OPTIMAL NETWORK CODING UNDER SOME LESS-RESTRICTIVE NETWORK MODELS

by

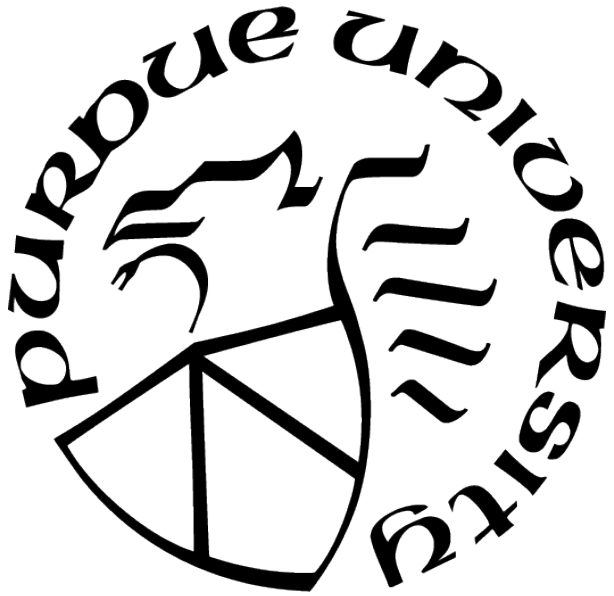
Chih-Hua Chang

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



School of Electrical and Computer Engineering

West Lafayette, Indiana

May 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Chih-Chun Wang, Chair

School of Electrical and Computer Engineering

Dr. Stanley H. Chan

School of Electrical and Computer Engineering

Dr. Xiaojun Lin

School of Electrical and Computer Engineering

Dr. Borja Peleato

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

To my family

ACKNOWLEDGMENTS

This thesis was supported in parts by the National Science Foundation (NSF) under Grant ECCS-1407603, Grant CCF-1422997, Grant CCF-1618475, and Grant CCF-1816013.

TABLE OF CONTENTS

LIST OF TABLES 10

LIST OF FIGURES 13

ABSTRACT 15

1 INTRODUCTION 17

1.1 Outline of Thesis 18

1.2 Main Contributions 19

2 CAPACITY-APPROACHING PROTOCOL FOR GENERAL 1-TO-K BROADCAST PACKET ERASURE CHANNELS WITH ACK/NACK 21

2.1 Introduction 21

2.2 1-to-K Broadcast Packet Erasure Channel Model 27

2.2.1 The 1-to-K Broadcast Packet Erasure Channel 28

2.2.2 The Block-Coding Setting 29

2.2.3 The Sequential-Coding Setting 31

2.3 A New Stability Region 34

2.4 The New Achievability Scheme 38

2.4.1 The Connection to the Virtual Network 38

2.4.2 A New Sequential Network Coding Scheme 41

2.5 Practical Sequential Coding Scheme with Overhead 49

2.5.1 A Header-based Implementation 49

2.5.2 Other Issues for Practical Implementation 52
Unknown Channel Statistic 52
Variable Header Length 52
Delay 53
Deadline 53
2.6 Simulation Results 53
2.7 Summary 57
3 CODED CACHING SYSTEM OF TWO USERS AND TWO FILES 59
3.1 Introduction 59
3.1.1 Comparison to Existing Results 62
3.2 General Coded Caching Model 63
3.3 The Two-User/Two-File Coded Caching Capacity 65
3.3.1 Basic Zero-Error Coded Caching Schemes 66
3.3.2 Lower Bounds of the PRCR 68
3.3.3 Coded Caching Capacity for $N = K = 2$ 71
3.4 Summary 76
4 CODED CACHING SYSTEM FOR TWO USERS WITH HETEROGENEOUS FILE DEMAND SETS 82
4.1 Introduction 82
4.2 Coded Caching Model with File Demand Set 84

4.3 Homogeneous File Demand Sets . . . 86
4.4 Heterogeneous File Demand Sets . . . 87
4.4.1 Disjoint File Demand Sets for arbitrary K . . . 89
4.4.2 One-Overlapping File Demand Sets for $K = 2$ . . . 89
4.4.3 Two-Overlapping File Demand Sets for $N = 3$ and $K = 2$ . . . 91
4.4.4 Large-Overlapping File Demand Sets for $K = 2$ with Selfish and Uncoded Prefetching . . . 94
4.5 Numerical Evaluations . . . 97
4.6 Summary . . . 98
5 CONCLUSION AND FUTURE WORK . . . 101
5.1 Capacity of 1-to-K Spatial-Independent Broadcast Packet Erasure Channels with ACK/NACK . . . 101
5.2 Linear Network Coding for Achieving Capacity of 1-to-K Broadcast Packet Erasure Channels with ACK/NACK . . . 102
5.3 General Lower Bounds for Coded Caching of Arbitrary N Files and K Users 102
5.4 Coded Caching with $K = 2$ user and Heterogeneous User File Popularity . . 103
REFERENCES . . . 104
A SUPPLEMENTARY MATERIALS FOR CHAPTER 2 . . . 111
A.1 Proof of Proposition 2.2.2 . . . 111
A.2 On The Optimality of Proposition 2.3.1 . . . 116
A.2.1 Proof of Corollary 1 . . . 116

A.2.2 Proof of Example 1 . . . 122
A.2.3 An Example of Unachievable Rates for $K = 4$ . . . 124
A.3 Proof of Lemmas 1 and 2 . . . 124
A.4 A Simple Schwartz-Zippel Lemma . . . 126
A.5 Proof of Proposition 2.4.1 . . . 127
A.6 Proof of Propositions 2.4.2 and 2.4.3 . . . 134
A.6.1 Proof of Proposition 2.4.2 . . . 137
A.6.2 Proof of Proposition 2.4.3 . . . 143
A.7 Proof of Lemma 3 . . . 144
B SUPPLEMENTARY MATERIALS FOR CHAPTER 3 . . . 146
B.1 Proof of Proposition 3.3.2 . . . 146
B.2 Proof of Proposition 3.3.3 . . . 185
B.3 Proof of Corollary 3 . . . 199
B.4 Re-derivation of Worst-Case Rate Capacity in [41] . . . 201
C SUPPLEMENTARY MATERIALS FOR CHAPTER 4 . . . 204
C.1 Proof of Proposition 4.4.1 . . . 204
C.2 Proof of Proposition 4.4.2 . . . 204
C.3 Proof of Proposition 4.4.3 . . . 214
C.4 Proof of Proposition 4.4.4 . . . 217
C.5 Proof of Proposition 4.4.5 . . . 218

