FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING
DEGREE PROGRAMME IN ELECTRONICS AND COMMUNICATIONS ENGINEERING

MASTER'S THESIS

REDUCED COMPLEXITY MULTICAST BEAMFORMING AND GROUP ASSIGNMENT SCHEMES FOR MULTI-ANTENNA CODED CACHING

Author	Shanuka Ranjitha Gamaethige
Supervisor	Prof. Antti Tölli
Supervisor	Dr. Himal A. Suraweera
Second examiner	Dr. Mohammadjavad Salehi
Technical supervisor	Hamidreza Bakhshzad Mahmoodi

July 2022

Shanuka Ranjitha Gamaethige. (2022) Reduced Complexity Multicast Beamforming and Group Assignment Schemes for Multi-antenna Coded Caching. Faculty of Information Technology and Electrical Engineering, Degree Programme in Electronics and Communications Engineering, 60 pages.

ABSTRACT

In spite of recent advancements in wireless communication technologies and data delivery networks, it is unlikely that the speeds supported by these networks will be able to keep up with the exponentially increasing demand caused by the widespread adoption of high-speed and large-data applications. One appealing idea proposed to address this issue is coded caching, which is an innovative data delivery technique that makes use of the network's aggregate cache rather than the individual memory available to each user. This proposed idea of coded caching helps boost the data rates by distributing cache material throughout the network and delivering independent content to many users at a time. Despite the initial theoretical predictions of considerable caching gains, coded caching actually experiences severe bottlenecks that significantly reduce these gains. Some of these bottlenecks are requiring complex successive interference cancellation (SIC) at the receiver, exponential increase in subpacketization, applicability to a limited range of input parameters, and performance losses in low- and mid- signal-to-noise ratio regimes. In this study, we present a novel coded caching scheme based on user grouping for cache-aided multi-input single-output networks. One special property of this new scheme is its applicability to every set of input values for the user count (K), transmitter-side antenna count (L), and the global coded caching gain (t). Moreover, for a fixed t, this scheme can achieve theoretical sum-DoF optimality with no limitations. This strategy yields superior performance in terms of subpacketization when input parameters satisfy $\frac{t+L}{t+1} \in \mathbb{N}$. This performance boost is enabled by the underlying user grouping structure during data delivery. However, when input parameters do not comply with $\frac{t+L}{t+1} \in \mathbb{N}$, in order to guarantee symmetry of the scheme and optimal degrees-of-freedom, multicast and unicast messages need to be constructed using a tree diagram, resulting in excess subpacketization and transmission count. Nevertheless, the simple receiver structure without the SIC requirement not only simplifies the implementation complexity but also enables us to use state-of-the-art methods to readily design optimized transmit beamformers maximizing the achievable symmetric rate. Finally, we use numerical analysis to compare our new proposed scheme with well-known coded caching schemes in the literature.

Keywords: Coded caching, Multi-antenna communications, Multicast, Beamforming, Subpacketization, Degrees-of-freedom.

CONTENTS

ABSTRACT
CONTENTS
PREFACE
LIST OF SYMBOLS AND ABBREVIATIONS
LIST OF SYMBOLS AND ABBREVIATIONS
1 INTRODUCTION 7
1.1 Cache-Aided Communication 7
1.2 Coded Caching 7
1.3 Research Problem 9
1.4 Thesis Outline and Contribution 10
1.5 Organization of the Thesis 10
1.6 Notation 11
2 LITERATURE REVIEW 12
2.1 Background and Review of Prior Work 12
2.1.1 Traditional Caching 12
2.1.2 Coded Caching 13
2.1.2.1 Multi-Antenna Coded Caching 14
2.1.3 Coded Caching Schemes in Finite SNR Regime 14
2.1.3.1 Beamforming in Coded Caching 14
2.1.4 Subpacketization Bottleneck 15
2.1.4.1 Signal-Level Coded Caching 17
2.1.4.2 Cyclic Caching 17
3 OVERVIEW OF CACHE PLACEMENT AND DELIVERY 19
3.1 Review of Multi-antenna Coded Caching in [1] 19
3.2 Review of Multi-Antenna Interference Management for Coded Caching in [2]. . 21
3.2.1 Complexity Reduction in beamformer optimization problem 22
3.3 Review of Low-Complexity High-Performance Cyclic Caching for Large MISO Systems in [3]. 24
4 PROPOSED NOVEL GROUP ASSIGNMENT CODED CACHING SCHEME . . . 26
4.1 System Model 26
4.2 The Proposed Scheme 27
4.2.1 Scenario 1 : $N_m \geq 1$ and $N_{mod} = 0$ 29
4.2.1.1 Example 1 29
4.2.1.2 Example 2 30
4.2.2 Scenario 2 : Special Case $N_m = 1$ and $N_{mod} > 0$ 31
4.2.3 Scenario 3: Most General Case $N_m \geq 1$ and $N_{mod} > 0$ 32
4.2.4 Scenario 4: $K > t + L$ 34
5 GENERAL CASE FORMULATION AND ALGORITHM 35
5.1 General Case Formulation 35
5.1.1 Algorithm for Building the Tree Diagrams 38
5.1.2 Building Transmission Vectors from the Tree Diagram 39
5.1.3 Number of Transmissions 40
5.1.4 General Subpacketization 41
5.2 Beamformer Design Suggestions 41

6 PERFORMANCE OF THE PROPOSED SCHEME 43
6.1 Performance comparison for the restricted parameter case 43
6.1.1 Numerical example comparing three schemes 44
6.2 Comparison of performance without parameter restrictions 44
6.2.1 Numerical Example Comparing Three Schemes, $N_{mod} > 0$ 45
7 CONCLUSIONS AND FUTURE WORK 47
7.1 Summary and Conclusions 47
7.2 Future Work 48
8 BIBLIOGRAPHY 49
8.1 Appendix 1 55
8.1.1 Circulant Matrices 55
8.1.1.1 Symmetric Circulant Matrices 55
8.2 Appendix 2 57
8.3 Appendix 3 59
8.3.1 Proof of the Subpacketization Given in (5.8) 59
8.3.1.1 Inner and Outer Loops 59
8.3.1.2 First Branch of the Tree Diagram 59
8.3.1.3 Second and Every Other Preceding Branch of the Tree Diagram 59

PREFACE

First and foremost, I would like to express my gratitude to Prof. Antti Tölli, Dr. Himal A. Suraweera, Dr. Mohammadjavad Salehi and Hamidreza Bakhshzad Mahmoodi, my supervisors, for their full support and guidance. I want to express my gratitude to them in particular for helping in defining my thesis topic, their innovative ideas, their technical assistance, and their excellent suggestions on scientific writing which was instrumental to shape my research in the correct direction. During the course of this thesis work, I feel as though I have entered a new realm of research and Mohammadjavad and Hamidreza are to thank for this discovery. I also want to sincerely thank you both for guiding me on the correct path. I was fortunate to have their support when I needed it during the inevitable moments of uncertainty and hesitancy that come with finishing a master's thesis. Hamidreza, Dr. Mohammadajavad and Dr. Himal deserves special thanks for their reviews and comments during the writing stage of the thesis. Special thanks goes to Hamidreza. I would not have been able to complete the research without his assistance and kind help.

LIST OF SYMBOLS AND ABBREVIATIONS

Acronyms

BC	Broadcast Channel
CSI	Channel State Information
DoF	Degrees-of-Freedom
MAC	Multiple Access Channel
MAPDA	Multiple-antenna Placement Delivery Array
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MISO-BS	Multiple Input Single Output - Broadcast Channel
MTS	Multicast Transmission Symbols
PDA	Placement Delivery Array
SIC	Successive Interference Cancellation
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
UTS	Unicast Transmission Symbols
ZF	Zero Forcing

Symbols

α	Size of the user subset served during a given time interval
β	Number of parallel messages the receiver should be able to distinguish
C	Circulant matrix
c_i	Any element in circulant matrix
$\tilde{C}^{t+1}_{[K]-g({}^{k}M^{l}_{x})}$	A matrix constructed using first $(t+1)$ columns of another C
d_K	Kth element of demand vector
$\gamma_C(i)$	Common symmetric SINR for all users served in time slot i
Λ	Parameter which is same as β
λ_j	Eigenvalues of any real symmetric circulant matrix
ϕ_L	Greatest common divisor of K
τ	Subset of users
$\mathcal{A}$	Remaining indices of $k - i(t+1)$
F	Size of a file
$g({}^{k}M^{l}_{x})$	Union of all the multicast user-sets appeared on the branches before
$\mathbf{h}_k$	Channel gain
K	Number of users
L	Number of antennas
$\mathcal{M}$	All multicast user sets initally stored

LIST OF SYMBOLS AND ABBREVIATIONS

M	Cache size
m_k	Coded messages transmitted simultaneously
$\mathcal{T}[k]$	Represents kth tree
${}^{k}M_{x}^{1}$	An arbitrary multicast user-set in kth tree first phase
${}^{k}M_{y}^{l}$	An arbitrary multicast user-set in kth tree second phase
$\mathcal{N}$	All the sets allocated to a node
N	Library size
N_m	Maximum number of multicast messages
N_{mod}	Number of unicast messages
N_0	Gaussian noise
$\mathcal{P}$	All the rolled sets allocated
$\mathsf{P}(i)$	Possible partitioning of users in to groups
$R_C(i)$	Common symmetric rate for all users served in time slot i
t	Aggregate cache size normalized by the size of the library
U_Y	An arbitrary unicast user index
$\mathcal{W}$	File in the library
$w_{\mathcal{T}}^{\mathcal{S}}$	General multicast beamforming vectors
$W_{n,\mathcal{T}}^{q}$	Subfile stored at the cache memory of all the users k
$\mathbf{x}(\mathcal{S})$	Transmitted all the code words
$X(\mathcal{V})$	Transmitted selected set of code words
$\tilde{X}_{\mathcal{T}}^{\mathcal{S}}(i)$	Multicast message chosen from a unit power complex Gaussian codebook
$\mathcal{Z}_k$	Messages stored in kth cache

1 INTRODUCTION

Online communication has made it possible for information to reach audiences and people who were previously inaccessible.The general population is now more aware of the events taking place around the world. Mobile communication has become one of the largest and most significant platforms in history, revolutionizing the ways in which we interact, enjoy entertainment, and utilize the Internet. The ever-increasing amounts of multimedia material are putting a significant strain on wireless communication networks, which are already under intense demand. The International Telecommunication Union has released a report indicating that mobile data traffic has increased significantly over the past several years, with more rise predicted beyond 2025 [4]. As a response, there has been vast developments in the 5G technology recently, and it is anticipated that by 2025, it would account for one-fourth of all mobile connections [5]. While 5G is still in its infancy and that the full potential of the technology has not yet begun to be realized, the mobile industry is already making preparations for 6G [6].

Despite recent developments in data delivery networks and wireless communication technologies, it is unlikely that the speeds supported by these networks will be able to keep up with the exponentially increasing demand that is being caused by the widespread adoption of high-speed and large-data applications [7]. Conviva, a California-based online video optimization and analytics company, recently found that video start-up times were 11% slower on mobiles compared to wireline internet connections in the United States and 9% slower in Asia due to buffering [8]. While this may not seem like a significant difference, it resulted in 5.1 billion hours of lost viewing time across these regions [8]. Upgrading from 4G to 5G or from WiFi 5 to WiFi 6/6E improves the problem, but not necessarily enough to match escalating quality demands. Moreover, a distinguishing feature of mobile video delivery is their temporal variability which results in much more network traffic during peak hours than during off-peak hours [9].

1.1 Cache-Aided Communication

The use of cache-aided communications allows to take advantage of off-peak hours and move some of the network's traffic away from times of high traffic. The emergence of predictable content has elevated the importance of caching as a key component for managing the exponential global data growth. Cache-assisted communication includes prefetching the most popular items at the network edge during less busy hours to prevent network overcrowding when real user requests are made [10]. Moreover, hardware in the form of memory is significantly less expensive and they are abundantly accessible on mobile devices and small cell base stations.

The caching process transforms memory into bandwidth, and consequently, is of significant interest. Caching provides a variety of advantages, the most notable of these are increased network speed and being able to improve the fairness among different users. As a result, the technology used for content caching will be an essential and integral component of next-generation mobile networks. However, the benefits of traditional caching are restricted to a fraction of the database stored at each individual user, which is in practice generally insignificant [11]. The main inefficiency of traditional caching is that the receiver only uses the cached part of one file that the receiver requests, leaving the rest of the data in the cache unused.

1.2 Coded Caching

Coded caching is a novel approach to the distributed caching problem. The work of Maddah-Ali and Niesen in [12] suggested the notion of coded caching as an efficient way to distribute

multimedia information. This original model introduced in [12], stores multiple files in a server connected to multiple cache aided users over an error free shared link. After all requests have been sent to the server and the local caches have been fully utilized, the server can begin sending coded multicast messages to all users in order to fulfill their requests. This proposed idea of coded caching helped to boost the data rates by distributing cache material throughout the network and delivering independent content to many users at a time. The underlying principle behind coded caching is to store non-identical file chunks in the caches of many users in order to make multicasting opportunities available via coding.

Compared to uncoded placement, coded placement strictly improves performance, and it can be exactly optimal. There are two different kinds of gains. One of them is a local gain because some of the information each user wants is already in that user's cache and doesn't need to be sent again. The second is a global gain, which comes from the fact that caching the data that one user wants gives other users the chance to cancel out interference, leading to broadcasting opportunities. This global caching gain is proportional to the total size of the cache shared by all of the users in the network and has the potential to be significant due to the growth of cache-enabled communication devices. The standard coded caching configuration in [12] has the drawback that before the delivery phase can start, all users present during the placement phase must be active and send their demand synchronously.

The idea of coded caching introduced in [12] can be illustrated by a simple example. In this example, the number of users $(K) = 2$, the library size is $(N) = 2$, and the cache size is $(M) = 1$. Note that this is a normalized size, i.e., We assume that each user has enough cache memory to store one of the library's files. The coded caching scheme in [12] is as follows. First in the placement phase, file A and B are split into two non overlapping subfiles such that $A = [A_1; A_2]$, $B = [B_1; B_2]$. User 1 caches A_1 and B_1, and User 2 caches A_2 and B_2. Next, let's assume that User 1 requests File A while User 2 requests File B during the delivery phase. So now User 1 needs sub-file A_2 and User 2 needs sub-file B_1 from the server. So this request can be served by transmuting A_2 and B_1 over the shared link by the server. This way, the transmission rate happens to be 1 file and this is the un-coded transmission rate. However, with coded caching, as shown in Figure 1.1, the server simply transmits $A_2 \oplus B_1$, where $\oplus$ denotes the bitwise XOR operation. It is clear that User 1 already has A_1, therefore it can recover A_2 from $A_2 \oplus B_1$. Similarly, User 2 already has B_2 and can recover B_1 from from $A_2 \oplus B_1$. As a result, the transmission of $A_2 \oplus B_1$ can guarantee that both requests are recovered, and the transmission rate is half file. This careful cache placement, optimized delivery, and recovering files using cache is the baseline of coded caching. Here it shows coded transmission rate is only half of the un-coded transmission rate.

Shariatpanahi *et al.* investigated the cache-aided communication in multi-antenna settings, and their results showed that the same amount of caching gain could be kept to the fullest extent as in [12]. Afterward, the same concept has been applied to multi-antenna wireless communications in [13]. It is shown in [14] that a network can have $(L + t)$ degrees-of-freedom (DoF), where L is the number of antennas and t is the aggregate cache size normalized by the size of the library. This is a significant improvement compared to the L DoF possible with L antennas with no caching. Despite the fact that there is a current trend toward massive multiple input multiple output (MIMO) [15], [16], which requires the utilization of antenna arrays that contain a large number of components, potential benefits from caching are still substantial. This is especially true when taking into account the high expense of deploying antenna arrays, while, in contrast, the usage of a small cache at each mobile device is incredibly inexpensive and these caches may be rapidly scaled up to a total size capable of delivering significant gains.

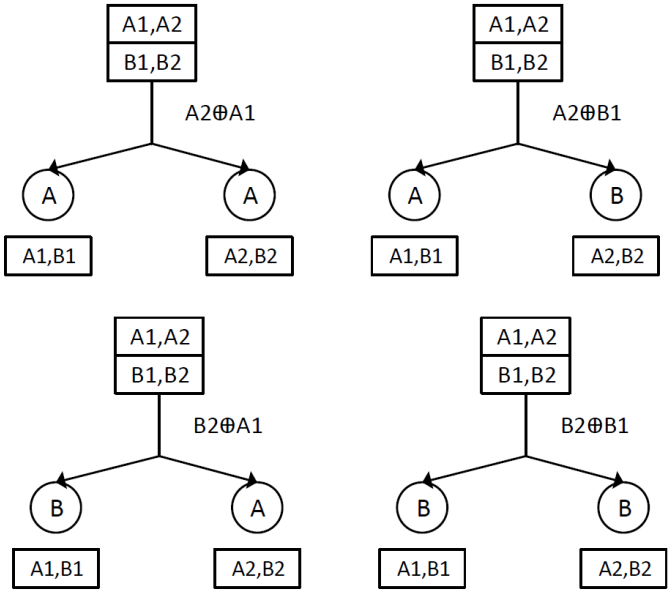


Figure 1.1: Cache placement for $K = 2$, $N = 2$ and cache size $M = 1$. Each file here is split into two sub files and four possible user requests are addressed.

1.3 Research Problem

Using coded caching in wireless networks, especially cellular networks, necessitates addressing a number of practical concerns.Cache allocation should be optimized based on network traffic, the quality of each user's channel, the user's accessible storage, and other network parameters since each user in a practical system has a channel with a unique capacity and statistics. Despite the initial theoretical predictions of significant caching gains, coded caching can in reality experience severe bottlenecks that sharply reduce these gains. Some of these bottlenecks are requiring complex successive interference cancellation (SIC) at the receiver, exponential increase in subpacketization, applicability to a limited range of input parameters, and performance losses in low- and mid- signal-to-noise ratio (SNR) regimes.

As shown in [13], using optimum multigroup multicast beamforming vectors instead of zero forcing (ZF), helps substantially improve the performance, especially at the low-SNR regime. However, doing so necessitates solving complex beamformer optimization problems, requiring significant amounts of computational power. There exists a total number of $(t+L)(2\binom{t+L-1}{t}-1)$ rate constraints in the beamformer optimization problem as shown in the MATLAB code for the multiserver scheme[1] requiring more than 20 minutes to find the achievable rate for just one realization in the CVX-based code, which had input parameters as 12 users, six streams and three cache memory in each user. Furthermore, the time spent to solve this optimization problem keeps getting increased with the increased number of users (K) and antenna count (L). In this work, each user k will have to decode $m_k = \binom{t+L-1}{t}$ different messages, and increasing m_k exponentially makes optimization problem complicated. In this context it is evident that using optimized beamformers is not feasible in large-scale networks.

Most of the contemporary literature available on coded caching are employed with limited input parameters and their scalability is limited. On the contrary, in emerging generations of cellular networks such as 6G, it comes as no surprise that networks with a large number of users and antennas are the best candidates for cache-aided communication. Hence, a novel coded

[1]The MATLAB code for the multiserver scheme presented in [13] is available online from: https://gist.github.com/shanuoulu/4848540b81c0e829c7c9f70284c77b6a.

caching scheme is required to reduce the complexity at both the transmitter and receiver which also works with every practical input parameter.

1.4 Thesis Outline and Contribution

The scope of this thesis only concerns coded caching methods and we limit our review of the literature accordingly.

