Ghasemi and A. Ramamoorthy, "Algorithms for asynchronous coded caching," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, Oct. 2017, pp. 636–640.

[8] U. Niesen and M. A. Maddah-Ali, "Coded caching for delay-sensitive content," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, Jun. 2015, pp. 5559–5564.

[9] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 355–370, 2014.

[10] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. M. Prabhakaran, "Private coded caching," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 685–694, 2017.

[11] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 1146–1158, 2016.

[12] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 2, pp. 836–845, 2016.

[13] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3212–3229, 2016.

[14] M. M. Amiri and D. Gündüz, "Cache-aided content delivery over erasure broadcast channels," *IEEE Transactions on Communications*, vol. 66, no. 1, pp. 370–381, 2017.

[15] E. Tuncel, "Slepian-wolf coding over broadcast channels," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1469–1482, 2006.

[16] S. Kamel, M. Sarkiss, and M. Wigger, "Decentralized joint cache-channel coding over erasure broadcast channels," in *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*. IEEE, Apr. 2018.

[17] R. Combes, A. Ghorbel, M. Kobayashi, and S. Yang, "Utility optimal scheduling for coded caching in general topologies," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1692–1705, Aug. 2018.

[18] W. Huang, S. Wang, L. Ding, F. Yang, and W. Zhang, "The performance analysis of coded cache in wireless fading channel," *arXiv preprint arXiv:1504.01452*, 2015.

[19] M. Ji and R.-R. Chen, "Caching and coded multicasting in slow fading environment," in *2017 IEEE Wirel. Commun. and Netw. Conf. (WCNC)*, Dec. 2017.

[20] A. Ghorbel, K.-H. Ngo, R. Combes, M. Kobayashi, and S. Yang, "Opportunistic content delivery in fading broadcast channels," in *GLOBECOM 2017-2017 IEEE Global Commun. Conf.*, Dec. 2017.

[21] K.-H. Ngo, S. Yang, and M. Kobayashi, "Cache-aided content delivery in MIMO channels," in *2016 54th Annual Allerton Conf. on Commun., Control, and Computing (Allerton)*. IEEE, 2016, pp. 93–100.

[22] ——, "Scalable content delivery with coded caching in multi-antenna fading channels," *IEEE Trans. on Wirel. Commun.*, vol. 17, no. 1, pp. 548–562, Jan. 2018.

[23] J. Zhang and P. Elia, "Wireless coded caching: A topological perspective," in *2017 IEEE Int. Symp. on Inf. Theory (ISIT)*, Jun. 2017.

[24] J. Catlett, "Megainduction: machine learning on very large databases," *PhD thesis, Basser Department of Computer Science, University of Sydney*, 1991.

[25] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," in *Advances In Neural Information Processing Systems*, 2016, pp. 2100–2108.

[26] V. Kumar, A. Grama, G. Anshul, and G. Karypis, "Introduction to parallel computing: Design and analysis of algorithms," *Benjamin/Cummings Publishing Company, Redwood City, CA*, vol. 18, pp. 82–109, 1994.

[27] G. C. Fox, S. W. Otto, and A. J. Hey, "Matrix algorithms on a hypercube i: Matrix multiplication," *Parallel computing*, vol. 4, no. 1, pp. 17–31, 1987.

[28] G. Ananthanarayanan, S. Kandula, A. G. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, "Reining in the outliers in map-reduce clusters using mantri." in *Osdi*, vol. 10, no. 1, 2010, p. 24.

[29] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, and I. Stoica, "Improving mapreduce performance in heterogeneous environments." in *Osdi*, vol. 8, no. 4, 2008, p. 7.

[30] K.-H. Huang and J. A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE transactions on computers*, vol. 100, no. 6, pp. 518–528, 1984.

[31] D. Wang, G. Joshi, and G. Wornell, "Efficient task replication for fast response times in parallel computation," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 1. ACM, 2014, pp. 599–600.