C.6 Proof of Proposition 4.4.7 and Corollary 6 221

VITA 229

LIST OF TABLES

2.1 Comparison of the numbers of coding choices. COSM stands for "collapsed overhearing set matching" and B_m represents the m-th Bell number. For reference, $B_6 = 203$, $B_{11} = 678570$, and $B_{16} \approx 10^{10}$. 26

2.2 A $K = 3$ example of computing $q(k, S)$ and the backpressure term $\text{bp}(Q_S^{(k)})$ from $\left|Q_S^{(k)}\right|$. 44

3.1 Comparisons of existing results 63

3.2 Basic coded caching schemes for two files of size (f_1, f_2) and two users of memory (m_1, m_2). It is possible to have $f_1 \geq f_2$, or $f_1 < f_2$, or $m_1 \geq m_2$, or $m_1 < m_2$. The intuition of the schemes are Mix.Emp for premixing at d_1, Emp.Mix for premixing at d_2, Ha.Fi for splitting files in halves, and a.b.Cov for covering $\vec{d} = (a, b)$. . . . 66

3.3 The expressions of all 28 possible corner points. 73

B.1 Vertex 1 $(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$ with applicable range: $M_2 \leq M_1 \leq \min(F_1, F_2 + M_2)$. 185

B.2 Vertex 2 $(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$ with applicable range: $M_1 + M_2 \leq \min(F_1, F_2)$. 187

B.3 Vertex 3 $(F_1, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2 - M_2)$ with applicable range: $M_2 \leq M_1 \leq \min(F_1 + M_2, F_2)$. 188

B.4 Vertex 4 $(F_1 - 1/2 M_2, F_1 + F_2 - M_1 - 1/2 M_2, F_1 + F_2 - M_1 - 1/2 M_2, F_2 - 1/2 M_2)$ with applicable range: $M_2 \leq M_1 \leq \min(F_1, F_2)$. 188

B.5 Vertex 5 $(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$ with applicable range: $\max(M_1, M_2) \leq F_2 \leq \min(F_1, M_1 + M_2)$. 189

B.6 Vertex 6 $(F_1, F_1, F_1 + F_2 - M_1 - M_2, F_2)$ with applicable range: $\max(M_1, M_2) \leq F_2 \leq \min(F_1, M_1 + M_2)$. 189

B.7 Vertex 7 $(F_1 + 1/2(F_2 - M_1 - M_2), F_1 + 1/2(F_2 - M_1 - M_2), F_1 + 1/2(F_2 - M_1 - M_2), F_2 + 1/2(F_2 - M_1 - M_2))$ with applicable range: $\max(M_1, M_2) \leq F_2 \leq \min(F_1, M_1 + M_2)$. 189

B.8 Vertex 8 $(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ with applicable range: $F_2 + M_2 \leq M_1 \leq F_1$. 190

B.9 Vertex 9 $(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$ with applicable range: $M_2 \leq F_2 \leq M_1 \leq F_1$. 190

B.10 Vertex 10 $(F_1, F_1 + F_2 - M_1, F_1, F_2 - M_2)$ with applicable range: $M_2 \leq F_2 \leq M_1 \leq F_1 + M_2$. 190

B.11 Vertex 11 $(F_1 - 1/2 M_2, F_1 + F_2 - M_1 - 1/2 M_2, F_1 - 1/2 M_2, F_2 - 1/2 M_2)$ with applicable range: $M_2 \leq F_2 \leq M_1 \leq F_1$. 191

B.12 Vertex 12 $(F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ with applicable range: $\max(M_2, F_2) \leq M_1 \leq \min(F_1, F_2 + M_2)$. 192

B.13 Vertex 13 $(F_1 + 1/2(F_2 - M_1 - M_2), F_1 + 1/2(F_2 - M_1 - M_2), F_1 + 1/2(F_2 - M_1 - M_2), 1/2(F_2 + M_1 - M_2))$ with applicable range: $\max(M_2, F_2) \leq M_1 \leq \min(F_1, F_2 + M_2)$. 192

B.14 Vertex 14 $(F_1 - M_2, F_2, F_1 - M_2, F_2)$ with applicable range: $\max(F_1, F_2 + M_2) \leq M_1 \leq F_1 + F_2$. 192

B.15 Vertex 15 $(F_1, F_2 - M_2, F_1, F_2 - M_2)$ with applicable range: $M_2 \leq F_2 \leq F_1 + M_2 \leq M_1$. 193

B.16 Vertex 16 $(F_1, F_2 - M_2, F_1, F_1 + F_2 - M_1)$ with applicable range: $F_2 - F_1 \leq M_2 \leq F_2$ and $F_1 \leq M_1 \leq F_1 + M_2$. 194

B.17 Vertex 17 $(1/2(F_1 + M_1 - M_2), F_2 + 1/2(F_1 - M_1 - M_2), 1/2(F_1 + M_1 - M_2), F_2 + 1/2(F_1 - M_1 - M_2))$ with applicable range: $M_2 \leq F_2 \leq M_1 \leq F_1 + M_2$ and $F_1 \leq M_1$. 194

B.18 Vertex 18 $(F_1 - M_2, F_2, F_1 + F_2 - M_1, F_2)$ with applicable range: $M_2 \leq F_1 \leq M_1 \leq F_2 + M_2$. 194

B.19 Vertex 19 $(F_1 + F_2 - M_1, F_2, F_1 - M_2, F_2)$ with applicable range: $M_2 \leq F_1 \leq M_1 \leq F_2 + M_2$ and $F_2 \leq M_1$. 195

B.20 Vertex 20 $(F_1 + 1/2(F_2 - M_1 - M_2), 1/2(F_2 + M_1 - M_2), F_1 + 1/2(F_2 - M_1 - M_2), 1/2(F_2 + M_1 - M_2))$ with applicable range: $M_2 \leq F_1 \leq M_1 \leq F_2 + M_2$ and $F_2 \leq M_1$. . . 195