- We briefly review content placement and content delivery phases presented for a multi-antnenna setting in [1], generic beamformer design for the state-of-the-art coded caching scheme in [2] and cyclic coded caching with reduced subpacketization in [3].
- Performance of [2] is investigated in terms of DoF and SNR using a MATLAB code. It is found out that the main causes of increased complexity of the scheme in [2] are the complex beamformer design and use of a SIC receiver.
- A novel coded caching scheme is developed having the primary goal of reducing the complexity in terms of beamformer optimization problem and receiver complexity in [2]. To accomplish this goal, a strategy is selected to transmit multicast streams in L orthogonal time intervals/slots, instead of transmitting all in parallel.
- In the proposed new scheme initial cache placement is followed as in [12] and using a tree diagram, messages are subdivided in a novel way. Construction of the transmission vectors is done using both tree diagram and [12].
- In this scheme overlap among the multicast messages transmitted in parallel is prevented by partitioning users into groups where each message only serves a group of $t+1$ ($\in (t+L)$) users. This results in a linear receiver implementation, which does not require SIC unlike in [2]. As no overlap is allowed, each user will decode a single multicast message making the beamforming problem simpler.
- This proposed new scheme works with any input parameter (i.e., number of users, cache memory size, number of streams) and achieves $t+L$ DoF with the help of unicast messages.

1.5 Organization of the Thesis

This thesis consists of eight chapters, and it is organized in the following manner: Chapter 2 presents a critical evaluation of the existing literature related to traditional caching, coded caching, subpacketization bottleneck, and energy efficiency in wireless networks. Next, Chapter 3 contains a concise overview of the original scheme of coded caching, multi-antenna coded caching by Shariatpanahi *et al.*, a review of generic beamformer design for state-of-the-art coded caching by Tolli *et al*. and cyclic caching for large multiple input single output (MISO) systems by Salehi *et al*. After that, Chapter 4 investigates the system model, and the proposed scheme is elaborated using four straightforward scenarios. Chapter 5 studies the generalization of the proposed scheme and here we introduce tree diagrams to build transmission vectors. In Chapter 6 performance of the proposed new scheme is discussed. Specifically, the proposed scheme is compared and contrasted critically with [2] and [3]. Chapter 7 presents the conclusion where findings are summarized and future work is described.

1.6 Notation

$\mathbb{N}$ characterizes the set of natural numbers

$[K]$ characterizes the set of numbers from 1 to K as $[K] \equiv \{1, 2, \cdots, K\}$

$\oplus$ denotes a XOR operation

$\binom{n}{k}$ is the binomial coefficient

Sets are denoted by calligraphic letters i.e., $\mathcal{C}$

Boldface capital letters represent matrices i.e., $\mathbf{V}$

$|K|$ represents the number of elements in K

2 LITERATURE REVIEW

2.1 Background and Review of Prior Work

Over the years, numerous services and applications have emerged due to the development of wireless communications. These applications are enabled by advanced technologies such as MIMO, millimeter wave communication, non-orthogonal multiple access and full-duplex communications [17–19]. Multimedia services have gained considerable popularity because of the availability of applications paving the way for higher data rates and seamless connectivity [20]. Despite the use of a variety of techniques to account for the rise in data demand, wireless communication network providers today are struggling to meet the growing demand [21]. To this end, among the numerous proposed solutions, cache-assisted communication is one of the most promising [22].

In computer systems, the word "cache" was invented to describe a memory with extremely quick access but typically limited capacity. A small cache can significantly boost system performance by taking advantage of correlations in memory access patterns.The concept of caching was later applied to the Internet, where popular web pages were replicated on smaller servers (caches) all over the world in order to save network bandwidth and reduce server congestion. The management of these caches grew challenging in the late 1990s with the explosive growth of the Internet traffic.

Although caching in content delivery networks is a well-established technique, these web caching techniques are insufficient for wireless caching because they disregard fundamental wireless network characteristics. A key distinction between web caching and wireless caching is the broadcasting nature of the wireless channel, which allows multiple clients to receive a message sent by a transmitter. Using coding techniques from information theory, this property enables smarter designs of caching networks and promises significant gains in reducing network traffic [23]. In general, caching is utilized for three key goals: lowering network load, load balancing, and lowering delay.

2.1.1 Traditional Caching

Caching was initially implemented to improve system speed by storing frequently accessed data in memories with quick access speeds [24]. Later, the concept of caching popular data in network-distributed memories was investigated in several wireline contexts, such as web caching systems and content delivery networks [25], [26]. In the recent literature, caching has received a lot of attention in wireless networks as a key technique for reducing the traffic load and meeting the requirement for timely service. The cache placement strategy in a caching system describes what data are to be kept in the users' local memories and how they are to be stored in advance. This all happens without knowledge of the users' real requests [27]. Cache placement design has been studied extensively in recent years to better understand how caching affects network load reduction [28].

Several studies have highlighted the importance of caching in different contexts of wireless communications. One such work is [29] where the caching idea is employed for collaborative relaying. Specifically, the authors showed with the help of wireless edge caching, how the efficiency of relaying systems can be improved. Results of the work in [29] demonstrates that substantial outage performance gains over traditional relaying methods without caching can be achieved. Moreover, traditional uncoded caching schemes such as [30] can increase the hit rate and the local caching gain; however, it was found out by [31] that uncoded caching schemes are inefficient when there are multiple caches.

2.1.2 Coded Caching

Coded caching is an innovative data delivery technique that makes use of the network's aggregate cache rather than the individual memory available to each user. The seminal paper by Maddah-Ali and Niesen [12] recommended coded caching as a method for accelerating content delivery by utilizing receiver-side cached content to eliminate interference. The work in [12] which is also the first coded caching scheme, considers a single-stream broadcast channel (BC) where a single-antenna transmitter has access to a library of N files and serves K receivers, each having a cache of size equal to the size of M files. The case study in [12] takes into account two distinct phases: the cache placement phase, which occurs during the off-peak hours, and the content delivery phase, which occurs during the peak hours. The first phase allows library content to be loaded into users' cache memories, and the second phase uses simple cache-aided multicasting to serve $t + 1$ users at once, resulting in $(t + 1)$ DoF.

As previously mentioned, the storage of popular content usually takes place during the off-peak hours to accommodate future user demands during the peak hours. In the cache placement phase of [12], each file W_n should be divided into $\binom{K}{t}$ subfiles (i.e., $W_n = \{W_{n,\tau}, \tau \subset [K], |\tau| = t\}$). Then, user k caches all subfiles $W_{n,\tau}$ in which $k \in \tau, \ \forall \ n$. In the delivery phase, it can be easily seen that for each $(t + 1)$ subset of users $\mathcal{T}$, a common coded multicast message $\oplus_{k\in\mathcal{T}} W_{d_k,\mathcal{T}\setminus k}$ would benefit all the users in the subset $\mathcal{T}$ with providing them one subfile of their requested file. If all such multicast coded messages are properly transmitted, then it can be demonstrated that all the users will receiver the missing subfiles. The key to achieving these significant gains is the multicasting of carefully designed codewords to different groups of users, so that each user can use its cache content to filter out unwanted parts of the received signal. Hence, in [12], total normalized delay will be $\binom{K}{t+1}/\binom{K}{t} = (K - t)/(t + 1)$.

The main objective of coded caching is to jointly optimize the cache placement phase and the delivery phase to reduce the overall communication load needed to deliver the user files that have been requested [32]. In wired networks, it is common practice to obtain the so-called normalized communication load as a performance metric by dividing the communication load by the file size [33]. In wireless environments, the DoF or the normalized delivery time are used as metrics instead [34]. According to [35], this is because these measures relate the performance to that of a wireless point-to-point channel with a power constraint P. In order to comprehend the fundamental limitation of coded caching, information-theoretic converse bounds were developed. A lower bound was first developed in [12] and the bound is generally loose due to the number of file groups depending on popularity distribution. After that, several works have developed and improved the lower limits for the peak and average rates for files with uniform popularity [36, 37]. By classifying the most popular files using a different method, the paper [38] was able to derive a lower bound for any file popularity distribution.

After that, Maddah-Ali and Niesen [39] came up with a decentralized coded caching scheme based on independent random content placement in a single transmitter set-up. It was shown that, for large networks, the gain from multicasting is almost the same as the gain from centralized caching when files are large. In [40], the finite file size regime was looked at. Also, in [41], the authors suggested a caching method that works better than the method in [39] when the file size is not too large. In parallel to the decentralized coded caching literature, techniques for jointly designing placement and delivery were investigated in other types of networks, such as device to device networks [42], combination networks [43], and networks with shared cache [44] have been studied. The case of networks with shared caches is especially intriguing because, in these types of networks, users can share caches, which, from a practical standpoint, aids in memory efficiency [45]. Also extensively studied is the use of coded caching in networks with non-uniform demand, or networks with files of varying popularity.

2.1.2.1 Multi-Antenna Coded Caching

Extension of [12] to multiserver scenario was studied in [14], and this work was extended to wireless networks with multiple antennas at transmitters and receivers in [1, 46, 47]. The initial deployment of coded caching techniques in multi-antenna wireless networks stressed the DoF configuration in the high SNR and, as a result, promoted the usage of ZF precoders [1, 46, 48]. According to Shariatpanahi *et al.*, caching at end-user locations and multicasting coded file chunks to specific user groups allow for a reduction in delivery bandwidth over the BC in a wireless MISO-BC model.

Decoding is accomplished by utilizing each user's locally cached content to eliminate t unwanted messages from the received signal. A most important scenario is the multi-antenna/multi-transmitter case, for which the scheme in the aforementioned work in [14] achieves a sum-DoF of $(t+L)$, where L is the antenna count. Later, this breakthrough sum DoF derived in the work [49] found to be optimal within a factor of two among all linear one shot schemes. Hence, servers can collaboratively send coded messages when L servers have access to the library instead of a single server. Then, each transmission would benefit $(t + L)$ users, instead of just benefiting $t + 1$ users [14]. regardless, each user k will decode $\binom{t+L-1}{t}$ varying messages, which increases exponentially when K, L, N are increased with identical ratio (linearly if $t = KM/N = 1$).

2.1.3 Coded Caching Schemes in Finite SNR Regime

The original coded caching scheme and analogous work at the time only focused on high SNR regime but the focus was subsequently shifted to lower SNR regime with the publication of [2, 3, 49]. This type of finite SNR analysis provided some important design guidelines that were not disclosed in earlier DoF analyses of wireless coded caching papers in the high SNR regime. Additionally, while the paper [50] takes into account a finite SNR configuration, the scheme used in the paper [1] is designed for a massive MIMO scenario with more transmit antennas than users. Multicasting opportunities provided by coded caching are unnecessary in this case [50] because the transmitter has access to complete CSI.

2.1.3.1 Beamforming in Coded Caching

Beamforming is an important topic of interest for cache-enabled networks. The joint design of data assignment and beamforming for a cooperative multicell network was taken into consideration in [51], while instantaneous beamforming and base station activation for a cloud-radio access network were addressed in [52]. Both studies assumed a given cache content placement in a short-term time scale. Using a mixed time-scale stochastic optimization scheme, beamforming and cache content placement were jointly optimized in [53]. Additionally, extensive research has been done on performance analysis of cache-enabled wireless networks in published works, such as [54], [55], and [56].

A significant contribution to coded caching, particularly in the low SNR regime, was made by the work in [13]. There, ZF precoders were replaced with optimized beamformers that introduced the ability to control inter-stream interference, as opposed to completely eliminating it. In order to balance the negative effects of noise and inter-stream interference as effectively as possible at low SNR, the general expressions for signal-to-interference-plus-noise ratio (SINR) are handled directly. Due to the fact that the resulting optimization problems are not essentially convex, efficient iterative algorithms are developed utilizing successive convex approximation

of non-convex SINR constraints. Here, they assume the receiver to be a SIC receiver, paving the way for the generalization of multigroup multicast beamformer design having any combination of overlapping user groups.

This helps to improve the performance in the low and mid SNR at the expense of significantly increasing the complexity of beamformer design. This added complexity is a result of interference between parallel multigroup multicast messages. The outcome of this work is a general content delivery scheme that works for any value of the problem parameters, such as the number of users K, the size of the library N, the size of the cache M, and the number of transmit antennas L, as long as $t = KM/N$ is an integer. However, it could be noted that the time required to solve this optimization problem increased as the number of users (K) and antennas (L) increased.

In [2], authors have highlighted performance and complexity of beamformer designs in [13]. Novel optimized precoders were introduced, which could properly control the multiplexing gain and the corresponding multiple access channel size when decoding. Hence now, instead of always serving a group of $(t + L)$ users as in [1, 14], The size of the user subset served for a specific period of time could be managed. Moreover, they could demonstrate that operating at a multiplexing gain smaller than L can reduce the complexity of beamformer design and produce higher beamforming gains, which are critical in the low SNR region. However, this new scheme only works with a specific range of system parameters, and complex SIC is still used as the receiver.

Additionally, controlling the spatial multiplexing gain is discussed in [57], where numerical simulations are performed to determine the optimal multiplexing gain for a variety of network variables, including the coded caching gain. The work in [58] proposes a new strategy that restricts the number of messages received by a user during each time slot in order to reduce complexity while maintaining acceptable speeds. Related works include [1], [59]. The aforementioned schemes can perform well, according to numerical assessments, but due to their extensive subpacketization requirements, this performance is only possible in very small network environments. Meanwhile, in more recent work [60] cache enabled ultra-dense networks with edge nodes having the capability of caching and signal processing tackle the beamformer problem with the weighted sum mean-square error minimization approach.

2.1.4 Subpacketization Bottleneck

Using coded caching in wireless networks, specifically cellular networks, necessitates addressing a number of practical concerns. The well-known subpacketization bottleneck is without a doubt the most damaging issue. The first coded caching scheme [12] requires subpacketization of $\binom{K}{t}$, and decentralized schemes [40], [39] also require exponential subpacketization in order to achieve linear caching gains. Most of the works in the literature on coded caching after the original work require each file to split into a number of subfiles that grows exponentially with the number of receiving users to achieve caching gains. To reduce the subpacketization, various combinatorial subfile assignments have been proposed in the literature afterward. Authors in [61] proposed a combination structure, referred to as the placement delivery array (PDA). In a PDA, the positions of the stars denote which subfiles are cached, and the integers denote which subfiles are jointly encoded into the multicast messages. Original scheme [12] can also be represented as a PDA, referred to as MN PDA. Remarkably, various schemes based on PDA, proposed in [62–64], have lower subpacketization than the MN scheme. However, the achievable performance of these PDA-driven schemes is far below the optimal one achieved by [12]. Authors in [65] prove that the maximum sum-DoF under the multiple-antenna placement delivery array

(MAPDA) construction is $L + \frac{KM}{N}$ and does so with a subpacketization of K when $\frac{KM}{N} + L = K$.

According to [62], there is no single-antenna coded caching scheme that enjoys both linear caching gains and linear subpacketization under basic assumptions. Hence, subpacketization requirements for single-antenna coded caching schemes are fundamental and unavoidable. In order to present a plan that well balances subpacketization with performance, authors in [66] used a theoretical strategy based on hypergraphs. Here, it is demonstrated that there are coded caching schemes that allow for a large number of users to have linear subpacketization level growth. Another interesting piece of research is the one that was published in [67]. This study demonstrates that it is possible to achieve gains that scale linearly with the number of users K while simultaneously requiring a level of subpacketization that also scales almost linearly with K. The multi-access coded caching problem is addressed in cite [68] by applying a linear subpacketization scheme to it. Under nonuniform file popularity, the problem is significantly more complicated, and research is scarce.

Within the context of multi-antenna coded caching schemes, the original work in [1] required an astronomical subpacketization of $\binom{K}{t}\binom{K-t-1}{L-1}$. However, the approach in [69] has demonstrated that these constraints do not pose a fundamental barrier in a multi-antenna environment. In fact, in [69], they exhibited that if $\frac{K}{L}$ and $\frac{t}{L}$ are both integers, the optimal DoF $(t + L)$ is achievable with a subpacketization of $\binom{K/L}{t/L}$ which was founded on the ideas of grouping and cache replication. This is significantly lower than the subpacketization in [12] and [14]. This implies that the addition of multiple antennas can multiply the real DoF by a factor of L for fixed subpacketization constraints and fixed file size.

The core idea behind the work in [69] is the implementation of fundamental user grouping techniques to give groups of users access to the same cache content, followed by the use of a particular precoding strategy to break down the user network into parallel coded caching problems. Although we know from [70] that having shared caches between users in single-antenna setups results in an inevitable loss of DoF, the research presented in [69] has demonstrated that multi-antenna shared-cache setups do not necessarily have to suffer from DoF losses. Obviously, in [69] the reduced subpacketization without DoF loss holds true under the sole assumption that $\frac{K}{L}$ and $\frac{t}{L}$ are integers. If either $\frac{K}{L}$ or $\frac{t}{L}$ is non-integer, the scheme of [69] experiences DoF losses as well as increased subpacketization. It can be noted that the amount of DoF that can be achieved may be significantly different if this requirement is not met.

A new method for flexibly choosing subpacketization is introduced in [71], which demonstrates how subpacketization and performance can be traded off. The results, however, are only applicable to the particular instance of $K = t + L$. In [72], joint reduction of the CSI and subpacketization requirements are taken into account, and it is demonstrated that subpacketization of $L_c\binom{K_c}{t}$ is possible, where $L_c := \frac{L+t}{t+1}$ and $K_c := \frac{K}{L_c}$. A small subset of network parameters can be used with the proposed scheme because both L_c and K_c must be integers. Additionally, a DoF loss of $(1 - \frac{t}{K})$ is also produced.

There exists a further intriguing work in [73], which presents a DoF-optimal strategy that reduces transmission and decoding complexity compared to the optimized beamformer system in [2], resulting in a reduction in transmission and decoding complexity but with a small decrease in performance and exponential subpacketization $\binom{K}{t}$. Most of the existing multi-antenna schemes either have subpacketization requirements that grow exponentially with K, or they don't have DoF optimality when $L > t$. However, the authors in [3] introduced a novel cyclic cache placement scheme that could accomplish the sum DoF $L + \frac{KM}{N}$ with subpacketization that is linear with K. It should be emphasized that this work is carried out while the constraint $L \geq \frac{KM}{N}$ is in force.Finally, Table 2.1 summarizes the limitations and subpacketization of some important coded caching schemes.

Table 2.1: Summary limitations and subpacketization of some coded caching schemes.

Coded Caching Scheme	Limitations in the scheme	Subpacketization
[12]	No limitations	$\binom{K}{t}$
[69]	No limitations	$\binom{K}{t}$
[74]	No limitations	$\binom{K}{t}\frac{t!(K-t-1)!}{(K-t-L)!}$
[14]	No limitations	$\binom{K}{t}\binom{K-t-1}{L-1}$
[69]	$\frac{K}{L}, \frac{t}{L} \in \mathbb{Z}^+$	$\binom{K/L}{t/L}$
[2]	$\frac{t+\alpha}{t+\beta} \in \mathbb{Z}^+$	$\binom{K}{t}\binom{K-t-1}{\alpha-1}\frac{(\alpha-1)!}{(\delta-1)!(\beta-1)!(t+\beta)!^{\delta-1}}$
[3]	$t \leq L$	$\frac{K(t+L)}{(\gcd(K,t,L))^2}$
[73]	$\frac{t+L}{t+1} \in \mathbb{Z}^+$	$\binom{K}{t}$

2.1.4.1 Signal-Level Coded Caching

From another viewpoint, the above-mentioned work in [69] proposed a novel cache-aided interference cancellation approach. This method, which we refer to as the signal-level approach, eliminates interference by making use of the cache contents before the signal is decoded at the receiver. However, in the classical bit-level approach, cache-aided interference elimination is performed after the decoding process has been completed. The signal-level approach has been used to reduce the required amount of subpacketization in a number of other papers found in the existing body of research literature, for e.g., [3]. Most of the research works on signal-level coded caching schemes only address the subpacketization issue. However in [3], it is also shown that there are additional advantages with the signal-level approach, such as enabling simpler optimized beamformer designs.