[32] ——, "Using straggler replication to reduce latency in large-scale parallel computing," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 3, pp. 7–11, 2015.

[33] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "A unified coding framework for distributed computing with straggling servers," in *2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2016, pp. 1–6.

[34] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2017.

[35] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.

[36] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Project adam: Building an efficient and scalable deep learning training system," in *11th USENIX Symp. on Operating Systems Design and Implementation (OSDI' 14)*, 2014, pp. 571–582.

[37] M. M. Amiri and D. Gunduz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *arXiv preprint arXiv:1901.00844*, 2019.

[38] G. Zhu, Y. Wang, and K. Huang, "Low-latency broadband analog aggregation for federated edge learning," *arXiv preprint arXiv:1812.11494*, 2018.

[39] M. M. Amiri and D. Gündüz, "Over-the-air machine learning at the wireless edge," in *2019 IEEE 20th Int. Workshop on Signal Processing Advances in Wirel. Commun. (SPAWC)*. IEEE, 2019.

[40] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Inf. Processing Systems*, 2017, pp. 1709–1720.

[41] S.-Y. Zhao, H. Gao, and W.-J. Li, "Quantized Epoch-SGD for communication-efficient distributed learning," *arXiv preprint arXiv:1901.03040*, 2019.

[42] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[43] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1175–1191.

[44] S. Abbott, *Understanding Analysis*. Springer, 2001.

[45] L. Turner, "Variants of shortest path problems," *Algorithmic Operations Research*, vol. 6, no. 2, pp. 91–104, 2011.

[46] C. Studer and G. Durisi, "Quantized massive MU-MIMO-OFDM uplink," *IEEE Trans. on Commun.*, vol. 64, no. 6, pp. 2387–2399, Apr. 2016.

[47] C. Mollen, J. Choi, E. G. Larsson, and R. W. Heath, "Uplink performance of wideband massive MIMO with one-bit ADCs," *IEEE Trans. on Wirel. Commun.*, vol. 16, no. 1, pp. 87–100, Oct. 2016.

[48] D. Dardari, "Joint clip and quantization effects characterization in OFDM receivers," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 53, no. 8, pp. 1741–1748, Aug. 2006.

[49] J. Zhang, L. Dai, X. Li, Y. Liu, and L. Hanzo, "On low-resolution ADCs in practical 5G millimeter-wave massive MIMO systems," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 205–211, Apr. 2018.

[50] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive MIMO uplink with low-resolution ADCs," *IEEE Trans. on Wirel. Commun.*, vol. 16, no. 6, pp. 4038–4051, Apr. 2017.

[51] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted Gaussian signals," 1952.

[52] Y. Li, C. Tao, G. Seco-Granados, A. Mezghani, A. L. Swindlehurst, and L. Liu, "Channel estimation and performance analysis of one-bit massive MIMO systems," *IEEE Trans. on Signal Processing*, vol. 65, no. 15, pp. 4075–4089, May 2017.

[53] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Jour. on Sel. Areas in commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.

[54] H.-S. Lee and C. G. Sodini, "Analog-to-digital converters: Digitizing the analog world," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 323–334, Feb. 2008.

[55] S. Wei, D. L. Goeckel, and P. A. Kelly, "Convergence of the complex envelope of bandlimited OFDM signals," *IEEE Trans. on Inf. Theory*, vol. 56, no. 10, pp. 4893–4904, Sep. 2010.

[56] P. Billingsley, *Probability and measure*. John Wiley & Sons, 2008.

[57] M. M. Amiri, T. M. Duman, and D. Gunduz, "Collaborative machine learning at the wireless edge with blind transmitters," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov. 2019.

[58] A. Mezghani and J. A. Nossek, "Capacity lower bound of MIMO channels with output quantization and correlated noise," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2012.

[59] Y. LeCun, "The MNIST database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[60] Q. Yang and D. Gündüz, "Coded caching and content delivery with heterogeneous distortion requirements," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4347–4364, 2018.