B.21 Vertex 21 $(F_1 + F_2 - M_2, F_1 - M_1, F_1 + F_2 - M_2, F_2)$ with applicable range: $F_2 \leq M_2 \leq M_1 \leq F_1$. 195

B.22 Vertex 22 $(F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0)$ with applicable range: $F_2 \leq M_2 \leq M_1 \leq F_1 + F_2$. 196

B.23 Vertex 23 $(F_1 + 1/2F_2 - M_2, F_1 + 1/2F_2 - M_1, F_1 + 1/2F_2 - M_2, 1/2F_2)$ with applicable range: $F_2 \leq M_2 \leq M_1 \leq F_1$. 196

B.24 Vertex 24 $(F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2, F_1 + F_2 - M_1)$ with applicable range: $F_2 \leq M_2 \leq M_1 \leq F_1 + F_2$ and $F_1 \leq M_1$. 196

B.25 Vertex 25 $(1/2(F_1 + F_2 + M_1) - M_2, 1/2(F_1 + F_2 - M_1), 1/2(F_1 + F_2 + M_1) - M_2, 1/2(F_1 + F_2 - M_1))$ with applicable range: $F_2 \leq M_2 \leq M_1 \leq F_1 + F_2$ and $F_1 \leq M_1$. . . 197

B.26 Vertex 26 $(0, F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2)$ with applicable range: $F_1 \leq M_2 \leq M_1 \leq F_1 + F_2$. 197

B.27 Vertex 27 $(F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2)$ with applicable range: $F_1 \leq M_2 \leq M_1 \leq F_1 + F_2$ and $F_2 \leq M_1$. 197

B.28 Vertex 28 $(1/2(F_1 + F_2 - M_1), 1/2(F_1 + F_2 + M_1) - M_2, 1/2(F_1 + F_2 - M_1), 1/2(F_1 + F_2 + M_1) - M_2)$ with applicable range: $F_1 \leq M_2 \leq M_1 \leq F_1 + F_2$ and $F_2 \leq M_1$. 198

C.1 The optimal $(M_1^c, M_1^u, M_2^c, M_2^u)$ values of the selfish and uncoded prefeching schemes for $\alpha \geq 2$. 228

LIST OF FIGURES

2.1 The 1-to-2 broadcast PEC example of source s and destinations d_1, d_2. The source s intends to delivery packets $X_{1,1}, \ldots, X_{1,nR_1}$ to destination d_1 and packets $X_{2,1}, \ldots, X_{2,nR_2}$ to destination d_2. 23

2.2 The virtual nodes and proper-cuts for $K = 3$ and $k = 2$. 34

2.3 The virtual sub-network for $K = 3$ and $k = 2$, where the virtual nodes are represented by circles and the auxiliary nodes by squares. The dotted edge marked by a set S is actually connected to $\text{vn}_S^{(2)}$ with ∞ capacity. For example, there are 4 outgoing edges of the auxiliary node (square) labeled by $\{1, 3\}$. They are connected to $\text{vn}_\emptyset^{(2)}$, $\text{vn}_{\{1\}}^{(2)}$, $\text{vn}_{\{3\}}^{(2)}$, and $\text{vn}_{\{1,3\}}^{(2)}$, respectively, with each edge having infinite capacity. Note that the edge from auxiliary node $\{1, 3\}$ to $\text{vn}_\emptyset^{(2)}$ forms part of a *self loop* in the virtual network. 38

2.4 The sum of queue lengths $\sum_{k,S} |Q_S^{(k)}|$ for different α. 55

2.5 Pmf of the header length: When $\alpha = 0.9$, the largest header length in the simulation of 10^5 time slots is 116 and the average is 2.5. When $\alpha = 0.95$, the largest and the average header lengths become 135 and 2.7. 56

2.6 Pmf of the total delay: When $\alpha = 0.9$, the largest total delay in the simulation of 10^5 time slots is 3332 and the average is 108.7. 94% of the packets have total delay ≤ 400 time slots. When $\alpha = 0.95$, the largest and the average delay become 13382 and 513.6. 67% of the packets have total delay ≤ 400 time slots. 57

3.1 The 3-user/3-file ($N = K = 3$) coded caching example. The sever has three files (A, B, C) of same size F and each file is split into three equal subfiles: $A = (A_1, A_2, A_3)$, $B = (B_1, B_2, B_3)$, and $C = (C_1, C_2, C_3)$. Each user k has same cache memory size F and store cache content (A_k, B_k, C_k). 61

3.2 Description of the regions of (M_1, M_2) and the corresponding corner points. The x-axis (resp. y-axis) is for the M_1 (resp. M_2) value. In this figure we assume $F_1 \geq F_2$ and only describe the cases when $M_1 \geq M_2$, thus the lower-half of the line $M_1 = M_2$. The cases of $F_1 < F_2$ and $M_1 < M_2$ can be obtained by swapping the file and user indices, respectively. Two scenarios are considered: (a) $F_1 \geq 2F_2$; (b) $2F_2 > F_1 \geq F_2$. 77

3.3 The average-rate capacity with $(F_1, F_2) = (1.5, 1)$ and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (4/15, 2/5, 2/15, 1/5)$. There are 12 facets and 14 corner points. Each corner point is labeled by a tuple $(M_1, M_2, \bar{R})$, where (M_1, M_2) give the location and the third coordinate specifies the corresponding exact average-rate capacity $\bar{R}$. The capacity is asymmetric with respect to (M_1, M_2) due to the heterogeneous file popularity. 78

3.4 Comparison of the average-rate capacity with the average rate of naïve likelihood-based uncoded caching, and the coded caching scheme in [41] that is optimized for the worst-case performance on some of the vertices in Fig. 3.3. 79

3.5 The average-rate capacity with $(F_1, F_2) = (1.5, 1)$ and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (2/15, 8/15, 4/15, 1/15)$. There are 10 facets and 13 corner points. Each corner point is labeled by a tuple $(M_1, M_2, \bar{R})$, where (M_1, M_2) give the location and the third coordinate specifies the corresponding exact average-rate capacity $\bar{R}$. The capacity is asymmetric with respect to (M_1, M_2) due to the heterogeneous file popularity. 80

3.6 Comparison of the average-rate capacity with the average rate of naïve likelihood-based uncoded caching, and the coded caching scheme in [41] that is optimized for the worst-case performance on some of the vertices in Fig. 3.5. 81