2.1.4.2 Cyclic Caching

The work in [3] introduced a scheme that is the first in the literature to achieve the optimal DoF with a slowly-increasing subpacketization with the network size, making the scheme salable to be applied in networks with a large set of users. This scheme achieves the optimal sum DoF of $(t + L)$ with a subpacketization requirement of only $S = \frac{K(t+L)}{\phi_L^2}$, with $\phi_L = \gcd(K, L, t)$. Interestingly, the underlying cache-aided interference cancellation mechanism of the proposed scheme not only eliminates the need for SIC receivers but also enables the data delivery to rely entirely on unicasting. In other words, each user receives at most one single message after every transmission, and at the same time, each data part sent during any transmission is intended for one user only. However, due to each and every message being unicast, this scheme happened to lose on multicasting gain. Also, as a design constraint, the spatial DoF α can not be smaller

than the coded caching gain t (i.e., the scheme works only when $t < \alpha$).

The coded caching scheme in [3] also enables the optimized minimum mean square error type beamformer design problem to be easily solvable using uplink-downlink duality. While ZF beamformers completely null out the interference at unwanted users, optimized beamformers enable an interplay between nulling out and controlling the interference, improving the performance considerably at the finite-SNR regime [2]. Together with the low-subpacketization requirement, the simple applicability of optimized beamformers results in a highly efficient, scalable multi-antenna coded caching scheme with an appropriate performance at the finite SNR region. However, the results of this scheme show a significant performance loss in the low and mid SNR regimes.

3 OVERVIEW OF CACHE PLACEMENT AND DELIVERY

This chapter reviews three prominent coded caching schemes that inspired this research. These three schemes form the foundation of our work, which are also considered as benchmarks for comparison purposes with the proposed method described in Chapter 6. First, the original error-free share medium system model in [12] was extended to cache-aided communication in multi-server/antenna settings by Shariatpanahi *et al.*. Later on, Tölli *et al.* designed optimal beamformers to balance the detrimental impact of the inter-stream interference caused by the coded messages (transmitted in paralle) and noise, boosting the performance at low SNR. Then, cyclic caching by MohammadJavad *et al.* provided a novel coded caching scheme, with low complexity in terms of both subpacketization and beamforming design, addressing the numerous flaws of the two preceding works, as discussed below.

3.1 Review of Multi-antenna Coded Caching in [1]

In the following, we will provide a concise overview of the scheme described in [1], which is substantial since it shares many concepts with the new scheme presented in this study. The authors of this work take into account a single cell MISO-BC with a multi-antenna base station that has access to a library. This library contains content that can fulfill requests from mobile devices with a single antenna over a shared wireless medium. These mobile devices are cache-enabled, allowing them to cache pertinent data from library during off-peak hours before the delivery phase begins. The paragraph that follows explains how data is subdivided for cache placement.

Authors in [1] divide each file W_n into $\binom{K}{t}$ equal-sized packets $W_{n,\mathcal{T}}$, where $\mathcal{T} \subset [K]$ and $|\mathcal{T}| = t$ and they further split into $\binom{K-t-1}{L-1}$ equal-sized sub packets prior to the placement step resulting $W^q_{n,\mathcal{T}}$, where each $W^q_{n,\mathcal{T}}$ subfile is stored at the cache memory of all the users $k \in \mathcal{T}$. Hence, in this work subpacketization requirement is $\binom{K}{t}\binom{K-t-1}{L-1}$. In the delivery phase, $(t + L)$ number of users are being served at each time slot. In this regard, a separate vector $\mathbf{x}(\mathcal{S})$ is transmitted, where $\mathcal{S} \subseteq [K]$ and $|\mathcal{S}| = t + L$. To create $\mathbf{x}(\mathcal{S})$, first for each $\mathcal{V} \subseteq \mathcal{S}$ with $|\mathcal{V}| = t + 1$, codeword $X(\mathcal{V})$ is built as

$$X(\mathcal{V}) = \bigoplus_{k \in \mathcal{V}} W^{q(k,\mathcal{V})}_{d_k, \mathcal{T}\setminus\{k\}}(k) \ , \tag{3.1}$$

where $\oplus$ denotes bit-wise XOR, $d_k \in [N]$ and $q(k, \mathcal{V})$ is defined such that it guarantees that each subpacket is transmitted only once, and each user is able to decode its requested file after all transmissions are concluded [14].

The work in [1] considered a wireless network operating in the high SNR regime, expressing the performance in terms of the DoF. However, there are several drawbacks to this strategy. First off, even moderate K values make the scheme impractical because subpacketization becomes increasingly more necessary with K. Secondly, employing ZF results in poor performance, especially in the low SNR regime.

Here, a simple example is provided to illustrate how the ZF precoders of this study operate. Here $K = 3$, $L = 2$, $N = 3$ and $M = 1$ input parameters were selected. These three files selected as $\{A, B, C\}$ and during the cache placement phase, each file is divided into three parts of equal size, and caches are filled as follows

$$Z_1 = \{A_1, B_1, C_1\}, \quad Z_2 = \{A_2, B_2, C_2\}, \quad Z_3 = \{A_3, B_3, C_3\}. \tag{3.2}$$

Suppose that, during the content delivery phase, Users $1-3$ request files $A-C$, respectively. The transmitter will then sequentially transmit the following blocks:

$$X_1 = \frac{1}{\sqrt{6}}(B_1 + A_2)\frac{\mathbf{h}_3^{\perp}}{|\mathbf{h}_3^{\perp}|} + \frac{1}{\sqrt{6}}(B_3 + C_2)\frac{\mathbf{h}_1^{\perp}}{|\mathbf{h}_1^{\perp}|} + \frac{1}{\sqrt{6}}(A_3 + C_1)\frac{\mathbf{h}_2^{\perp}}{|\mathbf{h}_2^{\perp}|},$$
$$X_2 = \frac{1}{\sqrt{6}}(B_1 + A_2)\frac{\mathbf{h}_3^{\perp}}{|\mathbf{h}_3^{\perp}|} + \frac{1}{\sqrt{6}}(C_2 - B_3)\frac{\mathbf{h}_1^{\perp}}{|\mathbf{h}_1^{\perp}|} - \frac{1}{\sqrt{6}}(A_3 + C_1)\frac{\mathbf{h}_2^{\perp}}{|\mathbf{h}_2^{\perp}|},$$

where the channel gain between the BS and user k is indicated by $\mathbf{h}_k \in \mathbb{C}^L$ $(k = 1, 2, 3)$. The first user will get $\mathbf{h}_1^H X_1 + \mathbf{n}_1$ and $\mathbf{h}_1^H X_2 + \mathbf{n}_2$, successively. The information is first extracted by the first user using both the received signal and the contents of their cache.

3.2 Review of Multi-Antenna Interference Management for Coded Caching in [2].

Following is a short overview of the scheme described in [2], which shares many concepts with the new scheme presented in this study. The original coded caching scheme and other similar works at the time focused only on the high SNR regime [1, 14, 31, 74]; however, the focus was subsequently shifted to the low SNR regime in [2]. Adjusting the multi-server coded caching scheme in [14] to the wireless multiple antenna setup improved the system's performance compared to traditional caching. However, employing ZF beamformers was later discovered to be highly suboptimal. This is due to the fact that at low SNR, the detrimental effects of inter stream interference and noise must be balanced to achieve an optimal performance. Authors in [2] follow the same cache placement as in [14] and the ZF beamformers in [14] are extended to generic multicast beamformers. In other words, ZF beamformers are replaced with the optimized beamformers, allowing the control of inter-stream interference, instead of completely eliminating the interference.

The received signal at the kth user terminal and time instant $i \in [n]$ can be written as

$$y_k(i) = \mathbf{h}_k^{\mathrm{H}}(i) \sum_{\mathcal{T} \subseteq \mathcal{S}} \mathbf{w}_{\mathcal{T}}^{\mathcal{S}}(i) \tilde{X}_{\mathcal{T}}^{\mathcal{S}}(i) + z_k(i), \tag{3.3}$$

where the channel vector between the base station and user k is denoted by $\mathbf{h}_k \in \mathbb{C}^L$, $\mathbf{w}_{\mathcal{T}}^{\mathcal{S}}$ is the multicast beamformer dedicated to users in subset $\mathcal{T}$ of set $\mathcal{S} \subseteq [K]$ of users, and $\tilde{X}_{\mathcal{T}}^{\mathcal{S}}(i)$ is the corresponding multicast message chosen from a unit power complex Gaussian codebook at time instant i. The size of $\mathcal{T}$ depends on the parameters K, M and N such that $|\mathcal{T}| = t + 1$, where $t \triangleq KM/N$. The generalized multicast beamformer design presented in this work can be created to fully take advantage of the interference-free signal space even when $L > K - t$. At low SNR, the general expressions for SINR is

$$SINR_k = \frac{|\mathbf{h}_k^{\mathrm{H}} \mathbf{w}_{\mathcal{T}}|^2}{\sum_{i \neq k} |\mathbf{h}_k^{\mathrm{H}} \mathbf{w}_i|^2 + N_0}, \tag{3.4}$$

which are handled directly in beamformer optimization problem in order to best balance the negative effects of inter stream interference and noise.

Authors have formulated the general optimization problem for symmetric rate maximization to minimize the delivery time. However, SINR constraints happen to be non-convex; hence, the resulting optimization problem is not convex. As a solution authors propose a successive convex approximation scheme to iteratively solve the problem. In their study it could be observed that in the low SNR regime, the symmetric rate is small while it becomes significantly larger as the the SNR is increased. Moreover, there exists a total number of $(t + L)(2\binom{t+L-1}{t} - 1)$ rate constraints in the beamformer optimization problem, it was emphasized in the study that even for small increase in input parameters like K and L, beamformer optimization problem became substantially complex.

This work assumes a SIC receiver and generalizes the multigroup multicast beamformer design to any combination of overlapping user groups. The number of multicast messages transmitted to $(t + L)$ users in a single transmission interval is $\binom{t+L}{t+1}$. Thus, here every user k will decode $\binom{t+L-1}{t}$ different messages, which increase exponentially with K, L, and N.

Here, a simple example is provided to illustrate how the optimized beamforming-precoders of this study operate. Here $K = 3$, $L = 2$, $N = 3$ and $M = 1$ input parameters were selected. These three files selected as $\{A, B, C\}$ and each file is split into three equal parts during the cache placement phase and caches are filled as follows:

$$Z_1 = \{A_1, B_1, C_1\}, \quad Z_2 = \{A_2, B_2, C_2\}, \quad Z_3 = \{A_3, B_3, C_3\}. \tag{3.5}$$

Suppose that, during the content delivery phase, Users $1-3$ request files $A-C$, respectively. Here, $t = 1$ and the subsets $\mathcal{S}$ and $\mathcal{T}$ will be of size 3 and $t+1 = 2$, respectively. The transmitter will then transmit the following signal vector

$$\sum_{\mathcal{T}\subseteq[3],|\mathcal{T}|=2} \mathbf{M}w_{\mathcal{T}}^{\mathcal{S}}\tilde{X}_{\mathcal{T}}^{\mathcal{S}} = \mathbf{w}_{1,2}[A_2 \oplus B_1] + \mathbf{w}_{1,3}[A_3 \oplus C_1] + \mathbf{w}_{2,3}[B_3 \oplus C_2], \tag{3.6}$$

where general $\mathbf{M}w_{\mathcal{T}}^{\mathcal{S}}$ are general multicast beamforming vectors. When User one receives a signal, it is recorded as

$$y_1 = (\mathbf{h}_1\mathrm{H}\mathbf{w}_{1,2})[A_2 \oplus B_1] + (\mathbf{h}_1\mathrm{H}\mathbf{w}_{1,3})[A_3 \oplus C_1] + (\mathbf{h}_1\mathrm{H}\mathbf{w}_{2,3})[B_3 \oplus C_2] + z_1.$$

The first user will extracts the information using $[A_2 \oplus B_1]$, $[A_3 \oplus C_1]$ received messages and with the help of its cache contents; appears as Gaussian interferenc $[B_3 \oplus C_2]$.

3.2.1 Complexity Reduction in beamformer optimization problem

Extending the fully overlapping strategy that was first used in the work, same authors later presented an efficient method for reducing the problem's complexity at both the transmitter and the receivers. They restrict the size of user subsets or the overlap between multicast messages. A parameter named $alpha$ determines how many users are served during a specific time period such that $t + \alpha \leq t + L$. In addition, the parameter $beta$ is added to manage the overlap between the parallel multicast messages. It specifies the maximum number of parallel messages that the SIC receiver should be able to distinguish from one another.

Here, a simple example is provided to illustrate how the optimized beamforming-precoders of this study operate. Here $K = 4$, $L = 3$, $N = 4$, $M = 1$ $\alpha = 3$ and $\beta = 1$ input parameters were selected. These three files selected as $\{A, B, C, D\}$ and during the cache placement phase, each file is divided into three parts of equal size, and caches are filled as follows:

$$\begin{aligned} Z_1 &= \{A_1, B_1, C_1, D_1\}, \quad Z_2 = \{A_2, B_2, C_2, D_2\}, \\ Z_3 &= \{A_3, B_3, C_3, D_3\}, \quad Z_4 = \{A_4, B_4, C_4, D_4\}. \end{aligned} \tag{3.7}$$

Suppose that, during the content delivery phase, Users $1-4$ request files $A-D$, respectively. Here, $t = 1$ and the subsets $\mathcal{S}$ and $\mathcal{T}$ will be of size 4 and $t+1 = 2$, respectively. The transmitter will then transmit the following signal vectors, Suppose that, during the content delivery phase, Users $1-3$ request files $A-C$, respectively. Here, $t = 1$ and the subsets $\mathcal{S}$ and $\mathcal{T}$ will be of size 3 and $t+1 = 2$, respectively. The transmitter will then transmit the following signal vectors as blocks

$$\begin{aligned} \tilde{X}_1 &= \mathbf{w}_{1,2}(A_2 \oplus B_1) + \mathbf{w}_{3,4}(C_4 \oplus D_3), \\ \tilde{X}_2 &= \mathbf{w}_{1,3}(A_3 \oplus C_1) + \mathbf{w}_{2,4}(B_4 \oplus D_2), \\ \tilde{X}_3 &= \mathbf{w}_{1,4}(A_4 \oplus D_1) + \mathbf{w}_{2,3}(B_3 \oplus C_2). \end{aligned} \tag{3.8}$$

It is evident that in this example, messages are grouped into $beta = 1$ groups so that each user only receives and decodes one message during each transmission block, resulting in low complexity decoding.

Introduction of the new variables (α, β) results in a higher subpacketization requirement. These new parameters results in transmitting the previously co-transmitted content in separate time intervals. Hence, the total number of mini files (subpacketization requirement) is

$$\binom{K}{t} \frac{(\alpha-1)!}{(\delta-1)!(\beta-1)!(t+\beta)!^{\delta-1}} \binom{K-t-1}{\alpha-1}, \tag{3.9}$$

where $\delta := \frac{t+\alpha}{t+\beta} \in \mathbb{N}$. Hence, the extended version of this work only applies for parameter values such that $t+\beta$ divides $t+\alpha$. Moreover, it is also pointed out in [2] that α and β can be varied such that this condition holds. In addition, by setting $\alpha = L$ and $\beta = \alpha$ full DoF can be achieved. Nevertheless, to achieve satisfactory finite SNR performance in cases where $t+\beta$ is not divisible $t+\alpha$, a novel scheme is required to overcome the inherent asymmetries in the scheme of [2].

3.3 Review of Low-Complexity High-Performance Cyclic Caching for Large MISO Systems in [3].

Following is a short overview of the scheme described in [3], which presents a novel coded caching scheme, with low complexity in terms of both subpacketization and beamformer design. This work could address many shortcomings of the two preceding works, such as an exponential increase in sub-packetization and computationally complex beamformer designs. However, this scheme has its own flaws, including limited input parameters, which are elaborated on in the followings. It is named as *Cyclic Multi-Antenna Coded Caching* because it employs diagonally shifted vectors in the placement and the content delivery stages.

The authors in [3] use a $K \times K$ binary placement matrix denoted as $\mathbf{V}$ for the cache placement. The first row of $\mathbf{V}$ contains t consecutive 1's and remaining elements in that row are all zeros. For example the cache placement matrix for $K = 5, t = 2$, and $L = 2$ is formed as

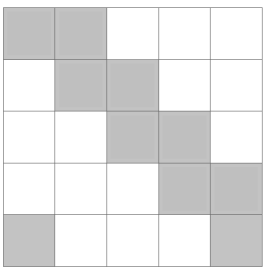
$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{3.10}$$

Thus, each file W is initially split into K packets W_p, $p \in [K]$, next each packet $W_p$is further split into $(t + L)$ smaller subpackets W_p^q. Then, W_p^q is stored in the cache memory of user k if $\mathbf{V}[p, k] = 1$. As an example, the cache contents of Users 1 and 2 are as follows:

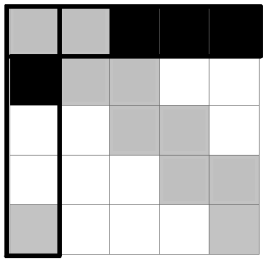
$$\begin{aligned} Z(1) &= \{W_1^q, W_5^q \; ; \; \forall W \in \mathcal{F}, \, q \in [4]\} \, , \\ Z(2) &= \{W_1^q, W_2^q \; ; \; \forall W \in \mathcal{F}, q \in [4]\} \, . \end{aligned} \tag{3.11}$$

From the above example, subpacketization of this scheme can be straightforwardly derived as $K \times (t + L)$. Resulting subpacketization of this example is computed as $K4 \times (2 + 2) = 20$.

In this work, the content delivery phase consists of K rounds, where $K - t$ transmission vectors are built in each round. Thus, the content delivery is completed after $K \times (K - t)$ transmissions. Here, in the placement matrix $\mathbf{V}$, illustrated in a graphical representation in Fig. 3.1a, each row denotes a packet index, and each column represents a user. Packets that are cached by each user are denoted by entries with a light shading. The darkly shaded entries in Fig. 3.1b indicate which packet indices of the requested files are transmitted during each transmission. The row and column indices of these darkly shaded entries are extracted from the packet and user index vectors. When building transmission vectors, circular shifts are made in two perpendicular directions, and as they move from round to round, they shift diagonally.



(a) Graphical illustration for Example 1.



(b) Illustration of the first transmission.

Authors in [3] have presented two different beamforming strategies one is the ZF beamforming and the other is optimized beamforming. The second technique, optimized beamforming, allows some interference to more evenly balance the effects of noise and inter-stream interference. According to the work of [3] traditional schemes (discussed above) that rely on multicast messages, i.e., [1, 2] benefits from multicasting gain but as discussed above they suffer from the added complexity in beamformer design. According to the authors unlike most other schemes in literature cyclic caching only relies on unicasting. However, it has been pointed out by [71] eliminating the use of multicasting causes performance losses. Anyhow, the work of [3] resulted high-performance optimized beamformers with low complexity at the receiver. This is due to the fact that unicasting of messages removes the requirement of decoding multiple data parts jointly at the same user during a single transmission. Thus, there is no requirement for complex receiver schemes such as SIC.