3.7 The average-rate capacity of uniform popularity, described for the case of $F_1 \geq F_2$. 81

4.1 The capacity $\tilde{R}$ of both the selfish and unselfish designs with $\Theta_1 = \Theta_2 = [N]$. 88

4.2 The minimum average rate $\tilde{R}$ of coded caching for $|\Theta_1| = N_1$, $|\Theta_2| = N_2$, and $\alpha = 1$. For any (M_1, M_2) inside each subregion, the rate $\tilde{R}$ is characterized by the corresponding equation marked in that region. 90

4.3 The uniform-average-rate capacity $\tilde{R} = \tilde{R}_{\text{gc}}$ with $\Theta_1 = \{1, 2\}$ and $\Theta_2 = \{1, 2, 3\}$. 92

4.4 The uniform-average-rate capacity $\tilde{R}_{\text{sc}}$ for selfish and coded prefetching schemes with $\Theta_1 = \{1, 2\}$ and $\Theta_2 = \{1, 2, 3\}$. 92

4.5 The uniform-average-rate capacity $\tilde{R}_{\text{gu}}$ for unselfish and uncoded prefetching schemes with $\Theta_1 = \{1, 2\}$ and $\Theta_2 = \{1, 2, 3\}$. 93

4.6 The minimum average rate $\tilde{R}$ for $|\Theta_1| = N_1$, $|\Theta_2| = N_2$, and $\alpha \geq 2$ with selfish and uncoded prefetching, where $v_2 : ((N_1 - \alpha)F, 0, (1 + (N_1 - 1)\alpha/(N_1N_2))F)$, $v_{10} : (0, (N_2 - \alpha)F, (1 + (N_2 - 1)\alpha/(N_1N_2))F)$, $v_{11} : (\alpha F/2, (N_2 - \alpha/2)F, (1 + (N_1 - N_2 - \alpha)\alpha/(2N_1N_2))F)$, $v_{12} : (\alpha F/2, \alpha F/2, (2 - (N_1 + N_2 + \alpha)\alpha/(2N_1N_2))F)$, $v_{13} : ((N_1 - \alpha/2)F, \alpha F/2, (1 + (N_2 - N_1 - \alpha)\alpha/(2N_1N_2))F)$, and $v_{14} : ((N_1 - \alpha/2)F, (N_2 - \alpha/2)F, (N_2 + N_1 - \alpha)\alpha F/(2N_1N_2))$. 96

4.7 Uniform-average rate as a function of the cache size M when $N_1 = N_2 = 128$ and $\alpha = 96$. 98

4.8 Uniform-average rate as a function of the FDS overlap α when $N_1 = N_2 = 128$ and $M = 16$. 99

B.1 The capacity of worst-case rate under the assumption $F_1 \geq F_2$. Each corner point is labeled by a tuple (M_1, M_2, R^*), where (M_1, M_2) describe the location and the third coordinate describe the corresponding exact worst-case rate capacity R^*. 202

ABSTRACT

Network Coding is a critical technique when designing next-generation network systems, since the use of network coding can significantly improve the throughput and performance (delay/reliability) of the system. In the traditional design paradigm without network coding, different information flows are transported in a similar way like commodity flows such that the flows are kept separated while being forwarded in the network. However, network coding allows nodes in the network to not only forward the packet but also process the incoming information messages with the goal of either improving the throughput, reducing delay, or increasing the reliability. Specifically, network coding is a critical tool when designing absolute Shannon-capacity-achieving schemes for various broadcasting and multicasting applications. In this thesis, we study the optimal network schemes for some applications with less restrictive network models. A common component of the models/approaches is how to use network coding to take advantage of a broadcast communication channel.

In the first part of the thesis, we consider the system of one server transmitting K information flows, one for each of K users (destinations), through a broadcast packet erasure channels with ACK/NACK. The capacity region of 1-to-K broadcast packet erasure channels with ACK/NACK is known for some scenarios, e.g., $K \leq 3$, etc. However, existing achievability schemes with network coding either require knowing the target rate $\vec{R}$ in advance, and/or have a complicated description of the achievable rate region that is difficult to prove whether it matches the capacity or not. In this part, we propose a new network coding protocol with the following features: (i) Its achievable rate region is identical to the capacity region for *all* the scenarios in which the capacity is known; (ii) Its achievable rate region is much more tractable and has been used to derive new capacity rate vectors; (iii) It employs *sequential encoding* that naturally handles dynamic packet arrivals; (iv) It automatically adapts to unknown packet arrival rates $\vec{R}$; (v) It is based on $\mathsf{GF}(q)$ with $q \geq K$. Numerically, for $K = 4$, it admits an average control overhead 1.1% (assuming each packet has 1000 bytes), average encoding memory usage 48.5 packets, and average per-packet delay 513.6 time slots, when operating at 95% of the capacity.

In the second part, we focus on the *coded caching system* of one server and K users, each user k has cache memory size M_k and demand a file among the N files currently stored at server. The coded caching system consists of two phases: Phase 1, the *placement phase*: Each user accesses the N files and fills its cache memory during off-peak hours; and Phase 2, the *delivery phase*: During the peak hours, each user submits his/her own file request and the server broadcasts a set of packet simultaneously to K users with the goal of successfully delivering the desired packets to each user. Due to the high complexity of coded caching problem with heterogeneous file size and heterogeneous cache memory size for arbitrary N and K, prior works focus on solving the optimal worst-case rate with homogeneous file size and mostly focus on designing order-optimal coded caching schemes with user-homogeneous file popularity that attain the lower bound within a constant factor. In this part, we derive the average rate capacity for microscopic 2-user/2-file ($N = K = 2$) coded caching problem with heterogeneous files size, cache memory size, and user-dependent heterogeneous file popularity. The study will shed some further insights on the complexity and optimal scheme design of general coded caching problem with full heterogeneity.

In the third part, we further study the coded caching system of one server, $K = 2$ users, and $N \geq 2$ files and focus on the user-dependent file popularity of the two users. In order to approach the exactly optimal uniform average rate of the system, we simplify the file demand popularity to binary outputs, i.e., each user either has no interest (with probability 0) or positive uniform interest (with a constant probability) to each of the N file. Under this model, the file popularity of each user is characterized by his/her *file demand set* of positive interest in the N files. Specifically, we analyze the case of user 1 and user 2 with file demand sets Θ_1 and Θ_2, respectively. We show the exact capacity results of one-overlapped file demand sets $|\Theta_1 \cap \Theta_2| = 1$ for arbitrary N and two-overlapped file demand sets $|\Theta_1 \cap \Theta_2| = 2$ for $N = 3$. To investigate the performance of large overlapped files we also present the average rate capacity under the constraint of selfish and uncoded prefetching with explicit prefetching schemes that achieve those capacities. All the results allow for arbitrary (and not necessarily identical) users' cache capacities and number of files in each file demand set.

1. INTRODUCTION

As the increasing demand of throughput and heterogeneous environment in nowadays network applications such like video streaming or data backup at cloud, continuous efforts dedicate to construct general network-capacity-approaching network schemes. Although the optimal rates for some of network models have been characterized, in many network settings, the complete theory and general network capacity region are still unknown. Recently, network coding is proposed to exploit multicasting opportunity with the goal of either improving the throughput, reducing delay, or increasing the reliability. In order to investigate the gain of network coding of different aspects in less-restrictive network models, we aim to design flexible network coding schemes that adapt the network models and achieve the network capacity regions.

In a network of point-to-point communication with constraints of channel capacity, the natural way for transporting information is to retransmit the data at each intermediate node. In traditional network scheme, the nodes apply decode-and-forward method, which decodes the packet from previous node and then transmits the copy of packet to the next one. In such paradigm, the delivery of data packets is similar to the transportation of commodity: The independent packet flows keep separate through the nodes in the network. Network coding is recently proposed to allow the nodes in the network not only to forward the packet but also to process the data with the goal of either improving the throughput, reducing delay, or increasing the reliability. The information flow in such schemes follows data processing relationship: For each node the output is a function of the input. By process the information, each transmission at input node in the network can increase more information content for the output nodes. For example. Consider the source aim to send information message X and Y to node 1 and 2, respectively, through a broadcasting channel. Suppose after first and second transmission, node 1 receives Y and node 2 receives X, then in the next transmission, instead of keeping sending X or Y, the source can broadcast message $[X + Y]$ to benefit both node 1 and 2 simultaneously.

From the illustrating, network coding is able to exploit multicasting opportunity to improve the network throughput. This leads to the straightforward question on how much

gain is achievable by using network coding in the broadcasting and multicasting applications. More specifically, given the network models, we aim to find network coding schemes that attain absolute Shannon capacity. In this thesis we focus on two well-defined network models: 1-to-K (one server and K destinations) broadcasting system with packet erasure channels and K-user/N-file coded caching system with an error-free broadcast channel. The 1-to-K broadcast packet erasure channel belongs to the extensively studied broadcast channels while using a simplified packet erasure model. The capacity-approaching result will given intuitions on more practical and complicate broadcast channel models for arbitrary number of users K. On the other hand, the K-user/N-file coded caching system currently draw research interest due to its throughput gain against traditional caching system. By studying the coded caching problem of heterogeneous file size, heterogeneous cache memory size, and heterogeneous file popularity, we will figure out the gain of general caching problem and the resulting network applications.

1.1 Outline of Thesis

The content of thesis is consisted of four chapters: (i) Introduction; (ii) Capacity-approaching protocol for general 1-to-K broadcast packet erasure channels with ACK/ NACK; (iii) Coded caching system of two users and two files; (iv) Code caching system for two users with heterogeneous file demand sets; and (v) Conclusion.

In Chapter 2, we first introduce the background of the 1-to-K broadcast packet erasure channels in Section 2.1 and then present the broadcast packet erasure channel model in Section 2.2. Section 2.3 describes the new achievable rate region for the 1-to-K broadcast packet erasure channels and Section 2.4 proposes the corresponding new scheme. Section 2.5 converts the theoretical achievability scheme to a practical protocol with the cost of a small amount of control overhead. The added overhead is not needed during the theoretical discussion but would highly facilitate practical implementation. Section 2.6 presents the simulation results and Section 2.7 summarizes the chapter.

In Chapter 3, we first introduce the background of the coded caching system in Section 3.1 and present the general coded caching system model of N files and K users in Section 3.2.

Section 3.3 shows *per-request capacity region* (PRCR) for 2-user/2-file ($N = K = 2$) coded caching system with heterogeneous file size, heterogeneous cache size, and user-dependent file popularity. Moreover, the corresponding achievable schemes are consisted of 7 basic coded caching schemes. Section 3.4 summarizes the work of the chapter.

In Chapter 4, we first recap the coded caching system discussed in Chapter 3 and then introduce the simplified user-dependent heterogeneous file popularity with *file demand set* in Section 4.1. We define the file demand sets and *selfish prefetching* for the coded caching system in Section 4.2. Section 4.3 and Section 4.4 proposes the average rate capacity for the scenarios of two users and $N \geq 2$ files with homogeneous and heterogeneous file demand sets, respectively. Section 4.5 illustrates the gap between the proposed selfish-and-uncoded prefetching and the optimal unselfish-and-coded prefetching scheme. Section 4.6 summarizes the work of the chapter.

In Chapter 5, we summarize how to use the network coding techniques to approach the capacity of the less-restrictive network models and shed some insights and thoughts for driving the capacity outer bounds and the achievable schemes or inner bounds.

1.2 Main Contributions

In Chapter 2, we introduce the new concept of *opportunistic packet evolution* and provide two main contributions on the capacity of general 1-to-K broadcast PEC with ACK/NACK:

1. Deriving a new inner bound that is provably tight for all the scenarios in which the capacity regions are known: (i) $K \leq 3$, (ii) the channel is *symmetric*, or (iii) the channel is spatially independent and the rate vectors are *one-sided fair*. That is, in terms of analytically characterizing the capacity region, the new inner bound is as good as any existing inner bounds.

2. Designing a new sequential coding scheme that has short delay and small memory usage, automatically adapts to dynamic packet arrivals, and attains the aforementioned new inner bound.