The second beamforming strategy is ZF beamforming, and it completely nulls interference at unwanted users and optimally allocates the power among the parallel streams. It is well-known that ZF heavily suffers in the low SNR regime, however authors in [3] have shown that decreasing the size of the user subset served during a given time interval from $(t + L)$ to $(t + \alpha)$ allows the worst user to be compensated while simultaneously achieving multiplexing gain. Consequently, employing these strategies achieves good performance in the low-to-medium SNR region at the expense of a reduced DoF. Despite the fact that this reduction in DoF is advantageous for subpacketization, which can be further reduce to $S = \frac{K(t+\alpha)}{\phi_\alpha^2}$, with $\phi_\alpha = \gcd(K, t, \alpha)$, where gcd(.) stands for greatest common divisor inside the brackets.

Overall, the structure of the novel scheme, which intrinsically does not require SIC receivers, and the flexibility provided by parameter α result in a scheme that achieves good performance over a broad range of the SNR spectrum while requiring minimal complexity in terms of precoder design and subpacketization. Thus, this scheme is the first in the literature to achieve the optimal DoF with a gradually increasing subpacketization as the network size increases, making it applicable to networks with a large number of users having a single limitation of $t \leq L$.

4 PROPOSED NOVEL GROUP ASSIGNMENT CODED CACHING SCHEME

4.1 System Model

Let us consider downlink transmission from a MISO cache-aided broadcast scenario. As shown in Fig. 4.1, a server with L antennas serves K cache enabled single-antenna users. The server has access to a library $\mathcal{F}$ of N files, where each file W has a size of F bits. Each user has a cache of MF bits, where $0 \leq M \leq N$. The total normalized cache size is denoted by $t = K\frac{MF}{NF}$. As in every coded caching scenario here also there are two phases, *Placement Phase* and *Delivery Phase*, which are briefly defined in the following.

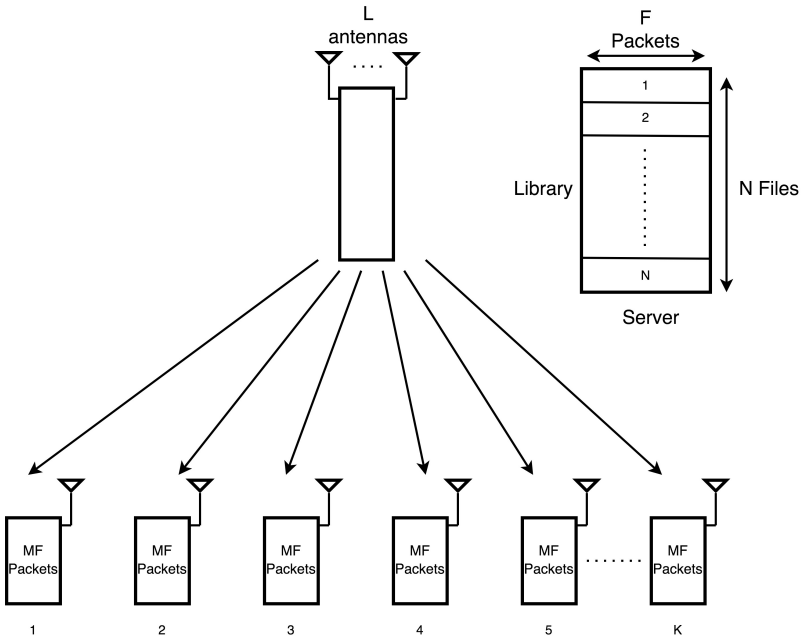


Figure 4.1: The coded caching system.

Placement Phase

During this stage, which occurs before the users' needs are made clear, the users' memories are partially filled with information from the library. Here every user $k \in [K]$ is able to store MF packets from the library, as $\mathcal{Z}_k$, where $\mathcal{Z}_k = \mathcal{Z}_k(W_1, \ldots, W_N)$ are the messages stored in the cache.

Delivery Phase

During this phase every user k request an arbitrary file $\mathbf{W}_{d_k}$ from the library, where $d_k \in [N]$. Demand vector is denoted as $\mathbf{d} = (d_1, d_2, \ldots, d_K)$. Upon a set of requests the base station transmits coded packets such that a selected set of $(t + L)$ users can reliably decode their requested files.

To generate decoded file $\widehat{W}_{d_k}$, user k must use its own cache contents Z_k in addition to the received signal from the wireless medium. The communication process at time slot t between the server and users can be modelled as

$$y_k(i) = \mathbf{h}_k^{\mathrm{H}}(i) \sum_{\mathcal{T} \subseteq \mathcal{S}} \mathbf{w}_{\mathcal{T}}^{\mathcal{S}}(i) \tilde{X}_{\mathcal{T}}^{\mathcal{S}}(i) + n_k(i), \tag{4.1}$$

where $\mathbf{h}_k \in \mathbb{C}^L$ represents the channel gain between the base station and user k, $\mathbf{w}_{\mathcal{T}}^{\mathcal{S}}$ is the multicast beamformer dedicated to users in subset $\mathcal{T}$ of set $\mathcal{S} \subseteq [K]$, and $\tilde{X}_{\mathcal{T}}^{\mathcal{S}}(i)$ represents multicast message chosen from a unit power complex Gaussian codebook at time instant i. In (4.1), the time index is denoted by i but it is ignored for simplicity in later parts. Here, n_k represents the noise of receiver k and it is assumed as circularly symmetric zero mean $z_k \sim \mathcal{CN}(0, N_0)$. In addition, it is presumed that this model is a slow fading one in which the channel vectors are constant over time steps i. Moreover, it is assumed that the CSI at all k users are perfectly known at the base station.

In (4.1) only a specific set of users $\mathcal{S} \subseteq [K]$ is catered at time instant i. This is due to the delivery of the requested files W_{d_k} may need multiple time slots depending on the selected transmission method and parametrization. This way a common coded message is benefited by all the members of the multicast group. Moreover, it can be noted that size of $\mathcal{T}$ varies based on the parameters M, N, K and $|\mathcal{T}| = t + 1$, where $t = KM/N$. This is known as global coded caching gain which is proportional to the aggregated cache size of users.

4.2 The Proposed Scheme

Throughout this thesis, cache placement is always followed according the original baseline scheme of [12]. Assuming $t = KM/N \in \mathbb{N}$, first each file (W_n) is divided into $\binom{K}{t}$ subfils such that $W_n = \{W_{n,\tau}, \tau \subset [K], |\tau| = t\}$, then user k caches all subfiles $W_{n,\tau}$ in which $k \in \tau$, $\forall\, n$. In the delivery phase, there are $\binom{K}{t+1}$ coded sub-files to be transmitted to $(t + 1)$ subsets of users $[K]$. These coded sub-files are transmitted according to

$$\oplus_{k \in \mathcal{T}} W_{d_k, \mathcal{T}\setminus\{k\}};\ \mathcal{T} \subseteq [K], |\mathcal{T}| = t + 1. \tag{4.2}$$

However, different from [12], there are L antennas available at the base station and they can collaboratively send coded messages. Thus, similar to the work of [1], here $(t + L)$ users can be served at a time. This implies that the global caching gain and the multiplexing gain are additives $(t + L)$. In [2], by combining all $\binom{t+L}{t+1}$ coded messages, with every message directed to the null space of $(L - 1)$ unwanted users, all of them are sent at once. Nevertheless, any user k will decode

$$m_k = \binom{t + L - 1}{t}$$

different messages, which grows at an exponential rate with K. Moreover, though using optimum multigroup multicast beamforming (as in [13]) significantly improves the system performance, it has a higher computational complexity than the simple ZF scheme. As discussed in Chapter 3, one such way to control these complications is to limit the overlap between the parallel multicast

messages being transmitted and it is specified using

$$N_p = \frac{t+L}{t+\Lambda}. \tag{4.3}$$

In (4.3) above, the number of parallel messages that the receiver should be able to distinguish from one another when using the SIC receiver is specified by $Lambda$. As mentioned in [2], the parameter $Lambda$, has a significant impact on receiver complexity. From the receiver perspective, $\Lambda > 1$ means that the desired multicast messages must be decoded using the SIC receiver structure. Note that here Λ can control the number of coded messages aimed at each user. When $\Lambda > 1$, the desired multicast messages must be decoded utilizing the SIC receiver structure from the receiver's point of view. Note that in this case, $Lambda$ has control over the number of coded messages sent to each user. For example if $K = 6$, $L = 5$ and $t = 1$, if we allow $\Lambda = L = 5$, then there will be $\binom{6}{2} = 15$ coded messages transmitted in parallel, of which every user would need to decode five messages. By contrast, in the same example if $\Lambda = 1$, there is only one coded message every user needs to decode from the many parallel messages. As a result, the beamformer optimization problem would be simplified, and each coded message could be delivered to its intended recipients at the highest possible rate.

In [13] certain number of parallel streams are allotted to each user which need to be decoded using the SIC receiver. In this thesis, however, we consider the same scenario as in [13] though, there can be no overlap between user groups that are served by different multicast messages. To implement this technique, we can limit the overlap among the multicast messages by partitioning users into several non-overlapping groups by setting $\Lambda = 1$ in (4.3). This leads to a simpler transmitter and receiver strategy where all multicast streams are delivered across different time slots as opposed to parallel transmission.

In the sequel, the floor operation of $\frac{t+L}{t+1}$ is considered as

$$N_m = \left\lfloor \frac{t+L}{t+1} \right\rfloor. \tag{4.4}$$

This way, at any given time, multiple messages (N_m) are transmitted simultaneously, each of which serves a different group of $t+1$ users. In contrast to scheme in [73] in this work reminder of $\frac{t+L}{t+1}$ is not ignored but considered by taking

$$N_{mod} = \mod\left(\frac{t+L}{t+1}\right). \tag{4.5}$$

Thus taking remainder after division (modulo operation) of $\frac{t+L}{t+1}$ results in serving all the practical input parameters of t and L. Latter sections of this paper demonstrate that the introduction of N_{mod} has complicated the new scheme. However this method results in a sizable reduction in complexity at the receiver compared to the [2], which necessitates the transmission of many more messages simultaneously at any given time.

In this proposed method we will have several multicast messages that will serve some non-overlapping groups of users and some unicast messages serving a single user which can be readily decodable using respective cache memories. This way full DoF of $t+L$ is achieved while subpacketization varies based on network parameters i.e., t, L, and K. To differentiate among different transmission techniques, we will first illustrate the procedures in four straightforward scenarios. Then, we elaborate on the generalization of the proposed scheme in Chapter 5.

4.2.1 Scenario 1 : $N_m \geq 1$ and $N_{mod} = 0$

First, for a better understanding of this scenario, two simple examples are provided. In both these examples when messages are transmitted no overlap is allowed among user groups, i.e., each user is served with one message at a time.

4.2.1.1 Example 1

As the first example in the following, $L = 4$, $K = 6$, $N = 6$, $M = 2$ is considered. According to (4.4), N_m and N_{mod} can be calculated. When only one user is served at a time, it leads to a simpler transmitter and receiver strategy where all $\binom{6}{3} = 20$ multicast streams are delivered across ten orthogonal time slots. In time slots 1–10, the multicast beamforming vectors are constructed as

$$\begin{aligned}
X_1 &= \mathbf{u}_{4,5,6}(A_{2,3} \oplus B_{1,3} \oplus C_{1,2}) + \mathbf{u}_{1,2,3}(D_{5,6} \oplus E_{4,6} \oplus F_{4,5}),\\
X_2 &= \mathbf{u}_{3,5,6}(A_{2,4} \oplus B_{1,4} \oplus D_{1,2}) + \mathbf{u}_{1,2,4}(E_{3,6} \oplus F_{3,5} \oplus C_{5,6}),\\
X_3 &= \mathbf{u}_{3,4,6}(A_{2,5} \oplus B_{1,5} \oplus E_{1,2}) + \mathbf{u}_{1,2,5}(F_{3,4} \oplus C_{6,4} \oplus D_{3,6}),\\
X_4 &= \mathbf{u}_{4,5,6}(A_{2,3} \oplus B_{1,3} \oplus C_{1,2}) + \mathbf{u}_{1,2,3}(D_{5,6} \oplus E_{4,6} \oplus F_{4,5}),\\
X_5 &= \mathbf{u}_{4,5,6}(A_{2,3} \oplus B_{1,3} \oplus C_{1,2}) + \mathbf{u}_{1,2,3}(D_{5,6} \oplus E_{4,6} \oplus F_{4,5}),\\
X_6 &= \mathbf{u}_{4,5,6}(A_{2,3} \oplus B_{1,3} \oplus C_{1,2}) + \mathbf{u}_{1,2,3}(D_{5,6} \oplus E_{4,6} \oplus F_{4,5}),\\
X_7 &= \mathbf{u}_{4,5,6}(A_{2,3} \oplus B_{1,3} \oplus C_{1,2}) + \mathbf{u}_{1,2,3}(D_{5,6} \oplus E_{4,6} \oplus F_{4,5}),\\
X_8 &= \mathbf{u}_{4,5,6}(A_{2,3} \oplus B_{1,3} \oplus C_{1,2}) + \mathbf{u}_{1,2,3}(D_{5,6} \oplus E_{4,6} \oplus F_{4,5}),\\
X_9 &= \mathbf{u}_{4,5,6}(A_{2,3} \oplus B_{1,3} \oplus C_{1,2}) + \mathbf{u}_{1,2,3}(D_{5,6} \oplus E_{4,6} \oplus F_{4,5}),\\
X_{10} &= \mathbf{u}_{4,5,6}(A_{2,3} \oplus B_{1,3} \oplus C_{1,2}) + \mathbf{u}_{1,2,3}(D_{5,6} \oplus E_{4,6} \oplus F_{4,5}).
\end{aligned} \tag{4.6}$$

These multicast beamforming vectors along with interference indicators are built according to the theory in Chapter 3 where $\oplus$ denotes bit-wise XOR and the brackets of the interference indicator sets are dropped for notational simplicity. Here, user indices for multicast messages are initially generated by choosing three multicasting sets from the set [1:6]; $\binom{1:6}{3}$. Next, they are grouped such that at each transmission block, only one message is received and decoded by each user. Then, based on (4.4), in every time slot, all six users are catered with two parallel multicast streams. Every stream causes inter stream interference to three different users who do not belong to the specified multicast group. Thus, a base station, having at least four antennas, will have sufficient spatial DoF to control the inter stream interference among multicast streams. Since overlap is not permitted, each user decodes a single multicast message during a specific time period. For these reasons, neither SIC receiver nor multiple-access channel rate region constraints are required in the formulation of the problem.

In this scenario where systems having input parameters satisfying conditions $\frac{t+L}{t+1} \in \mathbb{N}$ will have the same subpacketization level as the case with single antenna in [12]. Hence, when in every scenario where $N_{mod} = 0$, subpacketization is equivalent to

$$\text{Subpacketization} = \binom{K}{t}, \tag{4.7}$$

where in this case $\binom{6}{2} = 15$. For this specific input parameters it was observed that subpacketization is equivalent to the case of [2] work. In depth comparison among schemes are provided in Chapter 6.

4.2.1.2 Example 2

Secondly, $L = 5$, $K = 7$, $t = 1$, input parameters are selected. Similar to previous example N_m and and N_{mod} is calculated as three and zero respectively. Hence $\binom{6}{2} = 21$ multicast streams are delivered across seven orthogonal time intervals/slots. In time slots 1–7, the multicast beamforming vectors are generated as

$$\begin{aligned}
X_1 &= \mathbf{u}_{3,4,5,6}(A_2 \oplus B_1) + \mathbf{u}_{1,2,5,6}(C_4 \oplus D_3) + \mathbf{u}_{1,2,3,4}(E_6 \oplus F_5),\\
X_2 &= \mathbf{u}_{2,4,5,7}(A_3 \oplus C_1) + \mathbf{u}_{1,3,5,7}(B_4 \oplus D_2) + \mathbf{u}_{1,3,2,4}(E_7 \oplus G_5),\\
X_3 &= \mathbf{u}_{2,3,6,7}(A_4 \oplus D_1) + \mathbf{u}_{1,4,6,7}(B_3 \oplus C_2) + \mathbf{u}_{1,4,2,3}(F_7 \oplus G_6),\\
X_4 &= \mathbf{u}_{2,6,3,7}(A_5 \oplus E_1) + \mathbf{u}_{1,5,3,7}(B_6 \oplus F_2) + \mathbf{u}_{1,5,2,6}(C_7 \oplus G_3),\\
X_5 &= \mathbf{u}_{2,5,4,7}(A_6 \oplus F_1) + \mathbf{u}_{1,6,4,7}(B_5 \oplus E_2) + \mathbf{u}_{1,6,2,5}(D_7 \oplus G_4),\\
X_5 &= \mathbf{u}_{3,5,4,6}(A_7 \oplus G_1) + \mathbf{u}_{1,7,4,6}(C_5 \oplus E_3) + \mathbf{u}_{1,7,3,5}(D_6 \oplus F_4),\\
X_5 &= \mathbf{u}_{3,6,4,5}(B_7 \oplus G_2) + \mathbf{u}_{2,7,4,5}(C_6 \oplus F_3) + \mathbf{u}_{2,7,3,6}(D_5 \oplus E_4).
\end{aligned} \tag{4.8}$$

Similar to above example above multicast transmssion vectors are also built according to the theory in Chapter 3. Here, user indices for multicast messages are initially generated by choosing three multicasting sets from the set [1:7]; $\binom{1:7}{2}$. Next, they are grouped such that each user receives and decode only one message at each transmission block. Then, based on (4.4), each time slot includes 3 parallel multicast streams for each of the 7 users. Moreover, each stream causes inter stream interference to 4 other users not included in the given multicast group. In order to manage the inter stream interference between multicast streams, the base station, which has at least 5 antennas, has sufficient spatial DoF. No overlap is allowed, so each user decodes a single multicast message during a specific time period. The beamformer problem can therefore be expressed without the use of either SIC receiver or multiple-access channel rate region constraints.

Similarly to the example above, here we will have the same subpacketization level as the case with single antenna in [12], hence subpacketization sums up to $\binom{7}{1} = 7$. However unlike the example above, for these specific input parameters subpacketization of scheme in [2] is much higher. Assuming β in that particular scheme is one, subpacketization of the scheme in [2] can be calculated as

$$\binom{K}{t}\binom{K-t-1}{L-1}\frac{(L-1)!}{(\delta-1)!(\beta-1)!(t+\beta)!^{\delta-1}} = 105, \tag{4.9}$$

where,

$$\delta = \frac{t+L}{t+\beta} = 3.$$

Proposed new scheme could achieve a 15 times less subpacketization level compared to [2]. This is a direct result of avoiding further splitting at delivery phase. Even-though these two schemes share same strategy at the cache placement stage, scheme in [2], requires its sub-files again to be splitted into mini-files. More precisely, in this case each sub-file will be split into

$$\binom{K-t-1}{L-1}\frac{(L-1)!}{(\delta-1)!(t+1)!^{\delta-1}},$$

where delta is equal to three. In contrast, when $\frac{t+L}{t+1} \in \mathbb{N}$ proposed scheme has to group sub-files accordingly in the delivery phase without any need for further splitting.