In Chapter 3, we consider a microscopic 2-user/2-file coded caching system of heterogeneous file size and cache memory size. We introduce the concept of per-request capacity region (PRCR) as the fundamental performance metric for describing the coded caching system. We then characterized the exact PRCR of the 2-user/2-file setting with full heterogeneity, including the following propositions:

1. The *exact capacity* or PRCR of the 2-user/2-file ($N = K = 2$) coded caching problem but under the simultaneously heterogeneous settings (i) heterogeneous files sizes, (ii) heterogeneous cache sizes, (iii) user-dependent file popularity, and (iv) average-rate analysis.

2. The PRCR can be used to derive the uniform-average capacity and worst-case capacity with heterogeneous file and cache memory size.

In Chapter 4, we introduce a simplified model of heterogeneous file popularity, where each user has uniform interest among files in his/her *file demand set* and has zero interest to the files otherwise. We analyze the performance of the coded caching system with $K = 2$ users, arbitrary N files, and homogeneous/heterogeneous file demand set. Specifically, we propose the following results:

1. The exact capacity results of one overlapped file demand sets $|\Theta_1 \cap \Theta_2| = 1$ for arbitrary N and two overlapped file demand sets $|\Theta_1 \cap \Theta_2| = 2$ for $N = 3$.

2. The average rate capacity with selfish and uncoded prefetching and arbitrary number of overlapped files.

2. CAPACITY-APPROACHING PROTOCOL FOR GENERAL 1-TO-K BROADCAST PACKET ERASURE CHANNELS WITH ACK/NACK

In this chapter, we study the capacity-approaching network coding scheme for 1-to-K broadcast packet erasure channels (PECs). In the system, one server aims to transmit K information flows, one for each of K users (destinations), through a broadcast PEC. We introduce a new inner bound and a corresponding achievable network coding scheme that is provably tight for all the scenarios in which the capacity regions are known. That is, in terms of analytically characterizing the capacity region, the new inner bound is as good as any existing inner bounds. Moreover, the new inner bound has a simple form in representation. The corresponding network coding scheme employs sequential encoding for dynamic packet arrivals and adapts to unknown packet arrival rates.

2.1 Introduction

Broadcast channels (BC) have been extensively studied as one of the earliest subjects in network information theory [1]. Although the capacity for the most general broadcast channel model is still unknown, the exact capacity region is known for cases like the degraded broadcast channel models [2], degraded broadcast channels with message side information [3], etc. Recently, BC with causal channel output feedback is also considered along the lines of degree-of-freedom analysis [4] and the ACKnowledgement-feedback for broadcast packet erasure channels (PEC) [5]–[10]. Although the latter is based on a packet-based setting, which is quite different from the symbol-based studies of most physical layer solutions, the PEC setting is widely used to model potential packet loss in modern wireless communication networks [5], [11]–[18]. In addition, the insights derived from the PEC setting could also benefit the symbol-based settings, see the recent results in [19]–[21].

Specifically, the 1-to-K broadcast PEC[1] is a memoryless and stationary broadcast channel for which a single packet is transmitted for each time slot and may be received by a random subset of K destinations. For each k two possible outcomes may happen at destination d_k: The destination may receive the transmitted packet successfully without error, or it may receive "nothing" due to packet erasure. After transmission, all K destinations inform the source whether the packet is received or not, i.e., sending ACK if successful reception and NACK if erasure. Due to the broadcast nature, even if a packet is intended for one destination, other destinations may overhear the packet, which can later be used as side information.

Network coding is an important component when characterizing the capacity of the broadcast PEC model with causal ACK/NACK. Take the 1-to-2 broadcast PEC for example as shown in Fig. 2.1. The source s intends to delivery packets $X_{1,1}, \ldots, X_{1,nR_1}$ to destination d_1 and packets $X_{2,1}, \ldots, X_{2,nR_2}$ to destination d_2 though the 1-to-2 broadcast PEC. Consider at some time slot, destination 1 receives packets $(X_{1,1}, X_{1,2}, X_{2,2})$ and destination 2 receives packets $(X_{1,1}, X_{2,1}, X_{1,3})$. Although $X_{2,2}$ and $X_{1,3}$ are undesirable packets for destination 1 and 2, respectively, it is not necessary for the source to take two time slots to retransmit $X_{2,2}$ and $X_{1,3}$ independently. Instead, source s can send the coded packet $X_{1,3} \oplus X_{2,2}$ that takes one time slot but benefits both d_1 and d_2 simultaneously. This simple idea has been generalized for the 1-to-K broadcast PEC with causal ACK/NACK. The main challenge of designing a capacity-approaching network coding scheme for general 1-to-K broadcast PECs lies in how to encode packets based on the latest overheard *side information* at different destinations to ensure that each destination is able to decode its desired packets within the shortest amount of time.

The 1-to-K broadcast PEC with ACK/NACK has been studied in many scenarios. The capacity region of the memoryless 1-to-2 broadcast PEC was characterized in [6], which shows that ACK/NACK feedback strictly improves the capacity of 1-to-2 broadcast PEC.

[1]↑In this work, we use the information-theoretic terminology of broadcast channels. That is, there are K destinations and each of them desires one of the K independent streams of packets. In the networking terminology, such a setting is named differently as the *K-unicast setting*. The broadcast setting herein (i.e., the K-unicast setting) is sharply different from the so-called 1-to-K multicast setting in the networking literature, where all K destinations desire the same common stream of packets. Some results of the 1-to-K multicast setting can be found in [22], [23].

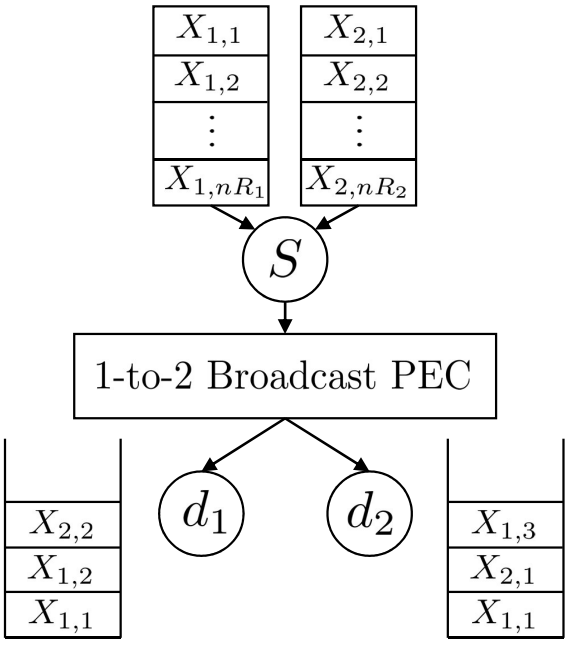


Figure 2.1. The 1-to-2 broadcast PEC example of source s and destinations d_1, d_2. The source s intends to delivery packets $X_{1,1}, \ldots, X_{1,nR_1}$ to destination d_1 and packets $X_{2,1}, \ldots, X_{2,nR_2}$ to destination d_2.

[24] extends the results of 1-to-2 broadcast PEC to characterize the *stability region* for stochastic sequential arrival sources. Optimal scheduling of 1-to-2 broadcast PECs with sequential packet arrivals and time-varying channels is proposed in [14], which shows that any sequential arrival rates within the Shannon capacity region can be stabilized by the proposed joint network coding and network scheduler. The broadcast PEC with ACK/NACK and arbitrary prior message side information[2] at destinations is investigated in [25]. Most achievability results in the literature are based on linear network coding operations, though recent results by [26] also explore typicality-based designs.

This work introduces the new concept of *opportunistic packet evolution* [27], which is used to study two separate but closely related subjects of the general 1-to-K broadcast PEC with ACK/NACK.

Subject 1: Deriving a new inner bound that sheds further insights on the capacity region. On this front, we derive a new inner bound that is provably tight for all the scenarios in which the capacity regions are known.[3] That is, in terms of analytically characterizing the capacity region, the new inner bound is as good as any existing inner bounds [5], [17], [18].

In terms of the proof/construction techniques, two main ingredients of the existing inner bounds [5], [17], [18] are the concepts of *user ordering* and *collapsed overhearing set matching*.[4] The new concept of opportunistic packet evolution completely subsumes these two concepts, and the resulting new inner bound admits a much simpler form, which allows for more efficient numerical evaluation of the achievable rates and has been used to analytically identify new capacity rate vectors, a difficult task with the existing inner bounds.

Subject 2: We depart from the traditional block code setting and design a new sequential coding scheme that has short delay and small memory usage, automatically adapts to dynamic packet arrivals, and attains the aforementioned new inner bound. Specifically, most

[2]↑The setting of prior side information assumes that a certain amount of side information is available *before* the transmission begins, which is different from the traditional no-side-information scenarios [5], [14], [17], [18] for which the side information is overheard *during* transmission.

[3]↑ The capacity region of the 1-to-K broadcast PECs is known if any of the following three conditions hold: (i) $K \leq 3$, (ii) the channel is *symmetric*, or (iii) the channel is *spatially independent* and the rate vectors are *one-sided fair*.

[4]↑The concepts of user ordering and collapsed overhearing set matching are powerful tools in terms of finding the largest achievable rate region. But they also have significant drawbacks both analytically, i.e., being too complicated to be tractable, and in practice, i.e., incurring too much delay and complexity. Also see the detailed discussion of Drawbacks 2 and 3 in Subject 2 for their practical implications.

achievability schemes, e.g., [5], [17], have the following drawbacks. *Drawback 1:* To transmit packets at a given rate vector $\vec{R}$, the schemes have to first solve a linear-programming (LP) problem with $\vec{R}$ as input and then use the optimal LP solution to adjust the design parameters for the target $\vec{R}$. However, in practice the packet arrival rates $\vec{R}$ are usually not known in advance.

Drawback 2: The network coding opportunity exists *after* some node d_k overhears some previous transmissions. Therefore, there is a strict causality relationship between when the coding opportunity is created (through overhearing) and when one can start to combine packets to capitalize the created coding opportunity. In order to maximize the achievable rates, the existing schemes [5], [17], [26] impose a strict *user/transmission order* that takes into account the causality constraints. Since a strict transmission order requires packets to "wait in the queues" before transmission, it incurs long queuing delay in practice.

Drawback 3: Intuitively, network coding transmits a linear sum of multiple packets for which the overhearing set of each constituent packet is "matched" with the other constituent packets. Since there are 2^K-1 non-empty subsets of all K destinations, there are 2^K-1 number of ways to "match" the overhearing sets. For each time instant, one of these 2^K-1 ways is used to generate the coded packet. Unfortunately, it turns out that strictly adhering to this simple rule yields strictly suboptimal performance. To remedy the deficiency, the concept of *collapsed overhearing set matching* (COSM) is introduced in [5], [17], [18] and has since been one of the central ingredients used to achieve the capacity region of the case of $K=3$ and to design high-performance inner bounds for general K. Nonetheless, searching for the COSM involves comparing roughly $O\left(\left(\frac{K+1}{\ln(K+2)}\right)^{K+1}\right)$ coding choices for each time instant, a dramatic increase from the intuitive number 2^K-1. The complexity of COSM is exceedingly high even for very small K.

Drawback 4: The theoretic schemes in [5], [17] allow for asymptotically large memory, asymptotically large queuing/decoding delay, and asymptotically large complexity, which make it difficult to implement in practice. Note that there are several existing schemes that aim to simplify the code design and make network coding practical, see [28] and the references therein, but at the cost of sacrificing the provable optimality.

Table 2.1. Comparison of the numbers of coding choices. COSM stands for "collapsed overhearing set matching" and B_m represents the m-th Bell number. For reference, $B_6 = 203$, $B_{11} = 678570$, and $B_{16} \approx 10^{10}$.

No. users	Proposed scheme	Intra-level scheme in [18] without COSM	Inter-level scheme in [18] with COSM
1	1	1	1
2	3	5	8
3	7	25	47
4	15	124	244
5	31	616	1227
K	$2^K - 1$	$\left(\sum_{m=0}^{K} \binom{K}{m} B_{m+1}\right) - 2^{K+1} + K + 1$	No clean expression available

Our new network code design addresses the above four drawbacks simultaneously. Firstly, our scheme requires only the knowledge of the underlying channel statistics, and will automatically adapt to the unknown arrival rates $\vec{R}$. Secondly, by using the new concept of opportunistic packet evolution, our scheme does not impose any user/transmission order and no longer needs to search for COSM. Instead, for each time instant the new scheme chooses one out of $2^K - 1$ possible coding choices but can still attain rates that are equal or better than all the existing schemes. Thirdly, the proposed scheme is based on sequential coding, which has short delay and low memory usage.