4.2.2 Scenario 2 : Special Case $N_m = 1$ and $N_{mod} > 0$

A straightforward illustration is provided in order to make this particular scenario more understandable. In this example, we consider $L = 3$, $K = 5$, $N = 5$, and $M = 2$, where as in the previous case, no overlap is allowed among user groups. According to (4.4) and (4.5) we can calculate, N_m= 1 and N_{mod} = 2. Here, only one multicast set is allowed to be transmitted while two messages can be unicasted. Since $N_m = 1$ all the selected multicast streams can be delivered across orthogonal time intervals/slots without needing them to group. The selection of multicast and unicast messages can be performed as outlined below.

$$
\begin{array}{cc}
\text{MTS} & \text{UTS} \\
(1\ 2\ 3) & +4+5, \\
(1\ 2\ 4) & +3+5, \\
(1\ 2\ 5) & +3+4, \\
(1\ 3\ 4) & +2+5, \\
(1\ 3\ 5) & +2+4, \\
(1\ 4\ 5) & +2+3, \\
(2\ 3\ 4) & +1+5, \\
(2\ 3\ 5) & +1+4, \\
(2\ 4\ 5) & +1+3, \\
(3\ 4\ 5) & +1+2,
\end{array}
\tag{4.10}
$$

where, MTS stands for Multicast Transmssion Symbols and UTS indicates Unicast Transmssion Symbols. Now we wil assume the demand set $\{A, B, C, D, E\}$. Then the first transmission vector in the first round is built as

$$X_1 = u_{4,5}(A_{2,3} + B_{1,3} + C_{1,2}) + u_{1,5}D_{2,3} + u_{1,4}E_{2,3}. \tag{4.11}$$

According to Fig. 4.2, these 10 streams consisting of unicast and multicast messages can not be simply delivered across 10 orthogonal time intervals/slots. This is because the number of transmissions must be increased in order to maintain the symmetry of the scheme, such that the same group of users receive different unicast messages along with the same multicast messages during three $(t + 1)$ distinct time slots as shown in Fig. 4.2.

In this scenario, systems satisfying input parameters $\lfloor \frac{t+L}{t+1} \rfloor = 1$ and $(\mod \frac{t+L}{t+1}) \geq 1$ would have a subpacketization level of $(t + L)$ times the original scheme [12] of single antenna case. However, this method provides a low complexity decoding,because each user must decode a single message for each transmission block. subpacketization level of this special case is

$$\text{Subpacketization} = \binom{K}{t}(t + L) = 50. \tag{4.12}$$

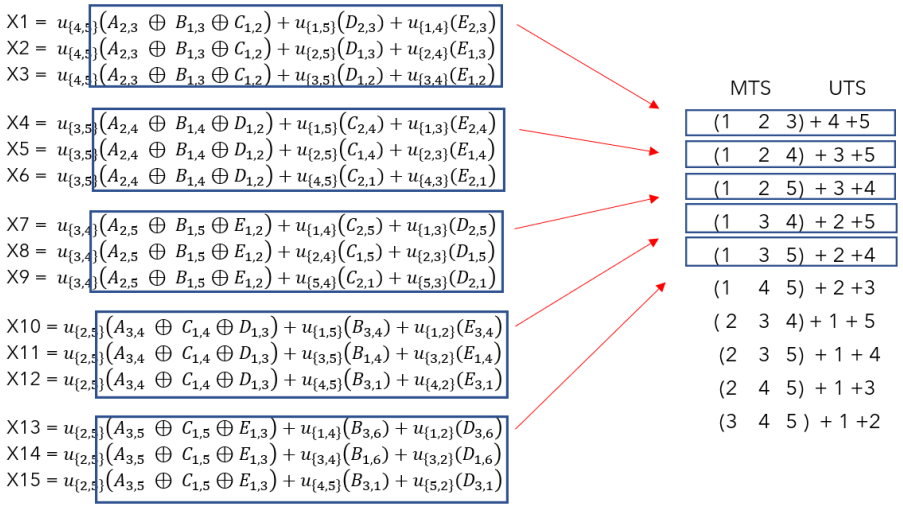


Figure 4.2: Repeating $(t + 1)$ transmissions.

4.2.3 Scenario 3: Most General Case $N_m \geq 1$ and $N_{mod} > 0$

A simple example is provided to illustrate the most general case. In this example, we consider $L = 4$, $K = 5$, $N = 5$, $M = 1$ and no overlap is allowed among user groups. Hence, according to (4.4) and (4.5) we can calculate, $N_m = 2$ and $N_{mod} = 1$. Hence, the maximum number of multicast messages that can be transmitted at the same time is two, whereas the rest can be unicasted. In this example, we transmit two multicast messages at a time to fully benefit from multicasting gain. Hence only a single unicast message is sent. The selection of multicast and unicast messages can be performed as outlined below.

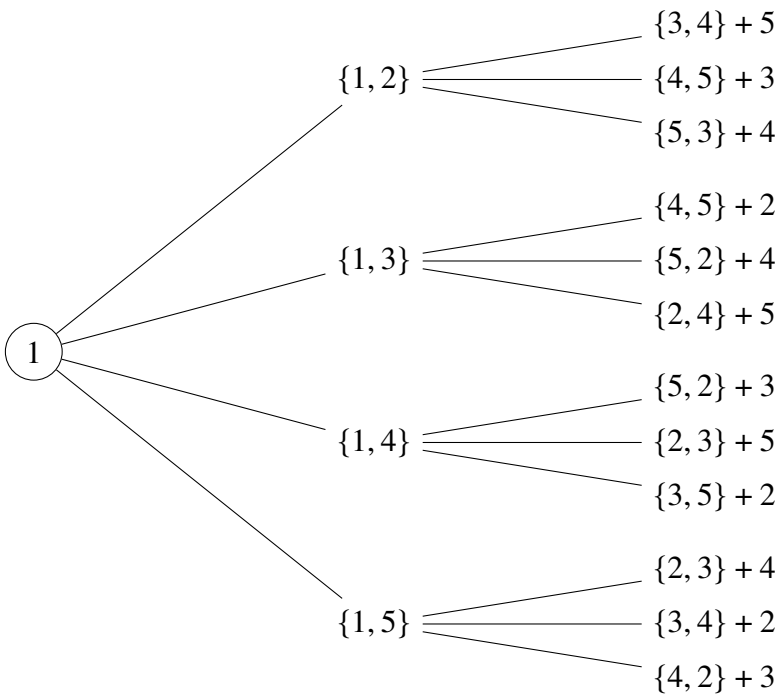


Figure 4.3: Tree diagram prioritizing the first user.

In this work tree diagrams are employed to select multicast and unicast message symbols, in each tree diagram one user is prioritized in each phase. Above tree Fig. 4.3 is for the first user and here all the missing data for the first user is targeted. Initially, multicast message symbols in the 1st branch are chosen according to the method outlined in Scenario 1. However, now in this specific tree diagram, only multicast message symbols starting with index 1 are selected (i.e., $\{1,2\}, \{1,3\}, \{1,4\}$ and $\{1,5\}$). Next, second multicast group in the second branch must be chosen such that it does not share any user from the first multicast group. After selecting such a multicast message symbol having a size of $K-(t+1)=4$ they are placed on the initial row of a circulant matrix with a backward shift. This is a symmetric circulant matrix and it is created by constructing new rows through circular shifting of the previous row,

$$\mathbf{C} = \begin{bmatrix} 3 & 4 & 5 \\ 4 & 5 & 3 \\ 5 & 3 & 4 \end{bmatrix}. \tag{4.13}$$

First two $(t+1)$ columns of this circulant matrix $\mathbf{C}$ is selected as the multicast message indices for the second branches. After all the multicast groups are selected, unicast transmission indices are selected by the remaining users. Those are clearly illustrated on the leaves of above tree diagram. Similarly, for Users $2-5$ another four tree diagrams are needed to construct all the transmission symbols, which are given in Appendix 2. This example is further elaborated when discussing the general case in the next chapter.

Unlike in [2], employing aforementioned method helps to reduce the frequency of serving same users. However, here, when constructing transmission vectors as described in the preceding scenario, the number of transmissions increases compared to [73]. For example, in this case, the same message must be sent four times to four distinct users. To clarify this, we will assume demand set as $\{A, B, C, D, E\}$. Then, accordingly, the preliminary transmission vector in the initial round is built according to

$$X_1 = \mathbf{u}_{3,4,5}(A_2 \oplus B_1) + \mathbf{u}_{1,2,5}(C_4 \oplus D_3) + \mathbf{u}_{2,3,4}E_1, \tag{4.14}$$

where the interference indicator sets' brackets have been removed for the ease of notation.

For better clarification, let us assume ZF beamformers are used here. Then, after x_1 is transmitted, User 1 receives

$$y_1(1) = h_1^H \mathbf{u}_{3,4,5}(A_2 \oplus \underline{B_1}) + \underline{\underline{h_1^H \mathbf{u}_{1,2,5}(C_4 \oplus D_3)}} + \underline{h_1^H \mathbf{u}_{2,3,4}E_1} + w_1, \tag{4.15}$$

where terms with a single or double underline indicate an interference. From the discussion in Chapter 3, it is evident that the, C_1, D_1, and E_1 are readily available in the cache of user 1. And also, single underlined terms are reconstructed in the signal-level and they are removed from received signal. In contrast, double-underlined term is suppressed by adhering to the definition of the interference indicator sets at User 1 by the beamforming vectors. Due to this User 1 can decode A_2 with minimal interference. Likewise Users 2 to 5 can decode B_1, C_4, D_3 and E_1, respectively. Subpacketization of this scenario is parameterized by (4.16) where $\Delta = \prod_{i=1}^{N_m-1}(k-i(t+1))$. Hence, a system with input parameters satisfying $\lfloor \frac{t+L}{t+1} \rfloor \geq 1$ *and* $(\text{mod } \frac{t+L}{t+1}) \geq 1$ would have a subpacketization level of $N_m^2(t+1)^2\Delta + \binom{K-1}{t}(L-1)$ times the original scheme [12] of single antenna case. Simplified subpacketization of the above example is

$$\text{Subpacketization} = \binom{K}{t}\left(4(t+1)^2\Delta + \binom{K-1}{t}(L-1)\right). \tag{4.16}$$

Due to the fact that each user must decode only one message per transmission block, this method provides a low complexity decoding.

4.2.4 Scenario 4: $K > t + L$

In this scenario, the method described in [1] is applied whenever the number of users exceeds $(t + L)$. For better understanding a simple example is provided. We consider $L = 2$, $K = 4$, $N = 4$, and $M = 1$, where no overlap is allowed among user groups. Hence, according to (4.4) and (4.5), $N_m = 1$ and $N_{mod} = 1$. Following Scenario 1, only a single unicast and single multicast message is sent in each transmission, serving $(t + L)$ users at a time. Similar to [1], there are $\binom{K}{t+L}$ round of transmissions. In the first round, user group, $\{1, 2, 3\}$ is served, where the transmission vectors are as follows

$$\begin{aligned} X_1^1 &= \mathbf{u}_3(A_2 \oplus B_1) + \mathbf{u}_2 C_1, \quad X_2^1 = \mathbf{u}_3(A_2 \oplus B_1) + \mathbf{u}_1 C_2, \\ X_3^1 &= \mathbf{u}_2(A_3 \oplus C_1) + \mathbf{u}_3 B_1, \quad X_4^1 = \mathbf{u}_2(A_3 \oplus C_1) + \mathbf{u}_1 B_3, \\ X_5^1 &= \mathbf{u}_1(B_3 \oplus C_1) + \mathbf{u}_3 A_2, \quad X_6^1 = \mathbf{u}_1(B_3 \oplus C_1) + \mathbf{u}_2 A_3. \end{aligned} \tag{4.17}$$

Similarly, subsequent user groups are served as $\{1, 2, 4\}$, $\{1, 3, 4\}$ and $\{2, 3, 4\}$, respectively. In (4.18) user group $\{1, 2, 4\}$ is being served as

$$\begin{aligned} X_1^1 &= \mathbf{u}_4(A_2 \oplus B_1) + \mathbf{u}_2 D_1, \quad X_2 = \mathbf{u}_4(A_2 \oplus B_1) + \mathbf{u}_1 D_2, \\ X_1 &= \mathbf{u}_2(A_4 \oplus D_1) + \mathbf{u}_4 B_1, \quad X_2 = \mathbf{u}_2(A_4 \oplus D_1) + \mathbf{u}_1 B_4, \\ X_1 &= \mathbf{u}_1(B_3 \oplus D_1) + \mathbf{u}_4 A_2, \quad X_2 = \mathbf{u}_1(B_3 \oplus D_1) + \mathbf{u}_2 A_4. \end{aligned} \tag{4.18}$$

Next in (4.19) user group $\{1, 3, 4\}$ is being served as

$$\begin{aligned} X_1 &= \mathbf{u}_4(A_3 \oplus C_1) + \mathbf{u}_3 D_1, \quad X_2 = \mathbf{u}_4(A_3 \oplus C_1) + \mathbf{u}_1 D_2, \\ X_1 &= \mathbf{u}_3(A_4 \oplus D_1) + \mathbf{u}_4 C_1, \quad X_2 = \mathbf{u}_3(A_4 \oplus D_1) + \mathbf{u}_1 C_3 \\ X_1 &= \mathbf{u}_1(C_4 \oplus D_3) + \mathbf{u}_4 A_3, \quad X_2 = \mathbf{u}_1(C_4 \oplus D_3) + \mathbf{u}_3 A_4. \end{aligned} \tag{4.19}$$

Lastly in (4.20) user group $\{2, 3, 4\}$ is being served as

$$\begin{aligned} X_1 &= \mathbf{u}_4(B_2 \oplus C_1) + \mathbf{u}_2 D_2, \quad X_2 = \mathbf{u}_4(B_2 \oplus C_1) + \mathbf{u}_1 D_3 \\ X_1 &= \mathbf{u}_3(B_3 \oplus D_1) + \mathbf{u}_3 C_2, \quad X_2 = \mathbf{u}_3(B_3 \oplus D_1) + \mathbf{u}_1 C_4 \\ X_1 &= \mathbf{u}_2(C_3 \oplus D_1) + \mathbf{u}_3 B_3, \quad X_2 = \mathbf{u}_2(C_3 \oplus D_1) + \mathbf{u}_2 B_4. \end{aligned} \tag{4.20}$$

In general, $K > t + L$ transmission round increase from one to $\binom{K}{t+L}$ rounds. Therefore, N should be further splitted in order to accommodate increased transmissions so that in every time instance the targeted users receive new mini files from each coded mini file. In the above example there are four rounds of transmissions and those extra rounds cause a $\binom{K-t-1}{L-1}$ multiplicative increase in subpacketization. From above (4.17) - (4.20) subpacketization level of above discussed example can be derived as

$$\binom{K}{t}(t + L)\binom{K - t - 1}{L - 1}. \tag{4.21}$$

.

5 GENERAL CASE FORMULATION AND ALGORITHM

This chapter presents a general case for each of the previously discussed scenarios. Formulation of the newly proposed group assignment beamformer design for the general case is discussed, as well as the algorithm for the same case. Proofs for this chapter are presented in Appendix 3.

In this thesis, cache placement is followed according to the original baseline scheme of [12] due to its low complexity. However, unlike in [12], there are L antennas available at the base station, and they can send coded messages together. Here $(t + L)$ users can be benefited like in the work of [1]. Global caching gain and the multiplexing gain are additive, reducing delay by a multiplicative based factor of $(t+1)/(t+L)$, which leads to a normalized delay of $(K-t)/(t+L)$. All the messages can be combined such that $\binom{t+L}{t+1}$ coded messages, with every message directed in to the null space of $(L-1)$ undesired users and they can be sent at the same time.

In the current state-of-the-art, any user k must be able decode $m_k = \binom{t+L-1}{t}$ different messages, which increases exponentially when K, L, N are increased. Moreover, when delivering messages, even though using optimum multigroup multicast beamforming (like in [13]) significantly increase the performance of the system, in particular at low SNR, it also introduces a complex beamformer optimization problem which is significantly complex than the simple ZF scheme. As discussed in Chapter 4, one way to control these complications is to control the overlap among the multicast messages transmitted in parallel. In order to implement this technique, we can limit the overlap among the multicast messages by partitioning users into groups by setting $\Lambda = 1$ in (4.3). This results in a more straightforward transmitter and receiver strategy where all multicast streams are delivered over various orthogonal time slots/intervals rather than transmitting them all simultaneously.

5.1 General Case Formulation

The Scenarios (1-4) discussed previously provided a thorough understanding (especially Scenario 3) of how this scheme operates. As a preliminary step, let us define $N_m = \lfloor \frac{t+L}{t+1} \rfloor$. This way, N_m coded messages are transmitted simultaneously at any given time, where each message serves a group of $(t + 1)$ users. In contrast to the scheme in [73], in this work the reminder of $\frac{t+L}{t+1}$ is not ignored. Thus, the resulting scheme can serve networks with all the practical parameters of t and L. To do so, we will have several multicast messages that will serve some non-overlapping groups of users and some unicast messages serving a single user. These unicast messages should be readily decodable using respective cache contents at each user. This way, full DoF of $(t + L)$ is achieved while subpacketization varies in different use cases.

The following tree diagram illustrates the essence of this scheme's formulation. It is a general tree diagram that illustrates how multicast and unicast symbols should be selected so that the symmetricity of the scheme is maintained at all times. There should be K total tree diagrams, but when $K > t + L$, the number of tree diagrams increases exponentially by $K\binom{K}{t+1}$. In the branches of the tree diagram, multicast symbols are indicated within $\{\}$, and unicast symbols are appended to multicast symbol sets at the end of each branch (as leaves). Each tree diagram can contain a maximum of N_m number of subbranches. The selection of multicast symbols and unicast symbols is discussed below.

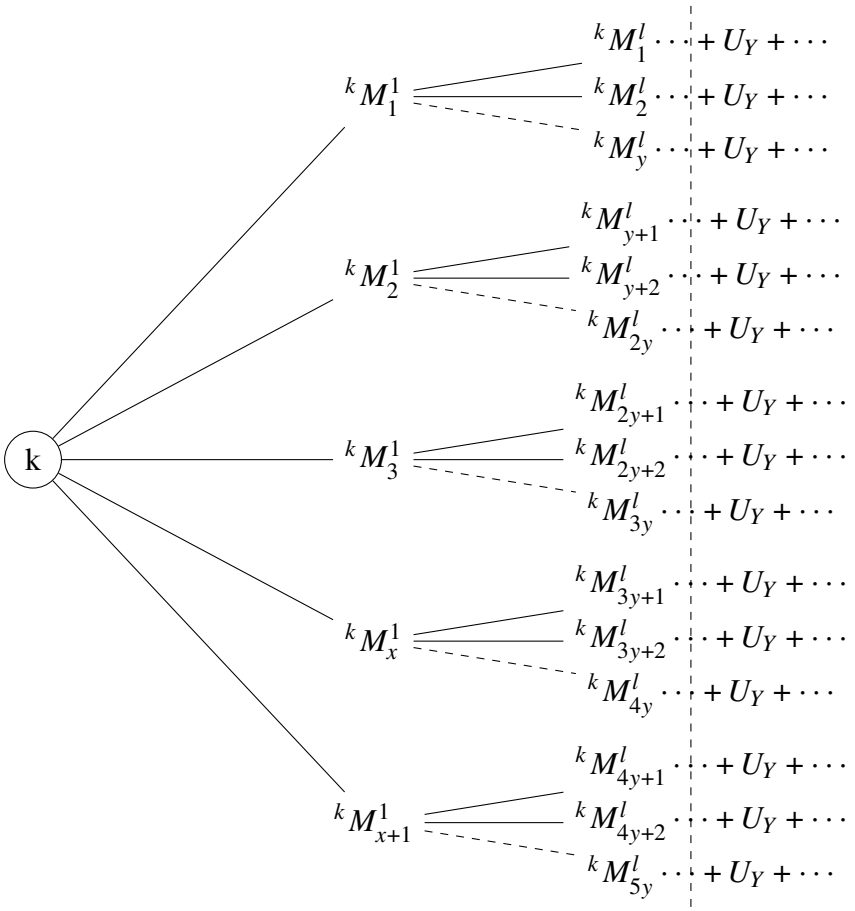


Figure 5.1: Tree diagram for an arbitrary kth user.

In the above tree diagram Fig. 5.1, ${}^kM_x^1$ is an arbitrary multicast user-set, where k represents an arbitrary tree diagram and x represents any multicast user-set in the first branch. Moreover, index l represents the number of multicast groups and index y represents arbitrary multicast user-sets in preceding branches. The path from the root to a leaf represents the transmission of a set of codewords consisting of multicast and unicast messages. Since there is no overlap among the codewords, each user receives and decodes only one message at each transmission block. The tree diagram is built in three phases; first selects multicast user-sets to first-most branches, the second selects multicast user-sets to second and all other subsequent branches; and lastly, unicast user-sets are selected to leaves.

As the first phase, in a given set of users, multicast user-sets are obtained by choosing $(t+1)$ user sets from the set of [K] users as in

$$\text{Multicast user sets} = \{S|\ S \subset [1:K], |S| = t+1\}.$$

These user-sets contain the multicast symbols assigned to all the users. Next, they must be reorganized so that when k is there inside an user-set, that user set should be included into the kth user group as in

$${}^kM_x^1 \in \{S|\ S \subset [1:K], |S| = t+1, k \in S\}. \tag{5.1}$$

Then, in the first phase, there are $\binom{K-1}{t}$ number of user sets in the first user group. These selected user sets are assigned to the first branches and hence they become the first muticasting group

(there are N_m multicasting groups in total). In Scenario 3, (cf. Chapter 3) all the multicast user-sets are selected as $\{1,2\}, \{1,3\}, \{1,4\}, \{1,5\}, \{2,3\}, \{2,4\}, \{2,5\}, \{3,4\}, \{3,5\}$ and $\{4,5\}$ but only multicast message symbols starting with index 1 (i.e., $\{1,2\}, \{1,3\}, \{1,4\}$ and $\{1,5\}$) are selected for the first-most branches of first tree diagram.

In the second phase, multicasting user-sets are selected for subsequent branches ($l > 1$). Here, multicasting sets must be chosen based on a symmetric circulant matrix. Moreover, second multicasting user-sets in the second branch must be chosen such that it does not share any users from the previous multicast groups. Before providing a detailed analysis of this phase, example of Scenario 3 is continued here. Assume {1,2} is on the first branch and second multicasting user-sets are to be selected, so now they are selected through a symmetric circulant matrix whose first row consist of {3,4,5} (which is the same as $[k] - [1,2]$). The resulting circulant matrix is given in (4.13) and after that, first $t+1$ columns of (4.13) is selected and a new matrix is formed

$$\tilde{C}^{t+1}_{[K]-M^l_x} = \begin{bmatrix} 3 & 4 \\ 4 & 5 \\ 5 & 3 \end{bmatrix}. \tag{5.2}$$

Consecutively, rows of (5.2) are selected and placed on sub-branches as second multicasting user-sets as in Scenario 3.

In general, second multicasting user-sets are assigned such that

$${}^{k}M^l_y \in \tilde{C}^{t+1}_{[K]-g({}^{k}M^l_y)}, \tag{5.3}$$

where, $\tilde{C}^{t+1}_{[K]-g({}^{k}M^l_x)}$ is a matrix constructed using first $(t+1)$ columns of another symmetric circulant matrix

$$C = \begin{bmatrix} c_0 & c_1 & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_1 & & c_2 \\ \vdots & & \ddots & \vdots & \\ & & \ddots & \ddots & \\ & & \cdots & & \end{bmatrix},$$

whose first row is $[K] - g({}^{k}M^l_y)$. Function $g({}^{k}M^l_x)$ is the union of the ancestors of ${}^{k}M^l_y$ (all the multicast user-sets appeared on the branches before). Hence resulting matrix $\tilde{C}$ will have a size of $(K - |g({}^{k}M^l_y)| \times (t+1))$. Lastly, rows of $\tilde{C}^{t+1}_{[K]-g({}^{k}M^l_x)}$ matrix are selected and placed on subsequent sub branches.

In the third phase, after N_m multicast groups are selected and placed on the tree diagram, user indices for unicasting (U_Y) should be added as leaves. As shown in the general tree diagram. Condition for selecting unicast users

$$U_Y \in \mathcal{K} \setminus \left({}^{k}M^{N_{\text{mod}}}_x \cup \left(\bigcup M^l_z \right) \right), Y = 1, \ldots, N_{mod}, \tag{5.4}$$

where M^l_z represent the ancestors of $(M^{N_{\text{mod}}}_x + U_1 + \ldots + U_{N_{\text{mod}}})$ (i.e., the x-th leaf of the tree), where '+' sign indicates the union. Simply these unicast user indices are constructed by the remaining users after all the multicasts. In Scenario 3, after all the multicasting user-sets are selected one user index always remains and that is selected as the unicast user.

Note: In the special case of $N_{mod} = 0$, $C^{t+1}_{[K]-g({}^{k}M^{l}_{y})}$ should only represent first row of the above said matrix, resulting only one branch for every step in the tree diagram.

5.1.1 Algorithm for Building the Tree Diagrams

In all the algorithms, we assume that sets are ordered (or equivalently, sets are arrays). Arrays are one-indexed (the first index is 1 as opposed to 0). If $\mathcal{A}$ is a set $\mathcal{A}[1 \dots n]$ means the first n elements of the set and $\mathcal{A}[1]$ is the first element of the set. As it was mentioned earlier, $[n]$ where $n \in \mathbb{N}$ denotes the set $\{1, \dots, n\}$. Algorithm 1 describes how the cycling procedure happens in an arbitrary set for n number of places.

Algorithm 1 Cycling procedure

procedure ROLL($\mathcal{A}$, n)
    $\mathcal{B} \leftarrow \mathcal{A}$
    $\mathcal{B}[1 \ \dots \ |\mathcal{B}| - n] \leftarrow \mathcal{B}[n+1 \ \dots \ |\mathcal{B}|]$
    $\mathcal{B}[|\mathcal{B}| - n + 1 \ \dots \ |\mathcal{B}|] \leftarrow \mathcal{B}[1 \ \dots \ n]$
    return $\mathcal{B}$

Algorithm 2 starts the first phase of the tree diagram construction. In this phase, the first multicasting groups are chosen for the first-most set of branches in all K tree diagrams. $\mathcal{T}$ describes the set of trees. **Node1** represents a node in a branch of a tree, and every **Node1** has a value, a parent, and an array of children stored in it. $\mathcal{M}$ represents all the multicast user sets in a matrix. Next grouping of those sets into first-most set of branches is described. k is some user and M is some multicast user set. If k is a member of M, the kth tree gets assigned M as a child node. N is the node associated with M and $\mathcal{T}[k]$ is the kth tree. This step will conclude assigning ${}^{k}M^{1}_{x}, \forall k, \forall x$ to all the K trees as described in (5.1).

Algorithm 2 Tree construction (Phase 1)

Require: $K, L, t \in \mathbb{N}$
Ensure: $\mathcal{T}$: the set of trees
struct NODE1
    value : ARRAY $\vee$ INT
    parent : NODE
    children : ARRAY
end struct
$\mathcal{M} \leftarrow \{\mathcal{S} | \mathcal{S} \subset [K], |\mathcal{S}| = t + 1\}$.
for $k \in [K]$ **do**
    for $M \in \mathcal{M}$ **do**
        if $k \in M$ **then**
            $N \leftarrow$ Node $(M, \mathcal{T}[k], \varnothing)$
            $\mathcal{T}[k].children \leftarrow \mathcal{T}[k].children \cup \{N\}$

Next, Algorithm 3 describes both the second and third phases of the tree diagram construction, where multicasting user-sets and unicast indices are selected, respectively, for subsequent

branches ($l > 1$). Unlike the first phase (Algorithm 2), here, **Node2** stores all the remaining indices of a set after one allocation is performed. This is indicated by the new entry inside **struct** as $left$. First, children in the branches who are already allocated from the first phase are again allocated into a matrix named $\mathcal{D}$. Next, D represents element from the $\mathcal{D}$ while $\mathcal{A}$ gets allocated with the remaining indices. If the length of the $\mathcal{A}$ is greater than $(t+1)$, $ROLL$ (Algorithm 1) is called to cyclically shift a chosen set. Then, as described in the previous section, the resulting sets are allocated to relevant branches. if the condition $(t+1) < |\mathcal{A}|$ is not satisfied, then the NODE property $isLeaf$ flag become true ($\top$) and they are allocated as unicast messages. This way we can differentiate between leaf nodes and inner nodes for later use.

Algorithm 3 Tree construction (Phase 2)

Require: $K, L, t \in \mathbb{N}$
Ensure: $\mathcal{T}$: the set of trees

struct NODE2
  $value$: ARRAY $\vee$ INT
  $parent$: NODE
  $children$: ARRAY
  $left$: ARRAY
  $isLeaf$: BOOLEAN
end struct
$\mathcal{D} \leftarrow \mathcal{T}[i].children$
for $D \in \mathcal{D}$ **do**
  $\mathcal{A} \leftarrow D.left$
  if $(t+1) < |\mathcal{A}|$ **then**
    for $i \in [|\mathcal{A}|]$ **do**
      $\mathcal{P} \leftarrow$ ROLL$(\mathcal{A}, i-1)$
      $N \leftarrow$ Node $(\mathcal{Q}, D, \varnothing, \mathcal{A} \setminus \mathcal{P}, \perp)$
      $D.children \leftarrow D.children \cup \{N\}$
  else
    $N \leftarrow$ Node $(\{\mathcal{A}\}, D, \varnothing, \varnothing, \top)$
    $D.children \leftarrow D.children \cup \{N\}$

5.1.2 Building Transmission Vectors from the Tree Diagram

In a multiple access channel, users must decode multiple messages at once [2], but in the newly proposed scheme, each user only receives and decodes one message per transmission block. There happens to be no overlap among the codewords and this fact is exhibited by the construction of transmission vectors from the tree diagram. The generalized tree diagram discussed above is used to select arbitrary multicast and unicast message symbols from Kth user, for simplicity, let's assume $L = 4$, $K = 5$, $N = 5$, $M = 1$ and the demand set is A, B, C, D, E. Hence, according to (4.4) and (4.5), we can calculate $N_m = 2$ and $N_{mod} = 1$. consequently, there are 2 multicasting user groups (2 stages in the tree diagram) and there is one unicast user-index at the leaf of the tree diagram. Now the first transmission vector is constructed using the initial branch (root $\longrightarrow$ leaf) of the 1st user's tree diagram is

$$
\begin{aligned}
{}^{1}X^{1}_{1,1} &= \mathbf{u}_{3,4,5}(A_2 \oplus B_1) + \mathbf{u}_{1,2,5}(C_4 \oplus D_3) + \mathbf{u}_{1,3,4}E_1,\\
{}^{1}X^{1}_{1,2} &= \mathbf{u}_{3,4,5}(A_2 \oplus B_1) + \mathbf{u}_{1,2,5}(C_4 \oplus D_3) + \mathbf{u}_{2,3,4}E_2,\\
{}^{1}X^{1}_{1,3} &= \mathbf{u}_{3,4,5}(A_2 \oplus B_1) + \mathbf{u}_{1,2,5}(C_4 \oplus D_3) + \mathbf{u}_{3,1,2}E_3,\\
{}^{1}X^{1}_{1,4} &= \mathbf{u}_{3,4,5}(A_2 \oplus B_1) + \mathbf{u}_{1,2,5}(C_4 \oplus D_3) + \mathbf{u}_{4,1,2}E_4.
\end{aligned}
\tag{5.5}
$$

Here u denotes the beamforming vector, and brackets of the interference indicator sets are dropped for notation simplicity. Unicast messages are constructed such that the same number of data to all the users are delivered. According to (5.5), $Nm(t+1) = 4$ number of transmission vectors having the same set of multicast messages but different unicasts are delivered to the same set of users. This is, to ensure the symmetricity of the system. However, unicast message indices must be chosen such that when delivering messages to respective users, they should be able to use their cache contents to null out (cache out) the interference.

Selecting indices for interference indicators of multicast should be used constructed according to

$$
i_l \in \mathcal{K} \setminus \left({}^{k}M^{l}_{x}\right), \quad i \subset [K], \quad |i| = L - 1. \tag{5.6}
$$

Finally, a common coded message can be constructed to serve $(t + L)$ users, however, while managing interaction between noise and interference between coded messages, each message only should serve a group of $t + 1 \in (t + L)$ users. Hence, this leads to an increased number of transmissions.

5.1.3 Number of Transmissions

There are k tree diagrams and in a single tree diagram, the first set of branches consists of $\binom{K-1}{t}$ branches. Second and subsequent branches featuring of multicast message indices consists of $\prod_{i=1}^{N_m-1}(k - i(t+1))$ sets of branches having $l(= 1 \dots N_m)$ states. Next, every unicast indices should be repeated $Nm(t+1)$ as explained in the above section. Finally when considering each and every case including $K > t + L$, every transmission discussed should repeat $\binom{K}{t+L}$ times. Therefore, the total number of transmissions in general is

$$
\text{Transmissions} = K\binom{K-1}{t}\Delta N_m(t+1)\binom{K}{t+L}, \tag{5.7}
$$

where $\Delta \equiv \prod_{i=1}^{N_m-1}(k - i(t+1))$. Since re-transmission of messages forces splitting subpackets into smaller parts, number of transmissions directly impacts the subpacketization.

5.1.4 General Subpacketization

In this proposed new scheme, it is assumed that cache content placement is similar to the scheme presented in [12] where each file is split into $\binom{K}{t}$ subfiles. Here due to the increased transmissions, messages had to be further divided into smaller pieces so that the overall number of mini-files were

$$\binom{K}{t}\left(N_m(t+1)^2\Delta \;+\; (N_m-1)N_m(t+1)^2\Theta \;+\; \binom{K-1}{t}(L-1)\right)\binom{K-t-1}{L-1}. \tag{5.8}$$

where $\Delta = \prod_{i=1}^{N_m-1}(k-i(t+1))$ and $\Theta = \prod_{j=1}^{N_m-1}\binom{K-j(t+1)}{t+1}$. This additional splitting is required to enable the transmission of distinct content in each additional time interval introduced by the parameters N_m and N_{mod}. The proof of (5.8) is given in the Appendix 3. In the above discussed *Scenario 1*, (5.8) reduced only to $\binom{K}{t}$ as in [73]. Similarly when $k \leq t+L$ and $N_{mod} = 0$ (i.e *Scenario 2 and 3*) subpacketization is very low compared to the general case of (5.8).

5.2 Beamformer Design Suggestions

It is critical to choose an appropriate beamformer design for the aforementioned scheme that is simple on both the transmitter and receiver sides. One candidate is the ZF beamformer, which completely eliminates interference from unwanted users and optimally allocates the power among the parallel streams. However, it is a well-known fact that ZF heavily degrades performance in the low SNR regime [2]. Instead, optimized beamformers allow some interference so that the effects of inter-stream interference and noise are balanced. Furthermore, they allow for an interplay between canceling out and controlling interference, significantly improving performance at low SNR [13]. Hence optimized beamformers would be the preferred choice. However, due to interference from unwanted terms, optimized beamformers may necessitate solving non-convex optimization problems, making the problem computationally complex even for moderate K values. Interestingly, this proposed new scheme manages to eliminate the need for MAC decoding, making it possible to design optimized beamformers with a lot less computational effort.

As discussed in the aforementioned sections, transmission vectors are constructed using tree diagrams. In each time slot, all $(t+L)$ users are served with Nm number of parallel multicast streams and N_{mod} number of unicast streams. The users in a given multicast group do not experience inter-stream interference from multicast streams, but rather, all other users in multicasting and unicasting groups do. To manage the inter-stream interference between multicast and unicast streams, the base station must be equipped with at least L antennas. As a result, the design of this new scheme is based primarily on this assumption. Each user decodes a single multicast message during a specific time slot because overlap is not permitted. As a result, unlike in [2], no SIC receiver or multiple-access channel rate region constraints are needed when formulating the problem. As a result, the achievable rate is uniquely defined by the SINR of the received data stream.

We now specify $\gamma_C(i)$ as the symmetric SINR for every user served in time slot i so that $R_C(i) = \log(1+\gamma_C(i))$. For the ith time slot, the multigroup multicast beamformer optimization

problem can be subsequently stated as the following SINR maximization problem:

$$\begin{aligned} \max_{\gamma_C(i),\mathbf{M}w_{\mathcal{T}}} \quad & \gamma_C(i) \\ \text{s. t.} \quad & \gamma_C(i) \leq \frac{|\mathbf{M}h_k^{\mathrm{H}}\mathbf{M}w_{\mathcal{T}}(i)|^2}{|\mathbf{M}h_k^{\mathrm{H}}\mathbf{M}w_{\bar{\mathcal{T}}}(i)|^2+N_0}, \\ & \forall k \in \mathcal{T}, \mathcal{T} \in \mathsf{P}(i), \bar{\mathcal{T}} \in \mathsf{P}(i) \setminus \mathcal{T}, \\ & \textstyle\sum_{\mathcal{T}\in\mathsf{P}(i)} \|\mathbf{M}w_{\mathcal{T}}(i)\|^2 \leq \mathrm{SNR}. \end{aligned} \tag{5.9}$$

where $\mathsf{P}(i)$ depicts the all possible partitionings of S $\{\mathcal{S} \subseteq [K]\}$ into $(t+\beta)$ groups. Here, the parameter $beta$ can regulate the overlap between the multicast messages transmitted in parallel and it specifies the number of parallel messages that should be able to be distinguished from one another by using the SIC receiver.

Beamforming vectors should be designed and optimized separately to maximize the symmetric rate for each transmission interval.Beamforming for multi-group multicast for common SINR maximization is the problem that now arises as a result. There are numerous ways to solve this, including semidefinite relaxation of beamformers and solving them iteratively through bisection as a semidefinite program [75]. Instead, in this thesis, the low-complexity iterative method from [76] is suggested, which is based on multi-stream multi-group multicasting beamforming to efficiently solve the stripped-out beamformer problem.

6 PERFORMANCE OF THE PROPOSED SCHEME

In this chapter, the performance of the two schemes presented in Chapter 3 is compared with the novel proposed scheme. The main factors of interest in this comparison are the complexity, limitations, the number of transmissions, and subpacketization level, which is how many pieces each file need to be divided into. The comparisons are also summarized in Table 6.1 and Table 6.2. The proposed scheme is divided in half for comparative purposes. First, input parameters were restricted to $\frac{t+L}{t+1} \in \mathbb{N}$ and secondly, no restrictions to input parameters were imposed.

6.1 Performance comparison for the restricted parameter case

In this section, input parameters are restricted in the proposed scheme such that $\frac{t+L}{t+1} \in \mathbb{N}$. Most of the recent literature available on coded caching is employed with limited input parameters. Depending on the application and context, this has advantages and disadvantages; for instance, subpacketization of the proposed scheme benefits as the large value in (5.8) is reduced to $\binom{k}{t}$.

In most state-of-the-art approaches such as [2], any user k will need to decode $\binom{t+L-1}{t}$ different messages, which increases exponentially when K, L, N are increased. However, when delivering messages using optimum multigroup multicast beamforming introduces a complex beamformer optimization problem which is significantly more complex than the simple ZF scheme. In contrast, [3] and the proposed new scheme both have low complexity in terms of beamforming design where each user receives at most one single message after every transmission, thus not requiring SIC receivers.

In terms of complexity in implementing the schemes, [3] is the most simple with its graphical representation of cache placement and delivering with unicast messages. Both [2] and the proposed new scheme have to handle the complexity in cache placement similar to [14]. However, the tree diagram in the proposed new scheme is easy to tackle in the case of restricted parameters. At each node, tree diagrams simplify into single branches (avoiding sub-branches) and restricted to one single tree diagram (first user). In this case, like [2], the proposed new scheme also only transmits multicast messages. This is in contrast to [3], which transmits only unicast messages. However, all three schemes achieve the optimal sum-DoF of $(t + L)$, even though [3] sacrifices the multicasting gain.

Table 6.1: Comparison of characteristics of schemes

Characteristics	[2]'s scheme	[3]'s scheme	Proposed scheme
Limitations	$\frac{t+L}{t+1} \in \mathbb{N}$	$t \leq L$	$\frac{t+L}{t+1} \in \mathbb{N}$
subpacketization	$\binom{K}{t}\binom{K-t-1}{\alpha-1}\gamma$	$K(t+L)$	$\binom{K}{t}$
Number of transmissions	$\frac{\binom{K}{t+1}}{N_m}$	$K \times (K-t)$	$\frac{\binom{K}{t+1}}{N_m}$
Decoding	MAC decoding	Single message decoding	Single message decoding

Considering the transmissions in the cyclic caching scheme, the content is delivered after $K(K-t)$ transmissions. When comparing the other two schemes, the number of transmissions in [2] is low, since that scheme combines $\binom{t+L}{t+1}$ coded messages, and that long coded message is sent to $(t + L)$ users. So each user k will have to decode many different messages. In contrast, multicast messages do not overlap in the proposed new scheme, which transmits them

in different time intervals/slots. Table 6.1 summarizes the above discussed characteristics where $\gamma = \frac{(\alpha-1)!}{(\delta-1)!(\beta-1)!(t+\beta)!^{\delta-1}}$.

6.1.1 Numerical example comparing three schemes

Consider a scenario in which a transmitter with $L \geq 3$ antennas must fulfill requests from $K = 8$ users from a library $\mathcal{W} = \{A, B, C, D, E, F, G, H\}$ of size $N = 8$ files each of F bits. Suppose that each user is permitted to cache $M = 1$ files with F bits without being aware of the specific requests beforehand during the placement of the cache content. Here, we have $t \triangleq KM/N = 1$.

In the cache placement phases of [2] and proposed new scheme, first, each file is divided into non-overlapping equal-sized subfiles according to the strategy as in [12]. For a fair comparison, it is assumed $\beta = 1$ in [2] (to avoid the multiple access channel similar to our scheme). This necessitates splitting subpackets into smaller min files, and hence, the subpacketization of [2] would be

$$\delta = \frac{t+L}{t+\beta} = 2, \quad \binom{K}{t}\binom{K-t-1}{L-1}\frac{(L-1)!}{(\delta-1)!(\beta-1)!(t+\beta)!^{\delta-1}} = 60. \tag{6.1}$$

However, in this special case, new scheme's subpacketization is similar to [12], i.e.,

$$\binom{K}{t} = 8, \tag{6.2}$$

as in Scenario 1 (cf. Chapter 4). Finally, the subpacketization of the cyclic caching sceheme is calculated as

$$K \times (t+L) = 32. \tag{6.3}$$

From the subpacketization results, it is safe to say that the proposed new scheme needs considerably low subpacketization when compared to both schemes reported in [2] and [3]. This superior performance in this special case was highlighted by the work of [73] as well.

In the content delivery, we assume each user requests single file from library. Considering the number of transmissions, both the proposed new scheme and the scheme in [2] have the same number of transmissions since in the latter, $\beta = 1$. Here, we need to serve $\binom{8}{2} = 28$ groups of users in each time block. Since $N_m = 2$, two groups are served at each time block, and a total number of $28/2 = 14$ transmission blocks are needed to complete the communication. However, cyclic caching, which uses an entirely different strategy, has $K \times (K-t) = 56$ transmissions in this case. Moreover, all three schemes in this setting remove the requirement of MAC decoding, thus eliminating the necessity of complex receiver structures such as SIC.

6.2 Comparison of performance without parameter restrictions

A special characteristic of the proposed new scheme is its ability to work with every parameter without any restrictions. In contrast both [2] and [3] restrict their input parameters to satisfy $\frac{t+L}{t+1} \in \mathbb{N}$ and $t \leq L$, respectively. Due to the elimination of SIC, the decoding complexity of the two other schemes is significantly less than that of [2], as discussed previously.

For any given values of K, L, and t, the proposed new scheme has several multicast messages that serve some non-overlapping groups of users and some unicast messages serving single users,

which are readily decodable using respective cache memories. This technique, which is referred to as the signal level approach, eliminates the interference using the cache contents of respective users before the signal is decoded at the receiver [77]. However, implementing SIC in signal level can be even more complicated to implement because of signal level reconstruction and subtraction [78]. Moreover, in order to guarantee symmetricity of the scheme while removing all the interfering segments, the number of transmissions has to be increased considerably. This can be observed in (5.7). Compared to the proposed new scheme, both other schemes have considerably less number of transmissions in this general case.

General subpacketization of the proposed new scheme is exponential and it gets more complicated with the number of multicast message groups (N_m). This is a direct impact of the above discussed increased transmissions. Cyclic caching is superior in this regard but it has the limitation of input parameters $t \leq L$. Since [2] is limited to $\frac{t+L}{t+1} \in \mathbb{N}$, comparisons are only limited to multiserver scheme and cyclic caching scheme. It is therefore clear that our focus is not limited to a specific range of system parameters unlike in both [2] and [3] schemes but as a result in number of transmissions and subpacketization have increased exponentially. Table 6.2 summarizes the above discussed characteristics where $\gamma = \frac{(\alpha-1)!}{(\delta-1)!(\beta-1)!(t+\beta)!^{\delta-1}}$ and $\Gamma = (N_m(t+1)^2\Delta) \; + \; (N_m - 1)N_m(t+1)^2\Theta \; + \; \binom{K-1}{t}(L-1)$.

Table 6.2: Comparison of characteristics of schemes

Characteristics	[2]'s scheme	[3]'s scheme	Proposed scheme
Limitations	$\frac{t+\alpha}{t+\beta} \in \mathbb{N}$	$t \leq L$	No Limitations
subpacketization	$\binom{K}{t}\binom{K-t-1}{\alpha-1}\gamma$	$K(t+L)$	$\binom{K}{t}\binom{K-t-1}{\alpha-1}\Gamma$
Number of Transmissions	$\binom{K}{t+L}$	$K \times (K-t)$	$K\binom{K-1}{t}\Delta N_m(t+1)\binom{K}{t+L}$
Decoding	MAC decoding	Single message decoding	Single message decoding

6.2.1 Numerical Example Comparing Three Schemes, $N_{mod} > 0$

Consider a scenario in which a transmitter with $L \geq 5$ antennas must fulfill requests from $K = 7$ users from a library $\mathcal{W} = \{A, B, C, D, E, F, G\}$ of size $N = 7$ files each of F bits. Suppose that each user is permitted to cache $M = 2$ files with F bits without being aware of the specific requests beforehand during the placement of the cache content. It can be seen that here $t \triangleq KM/N = 2$, and hence, $(t + L)$ is not divisible by $(t + 1)$; consequently, the scheme in [2] cannot be compared. Instead, the multiserver scheme in [1] is considered in the below comparisons.

In the cache placement phases of [1] and the proposed new scheme, each file is divided into non-overlapping equal-sized subfiles according to the strategy as in [12]. Then, further spliiting of files is needed, and hence, the subpacketization of [1] would be

$$\binom{K}{t}\binom{K-t-1}{L-1} = 21. \tag{6.4}$$

Following this, the subpacketization of the proposed new scheme is computed according to (5.8). For this, N_m and N_{mod} is calculated using (4.4) and (4.5), resulting in 2 and 1, respectively.

After substituting these values, (5.8) simplifies to

$$\binom{K}{t}\left(2(t+1)^2\Delta) \;+\; 2(t+1)^2\Theta \;+\; \binom{K-1}{t}(L-1)\right)\binom{K-t-1}{L-1}, \tag{6.5}$$

where $\Delta = (k-(t+1))$ and $\Theta = \binom{K-(t+1)}{t+1}$. When numerical values are substituted, subpacketization of proposed new scheme is summed up to 3360. Despite that, subpacketization of cyclic caching is calculated similar to the previous example and that scheme needed subpacketization of 49. In the content delivery phase, we assume that each user requests a single library file. Hence when considering the number of transmissions, [1] needs only one time slot to finish the transmission while the proposed new scheme needs 2520 time slots to finish all the transmissions. Also, cyclic caching requires $K \times (K-t) = 35$ number of transmissions.

7 CONCLUSIONS AND FUTURE WORK

7.1 Summary and Conclusions

Although the early theoretical claims of significant caching gains, coded caching is severely constrained by bottlenecks, which considerably lower these gains. Some of these bottlenecks are requiring complex SIC at the receiver, exponential increase in subpacketization, applicability to a limited range of input parameters, and performance losses in low- and mid- SNR regimes. It was highlighted in the literature survey of the thesis, most of the recent studies available on coded caching have limited input parameters and their scalability is very limited. However, in forthcoming generations of cellular networks, it is beyond question that the appeal of cache-aided communication is greatest in networks with a lot of users and antennas. Hence a novel coded caching scheme is required to reduce the complexity at both the transmitter and receiver which also works with every practical input parameter.

In this work, we proposed a novel coded caching scheme for cache-aided MISO networks which works with every practical input parameter of K, L and t. For a fixed t, proposed new scheme achieves theoretical sum-DoF optimality with no limitations. Precisely, DoF of $(t+L)$ is achieved by employing several multicast messages that will serve some non-overlapping groups of users and some unicast messages serving a single user which can be readily decodable using respective cache memories. One particular significance, in this scheme is when the transmission vectors are built in the delivery stage no overlap is allowed among user groups, only serving each user one message at a time. Therefore, neither SIC receiver nor multiple-access channel rate region constraints are needed in the problem formulation. Hence proposed new method has reduced complexity in terms of beamformer optimization problem and receiver complexity where it enjoys a linear receiver implementation.

Proposed new scheme could achieve a significant less subpacketization level compared to [2], when $\frac{t+L}{t+1} \in \mathbb{N}$, since new scheme has to only group sub-files accordingly in the delivery phase without any need for further splitting. Consequently, this scheme in this special case, could reduce the complexity of multicast beamforming significantly. However, when catering all other parameters ($N_{\text{mod}} > 0$), in order to guarantee symmetry of the scheme, unicast messages were constructed such a way that same number of data to all the the users are delivered. As a result, overall number of transmissions increased considerably leading to further spliting of sub-files into mini-files at the delivery stage. As a result, it was observed in the numerical comparison results, when input parameters do not adhere to $\frac{t+L}{t+1} \in \mathbb{N}$, performance of the proposed new scheme degraded significantly.

Overall, objective of this topic was achieved by designing a new coded caching scheme which could transmit m_k coded messages simultaneously using multiple antennas at any given time, where each message only serves a group of $t+1$ ($\in (t+L)$). Moreover, this new group assignment scheme could overcome the challenge of managing interaction between noise and interference between coded messages while transmission vectors are designed. Since no overlap is allowed among user groups served by multiple multicast messages transmitted in parallel, each user decodes a single multicast message making the multicast beamforming for common SINR maximization optimization problem simpler. As a consequence, the optimization problem could be solved in an iterative manner that quickly converges to an optimal solution while requiring less computational power. At the latter part of this study this optimization problem is suggested to solve using a special case in multi-stream multi-group multicasting beamforming study in [76].

7.2 Future Work

Current scope of this thesis only covered up-to the formulation of the new coded caching scheme, nevertheless, formulating the beamformer problem according to a special case in [76] and solving the problem efficiently will be the next immediate steps of this study. Although this work of new group assignment scheme shows superior performance when $\frac{t+L}{t+1} \in \mathbb{N}$, on going research work mainly focuses on finding a solution for the degradation of performance when $N_{mod} > 0$. Moreover, in this scheme the interference is removed using the cache contents before the signal is decoded at the receiver. However, it remains an open question to investigate possible performance improvements and complexity reduction in signal-level receivers over classical bit-level approach.

8 BIBLIOGRAPHY

[1] Shariatpanahi S.P., Motahari S.A. & Khalaj B.H. (2016) Multi-server coded caching. IEEE Transactions on Information Theory 62, pp. 7253–7271.

[2] Tölli A., Shariatpanahi S.P., Kaleva J. & Khalaj B.H. (2020) Multi-antenna interference management for coded caching. IEEE Transactions on Wireless Communications 19, pp. 2091–2106.

[3] Salehi M.J., Parrinello E., Shariatpanahi S.P., Elia P. & Tölli A. (2020) Low-complexity high-performance cyclic caching for large MISO systems. CoRR abs/2009.12231.

[4] Bogdan-Martin D. (2020) Measuring digital development facts and figures. Internatonal Telecommunicaton Union CH-1211 Geneva Switzerland.

[5] GSMAIntelligence (2022) The mobile economy. GSMA CH-1211 Geneva Switzerland.

[6] Uusitalo M.A., Rugeland P., Boldi M.R., Strinati E.C., Demestichas P., Ericson M., Fettweis G.P., Filippou M.C., Gati A., Hamon M.H., Hoffmann M., Latva-Aho M., Pärssinen A., Richerzhagen B., Schotten H., Svensson T., Wikström G., Wymeersch H., Ziegler V. & Zou Y. (2021) 6G vision, value, use cases and technologies from european 6G flagship project hexa-x. IEEE Access 9, pp. 160004–160020.

[7] Li J., Niu Y., Wu H., Ai B., Chen S., Feng Z., Zhong Z. & Wang N. (2022) Mobility support for millimeter wave communications: Opportunities and challenges .

[8] Bogdan D. & Martin (2022) Conviva's state of streaming. CoRR USA Califonia.

[9] Hasegawa H., Kouno S., Shiozu A., Sasaki M. & Shimogawa S. (01 2013) Predictive network traffic engineering for streaming video service. pp. 788–791.

[10] Zhao Y., Wat P., Laser M.S. & Medvidovic N. (2018) Empirically assessing opportunities for prefetching and caching in mobile apps. CoRR abs/1810.08861.

[11] Paschos G.S., Bastug E., Land I., Caire G. & Debbah M. (2016) Wireless caching: Technical misconceptions and business barriers. CoRR abs/1602.00173.

[12] Maddah-Ali M.A. & Niesen U. (2014) Fundamental limits of caching. IEEE Transactions on Information Theory 60, pp. 2856–2867.

[13] Tölli A., Shariatpanahi S.P., Kaleva J. & Khalaj B. (2018) Multicast beamformer design for coded caching. In: *IEEE International Symposium on Information Theory - Proceedings*, IEEE, volume 2018-June, pp. 1914–1918.

[14] Shariatpanahi S.P., Motahari S.A. & Khalaj B.H. (2016) Multi-server coded caching. IEEE Transactions on Information Theory 62, pp. 7253–7271.

[15] Yang H.H., Geraci G. & Quek T.Q.S. (2015) Energy-efficient design of MIMO heterogeneous networks with wireles backhaul. CoRR abs/1509.05506.

[16] Dhillon H.S. & Caire G. (2014) Wireless backhaul networks: Capacity bound, scalability analysis and design guidelines. CoRR abs/1406.2738.

[17] Rajatheva N., Atzeni I., Bjornson E., Bourdoux A., Buzzi S., Dore J.B., Erkucuk S., Fuentes M., Guan K., Hu Y., Huang X., Hulkkonen J., Jornet J.M., Katz M., Nilsson R., Panayirci E., Rabie K., Rajapaksha N., Salehi M., Sarieddeen H., Svensson T., Tervo O., Tolli A., Wu Q. & Xu W. (2020), White paper on broadband connectivity in 6G.

[18] Alves H., Riihonen T. & Suraweera H.A. (2020) Full-Duplex Communications for Future Wireless Networks. Springer.

[19] Ding Z., Lei X., Karagiannidis G.K., Schober R., Yuan J. & Bhargava V.K. (2017) A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends. CoRR abs/1706.05347.

[20] Amjad M., Rehmani M.H. & Mao S. (2018) Wireless multimedia cognitive radio networks: A comprehensive survey. IEEE Communications Surveys Tutorials 20, pp. 1056–1103.

[21] Du Q. & Zhang X. (2011) QoS-aware base-station selections for distributed mimo links in broadband wireless networks. IEEE Journal on Selected Areas in Communications 29, pp. 1123–1138.

[22] Chen H. & Xiao Y. (06 2006) Cache access and replacement for future wireless internet. Communications Magazine, IEEE 44, pp. 113 – 123.

[23] Liu D., Chen B., Yang C. & Molisch A.F. (2018) Caching at the wireless edge: Design aspects, challenges and future directions. CoRR abs/1810.13287.

[24] Chao F., Richard Y.F., Tao H., Jiang L. & Yunjie L. (2014) A game theoretic approach for energy-efficient in-network caching in content-centric networks. China Communications 11, pp. 135–145.

[25] Breslau L., Cao P., Fan L., Phillips G. & Shenker S. (1999) Web caching and zipf-like distributions: evidence and implications. In: *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, volume 1, pp. 126–134 vol.1.

[26] Korupolu M. & Dahlin M. (2002) Coordinated placement and replacement for large-scale distributed caches. IEEE Transactions on Knowledge and Data Engineering 14, pp. 1317–1329.

[27] Baev I. & Rajaraman R. (01 2001) Approximation algorithms for data placement in arbitrary networks. pp. 661–670.

[28] Krishnan P., Raz D. & Shavitt Y. (2000) The cache location problem. IEEE/ACM Transactions on Networking 8, pp. 568–582.

[29] Zheng G., Suraweera H.A. & Krikidis I. (2017) Optimization of hybrid cache placement for collaborative relaying. IEEE Communications Letters 21, pp. 442–445.

[30] Borst S., Gupta V. & Walid A. (2010) Distributed caching algorithms for content distribution networks. In: *2010 Proceedings IEEE INFOCOM*, pp. 1–9.

[31] Niesen U. & Maddah-Ali M.A. (2017) Coded caching with nonuniform demands. IEEE Transactions on Information Theory 63, pp. 1146–1158.

[32] Zhang Q., Zheng L., Cheng M. & Chen Q. (2020) On the dynamic centralized coded caching design. IEEE Transactions on Communications 68, pp. 2118–2128.

[33] Al-Shehri S.M., Loskot P., Numanoglu T. & Mert M. (2017) Common metrics for analyzing, developing and managing telecommunication networks. CoRR abs/1707.03290.

[34] Maddah-Ali M.A. & Niesen U. (2019) Cache-aided interference channels. IEEE Transactions on Information Theory 65, pp. 1714–1724.

[35] Sengupta A., Tandon R. & Simeone O. (2016) Cache aided wireless networks: Tradeoffs between storage and latency. In: *2016 Annual Conference on Information Science and Systems (CISS)*, pp. 320–325.

[36] Wang C.Y., Lim S.H. & Gastpar M. (2016) A new converse bound for coded caching. In: *2016 Information Theory and Applications Workshop (ITA)*, pp. 1–6.

[37] Wang C.Y., Bidokhti S.S. & Wigger M. (2017) Improved converses and gap-results for coded caching. In: *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 2428–2432.

[38] Zhang J., Lin X. & Wang X. (2018) Coded caching under arbitrary popularity distributions. IEEE Transactions on Information Theory 64, pp. 349–366.

[39] Maddah-Ali M.A. & Niesen U. (2015) Decentralized coded caching attains order-optimal memory-rate tradeoff. IEEE/ACM Transactions on Networking 23, pp. 1029–1040.

[40] Shanmugam K., Ji M., Tulino A.M., Llorca J. & Dimakis. A.G. (2016) Finite-length analysis of caching-aided coded multicasting. IEEE Transactions on Information Theory 62, pp. 5524–5537.

[41] Jin S., Cui Y., Liu H. & Caire G. (2016) Order-optimal decentralized coded caching schemes with good performance in finite file size regime. In: *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7.

[42] Wang J., Cheng M., Yan Q. & Tang X. (2019) Placement delivery array design for coded caching scheme in d2d networks. IEEE Transactions on Communications 67, pp. 3388–3395.

[43] Yan Q., Wigger M.A. & Yang S. (2018) Placement delivery array design for combination networks with edge caching. CoRR abs/1801.03048.

[44] Asadi B. & Ong L. (2019) Centralized caching with shared caches in heterogeneous cellular networks. In: *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5.

[45] Parrinello E., Ünsal A. & Elia P. (2020) Fundamental limits of coded caching with multiple antennas, shared caches and uncoded prefetching. IEEE Transactions on Information Theory 66, pp. 2252–2268.

[46] Shariatpanahi S.P., Caire G. & Khalaj B.H. (2017) Multi-antenna coded caching pp. 2113–2117.

[47] Salehi M.J., Tolli A. & Shariatpanahi S.P. (2020) Subpacketization-beamformer interaction in multi-antenna coded caching. In: *2nd 6G Wireless Summit 2020: Gain Edge for the 6G Era, 6G SUMMIT 2020*, pp. 1–5.

[48] Bergel I. & Mohajer S. (2018) Cache aided communications with multiple antennas at finite SNR. CoRR abs/1808.02780.

[49] Naderializadeh N., Maddah-Ali M.A. & Avestimehr A.S. (2017) Fundamental Limits of Cache-Aided Interference Management. IEEE Transactions on Information Theory 63, pp. 3092–3107.

[50] Ngo K.H., Yang S., Kobayashi M. & Huang K. (2016) On the complementary roles of massive mimo and coded caching for content delivery. In: *2016 International Conference on Advanced Technologies for Communications (ATC)*, pp. 237–242.

[51] Peng X., Shen J.C., Zhang J. & Letaief K.B. (2014) Joint data assignment and beamforming for backhaul limited caching networks. In: *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pp. 1370–1374.

[52] Tao M., Chen E., Zhou H. & Yu W. (2016) Content-centric sparse multicast beamforming for cache-enabled cloud ran. IEEE Transactions on Wireless Communications 15, pp. 6118–6131.

[53] Liu A. & Lau V.K.N. (2013) Mixed-timescale precoding and cache control in cached mimo interference network. IEEE Transactions on Signal Processing 61, pp. 6320–6332.

[54] Baştuğ E., Bennis M. & Debbah M. (2014) Cache-enabled small cell networks: Modeling and tradeoffs. In: *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, pp. 649–653.

[55] Yang C., Yao Y., Chen Z. & Xia B. (2016) Analysis on cache-enabled wireless heterogeneous networks. IEEE Transactions on Wireless Communications 15, pp. 131–145.

[56] Chen Z., Lee J., Quek T.Q.S. & Kountouris M. (2017) Cooperative caching and transmission design in cluster-centric small cell networks. IEEE Transactions on Wireless Communications 16, pp. 3401–3415.

[57] Lampiris E. & Elia P. (2019) Bridging two extremes: Multi-antenna coded caching with reduced subpacketization and CSIT. In: *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC*, IEEE, volume 2019-July.

[58] Zhao J., Amiri M.M. & Gündüz D. (2020) Multi-antenna coded content delivery with caching: A low-complexity solution. arXiv preprint arXiv:2001.01255 .

[59] Bergel I. & Mohajer S. (2018) Cache-aided communications with multiple antennas at finite SNR. IEEE Journal on Selected Areas in Communications 36, pp. 1682–1691.

[60] Huang W., Huang Y., He S. & Yang L. (2020) Cloud and edge multicast beamforming for cache-enabled ultra-dense networks. IEEE Transactions on Vehicular Technology 69, pp. 3481–3485.

[61] Yan Q., Cheng M., Tang X. & Chen Q. (2017) On the placement delivery array design for centralized coded caching scheme. IEEE Transactions on Information Theory 63, pp. 5821–5833.

[62] Yan Q., Tang X., Chen Q. & Cheng M. (2018) Placement delivery array design through strong edge coloring of bipartite graphs. IEEE Communications Letters 22, pp. 236–239.

[63] Wang J., Cheng M., Yan Q. & Tang X. (2019) Placement delivery array design for coded caching scheme in D2D networks. IEEE Transactions on Communications 67, pp. 3388–3395.

[64] Cheng M., Wang J., Zhong X. & Wang Q. (2021) A framework of constructing placement delivery arrays for centralized coded caching. IEEE Transactions on Information Theory 67, pp. 7121–7131.

[65] Yang T., Wan K., Cheng M. & Caire G. (2022) Multiple-antenna placement delivery array for cache-aided MISO systems. CoRR abs/2201.11462.

[66] Shangguan C., Zhang Y. & Ge G. (2018) Centralized coded caching schemes: A hypergraph theoretical approach. IEEE Transactions on Information Theory 64, pp. 5755–5766.

[67] Shanmugam K., Tulino A.M. & Dimakis A.G. (2017) Coded caching with linear subpacketization is possible using Ruzsa-Szeméredi graphs. In: *IEEE International Symposium on Information Theory - Proceedings*, IEEE, pp. 1237–1241.

[68] Mahesh A.A. & Rajan B.S. (2020) A coded caching scheme with linear sub-packetization and its application to multi-access coded caching. arXiv preprint arXiv:2009.10923 .

[69] Lampiris E., Zhang J., Simeone O. & Elia P. (2019) Fundamental limits of wireless caching under uneven-capacity channels. arXiv preprint arXiv:1908.04036 .

[70] Parrinello E., Ünsal A. & Elia P. (2020) Fundamental limits of coded caching with multiple antennas, shared caches and uncoded prefetching. IEEE Transactions on Information Theory 66, pp. 2252–2268.

[71] Salehi M., Tolli A., Shariatpanahi S.P. & Kaleva J. (2019) Subpacketization-rate tradeoff in multi-antenna coded caching. In: *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*, IEEE, pp. 1–6.

[72] Lampiris E., Elia P. & Caire G. (2019) Bridging the gap between multiplexing and diversity in finite SNR multiple antenna coded caching. In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, IEEE, pp. 1272–1277.

[73] Mohajer S. & Bergel I. (2020) MISO Cache-Aided Communication with Reduced Subpacketization. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, pp. 1–6.

[74] Naderializadeh N., Maddah-Ali M.A. & Avestimehr A.S. (2019) Cache-aided interference management in wireless cellular networks. IEEE Transactions on Communications 67, pp. 3376–3387.

[75] Karipidis E., Sidiropoulos N.D. & Luo Z.Q. (2008) Quality of service and max-min fair transmit beamforming to multiple cochannel multicast groups. IEEE Transactions on Signal Processing 56, pp. 1268–1279.

[76] Mahmoodi H.B., Gouda B., Salehi M.J. & Tölli A. (2021) Low-complexity multicast beamforming for multi-stream multi-group communications. CoRR abs/2105.09705.

[77] Salehi M.J., Mahmoodi H.B. & Tölli A. (2021) A low-subpacketization high-performance MIMO coded caching scheme. CoRR abs/2109.10008.

[78] Su X., Yu H., Kim W., Choi C. & Choi D. (09 2016) Interference cancellation for non-orthogonal multiple access used in future wireless mobile networks. EURASIP Journal on Wireless Communications and Networking 2016, pp. 231.

8.1 Appendix 1

8.1.1 Circulant Matrices

According to linear algebra, a circulant matrix could be defined as a square matrix in which all row vectors are composed of the same elements and each row vector is rotated one element to the right relative to the preceding row vector, particularly a kind of Toeplitz matrix. Consider circulant matrices $\boldsymbol{L}$

$$\boldsymbol{L} = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \cdots & a_{n-3} & a_{n-2} \\ a_{n-2} & a_{n-1} & a_0 & \cdots & & a_{n-3} \\ \cdots & & & \ddots & & \\ a_2 & \cdots & & a_{n-1} & a_0 & a_1 \\ a_1 & a_2 & \cdots & & a_{n-1} & a_0 \end{pmatrix}.$$

circulant matrix could be defined based on different sources in different ways,for instance as above, or with the vector c corresponding to the first row rather than the first column of the matrix; and possibly with a different direction of shift (which is sometimes called an anti-circulant matrix).

In numerical analysis, circulant matrices are important because they are diagonalized by a discrete Fourier transform,hence linear equations that contain them may be quickly solved using a fast Fourier transform. That could be interpreted analytically as the integral kernel of a convolution operator on the cyclic group C_n and hence frequently appear in formal descriptions of spatially invariant linear operations. This property is also critical in modern software defined radios, which utilize Orthogonal Frequency Division Multiplexing to spread the symbols (bits) using a cyclic prefix. This enables the channel to be represented by a circulant matrix, simplifying channel equalization in the frequency domain.

8.1.1.1 Symmetric Circulant Matrices

For a symmetric circulant matrix C one has the extra condition that $c_{n-i} = c_i$. Thus it is determined by $\lfloor n/2 \rfloor + 1$ elements.

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_1 & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_2 & & \ddots & \ddots & c_1 \\ c_1 & c_2 & \cdots & c_1 & c_0 \end{bmatrix}.$$

The eigenvalues of any real symmetric circulant matrix are real. The corresponding eigenvalues become:

$$\lambda_j = c_0 + 2c_1 \mathrm{Re}\omega_j + 2c_2 \mathrm{Re}\omega_j^2 + \cdots + 2c_{n/2-1} \mathrm{Re}\omega_j^{n/2-1} + c_{n/2}\omega_j^{n/2}$$

- Symmetric circulant matrices belong to the class of bisymmetric matrices.

- The sum of elements in each row and column of a symmetric circulant matrix is the same.
- A linear combination of symmetric circulant matrices is a symmetric circulant matrix.
- The inverse of a symmetric circulant matrix is a symmetric circulant matrix.
- The product of symmetric circulant matrices is a symmetric circulant matrix.

8.2 Appendix 2

The remainder of the example extends in Scenario 2 and is shown below.

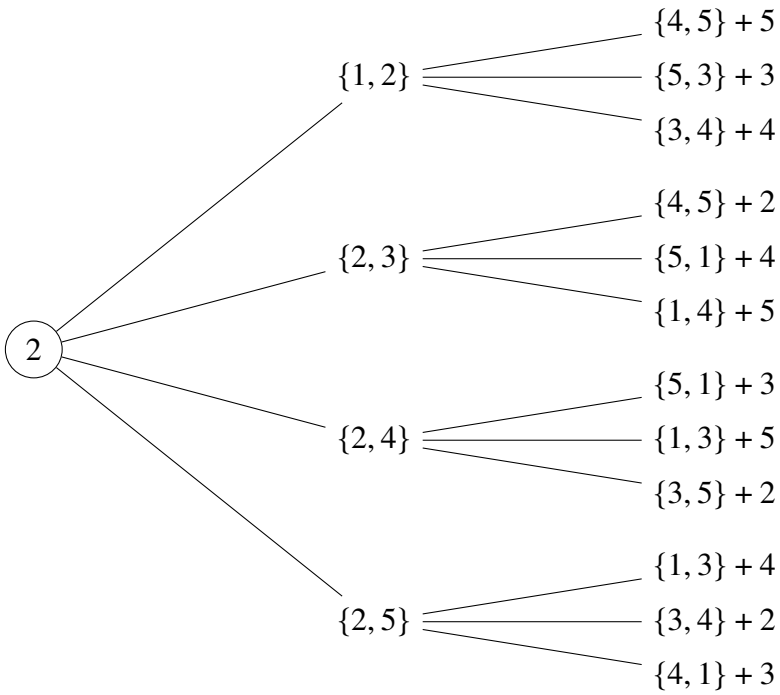


Figure 8.1: Tree diagram prioritizing the second user

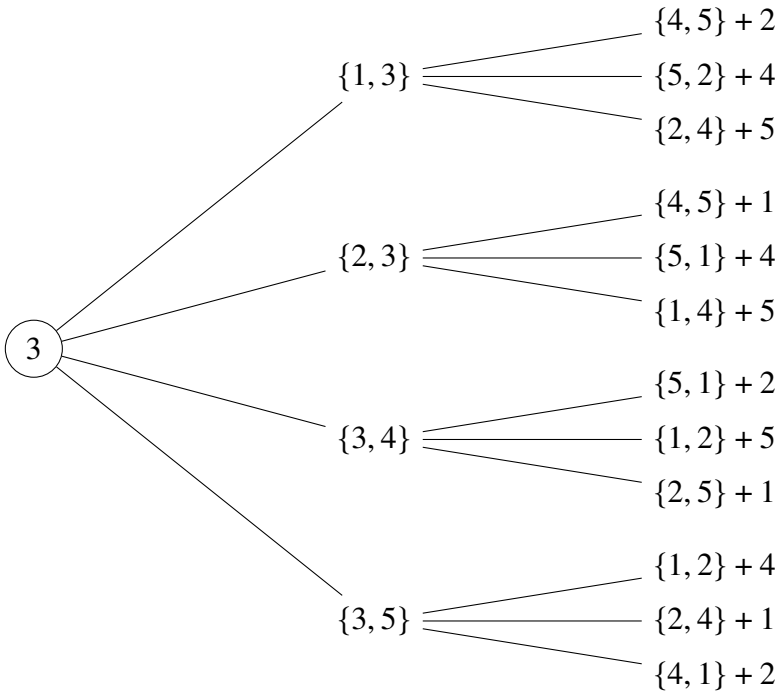


Figure 8.2: Tree diagram prioritizing the third user

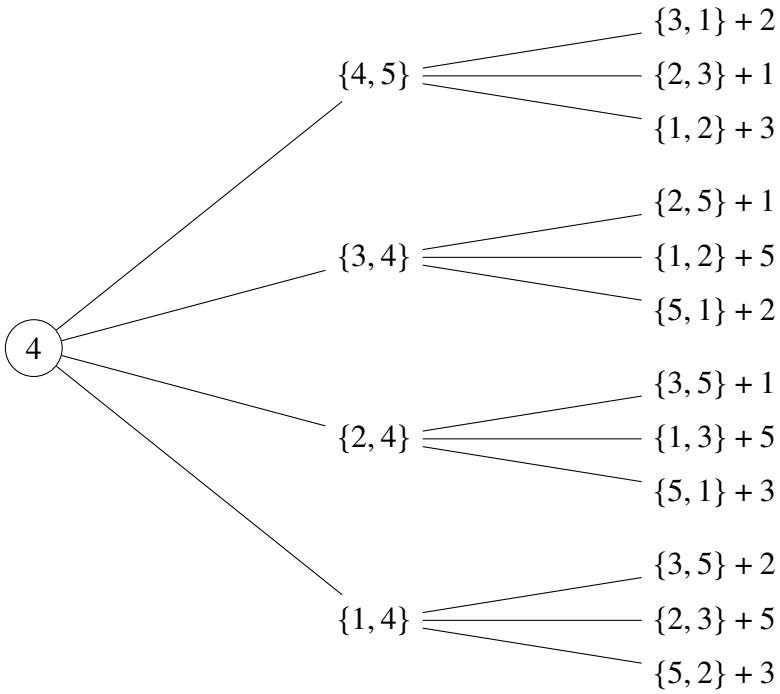


Figure 8.3: Tree diagram prioritizing the fourth user

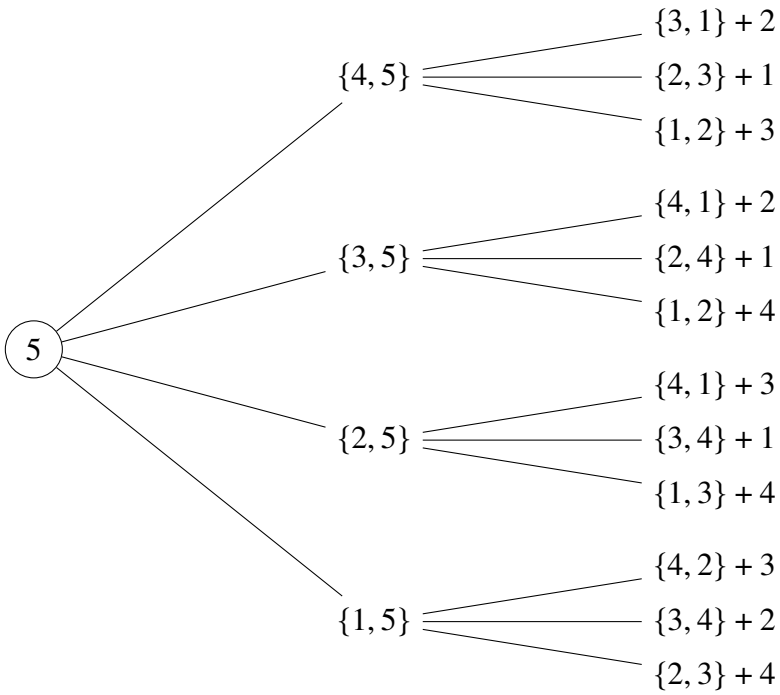


Figure 8.4: Tree diagram prioritizing the fifth user

8.3 Appendix 3

8.3.1 Proof of the Subpacketization Given in (5.8)

Due to no overlap is allowed among user groups served by multiple multicast messages transmitted in parallel each subset appears multiple times, so it is needed to transmit smaller coded messages called coded mini files. Here it is needed to ensure that each coded mini-file provides the targeted users with fresh (not transmitted before) mini files they require. This is the main reason dividing each subfile into more mini files as in (5.8). In the above (5.8) inner and outer loops have exponential functions ($\binom{K}{t}$ and $\binom{K-t-1}{L-1}$) however, due to their symmetric properties, overall symetricity of the system is preserved.

8.3.1.1 Inner and Outer Loops

Inner loop of (5.8) is a result of the basic cache placement strategy of [12] and the outer-loop function (last term of the (5.8)) of this scheme is a result of always choosing to serve only $(t+L)$ users out of all the K users. This resulted in increased transmissions of $\binom{K}{t+L}$ times and as a result each packet is further split into $\binom{K-t-1}{L-1}$ equal-sized subpackets.

8.3.1.2 First Branch of the Tree Diagram

This scheme's structural complexity (tree diagram based) results in further splitting of original messages into mini files. In the inside terms of (5.8), first term,

$$N_m(t+1)^2 \prod_{i=1}^{N_m-1}(k-i(t+1)),$$

one multicast user set, repeat $t+1$ times in different tree diagrams. Next, the same multicast user set, repeat $\prod_{i=1}^{N_m-1}(k-i(t+1))$ times in the second branch and all the other preceding branches. Finally, again multicast user sets again repeat because of duplication of unicast sets; precisely $N_m(t+1)$ times.

8.3.1.3 Second and Every Other Preceding Branch of the Tree Diagram

From the inner terms of (5.8), second term is

$$(N_m-1)N_m(t+1)^2 \prod_{j=1}^{N_m-1}\binom{K-j(t+1)}{t+1},$$

where it represents other (N_m-1) branches in the tree diagram and the first branch is excluded. Multicast user sets in the second branch onward repeat the same set $\prod_{j=1}^{N_m-1}\binom{K-j(t+1)}{t+1}$ times. After that it as the same as above discussed subsection, $N_m(t+1)$ times repeat of multicasting sets happen due to duplication of unicast sets.

After considering all the repeated entries of the same multicasting user sets in multicasting groups of each and every tree diagram, last additive term of (5.8), $\binom{K-1}{t}(L-1)$ is a result of repeating as unicasting user sets in the leaves of the tree diagram.