Comparison to the closest existing work [18]: The authors in [18] design a sequential coding scheme that addresses Drawbacks 1, 2, and 4 successfully. In addition, only binary XOR operations are used in [18] and the scheme admits the highly desired feature of *instantaneous decodability*. For comparison, our results are based on $\mathsf{GF}(q)$ instead of binary XOR. Our scheme is not instantly decodable, though numerical results show that the total delay, queueing plus decoding delay, is still quite manageable for practical scenarios.

The defining difference between our scheme and [18] is the new notion of opportunistic packet evolution, which enables our scheme to successfully address Drawback 3. In the following, we highlight the benefits of opportunistic packet evolution.

- Provable optimality: Our new scheme is provably capacity achieving for all the scenarios for which the capacity region is known, also see footnote 3. For comparison, the

inter-level scheme in [18] is provably optimal only when the following conditions hold simultaneously: (i) $K = 4$ and (ii) the channel is symmetric and spatially independent. Our new inner bound can also be used to identify new capacity vectors, which is difficult to do with the result in [18] due to its high complexity.

- Inner bound description: Our achievable rate region (ARR) has a very compact form that consists of 2 inequalities. For comparison, the scheme in [18] reacts to various sub-cases differently, which makes the corresponding ARR more difficult to describe. For example, for $K = 5$ and asymmetric channels, our ARR is described as an LP problem with 31 variables. The ARR of the general *inter-level scheme* in [18] is an LP problem with 1227 variables.

- Backpressure algorithms: Both this work and [18] are based on *backpressure scheduling* and can automatically adapt to any unknown $\vec{R}$. However, our backpressure scheduler has a significantly smaller number of "competing actions" to consider during each backpressure computation, which greatly reduces the complexity and delay inherent in the backpressure solution. Taking $K = 5$ for example, our scheme compares the backpressures of only 31 different actions while [18] compares the backpressures of 1227 actions. Also see Table 2.1.

2.2 1-to-K Broadcast Packet Erasure Channel Model

This work considers two major settings, the block-coding and the sequential-coding settings. The goal of the block-coding setting is that given a block of $n \cdot R_k$ messages how to transmit the messages with asymptotically small error probability within n channel usage. The goal of the sequential-coding setting is that given i.i.d. Poisson random arrival processes with arrival rates R_k, how to transmit the messages *error free* while keeping the queue lengths stable. The largest limit of the former setting is known as the *Shannon capacity region* and the limit of the latter is termed the *optimal stability region*. In the following, we formally define these two settings and prove that the stability region is an inner bound of the capacity

region. Since this work focuses exclusively on the inner bound, in the sequel we focus on characterizing a new stability region.

2.2.1 The 1-to-K Broadcast Packet Erasure Channel

For any positive integer K, we use $[K] \triangleq \{1, 2, \ldots, K\}$ to denote the set of integers from 1 to K and use the notation $2^{[K]} \triangleq \{S : S \subseteq [K]\}$ to denote the collection of all subsets of $[K]$. We consider a 1-to-K broadcast PEC from source s to destinations d_k, $k \in [K]$. There are K independent packet streams, one stream for each d_k. All packets are drawn independently and uniformly randomly from a fixed finite field $\mathsf{GF}(q)$ satisfying $q \geq K$. The scenario of $\mathsf{GF}(q)$ with $q < K$ is beyond the scope of this work.

At any discrete time slot $t \in \{1, 2, \cdots\}$, source s sends a packet $Y(t) \in \mathsf{GF}(q)$ as an input of the broadcast PEC. The channel outputs a K-dimensional vector $(Z_1(t), \cdots, Z_K(t))$, where for each k we have $Z_k(t) \in \{Y(t), *\}$. Herein $Z_k(t) = Y(t)$ means that the transmitted packet $Y(t)$ is received successfully by d_k and $Z_k(t) = *$ means that the transmitted $Y(t)$ is "erased" at d_k. We define $S_{\text{rx}}(t) \triangleq \{k : Z_k(t) = Y(t)\}$ as the set of indices k for which $Z_k(t) = Y(t)$, i.e., $S_{\text{rx}}(t)$ is the *reception set* at time t. The distribution of the random 1-to-K broadcast PEC is thus determined uniquely by the distribution of the reception set $S_{\text{rx}}(t)$.

We assume that the random process of $\{S_{\text{rx}}(t) : t \in \{1, 2, \cdots\}\}$ is *stationary* and *memoryless* and does not depend on the input $\{Y(t) : t\}$. The probability that $Y(t)$ is received by all d_k for $k \in S$ is then denoted by

$$p_S \triangleq \sum_{\tilde{S} \in 2^{[K]} : \tilde{S} \supseteq S} \Pr(S_{\text{rx}}(t) = \tilde{S}). \tag{2.1}$$

The above definition of p_S can be further generalized as follows: For any disjoint subsets $S, T \in 2^{[K]}$.

$$p_{S\overline{T}} \triangleq \sum_{\tilde{S} \in 2^{[K]} : \tilde{S} \supseteq S, \tilde{S} \cap T = \emptyset} \Pr(S_{\text{rx}}(t) = \tilde{S}). \tag{2.2}$$

That is, $p_{S\overline{T}}$ represents the probability that $Y(t)$ is received by all d_k for $k \in S$ but by none of the d_k for $k \in T$. The new input argument T excludes some events that were previously counted when computing p_S. Note that we do not care whether $Y(t)$ is received or not by those destinations in $[K]\backslash(S \cup T)$ thus the summation in (2.2). Comparing (2.1) and (2.2), it is clear that $p_S = p_{S\overline{\emptyset}} \geq p_{S\overline{T}}$ for all $S \cap T = \emptyset$.

For simplicity, when either S or T is empty, we use $p_{\overline{T}}$ and p_S as shorthand for $p_{\emptyset\overline{T}}$ and $p_{S\overline{\emptyset}}$ respectively. For the special instance of $T = \emptyset$ and $S = \{k\}$ containing only one element k, we use $p_k \triangleq p_{\{k\}\overline{\emptyset}}$ as shorthand, which is simply the marginal success probability for destination d_k.

For convenience, we also define

$$p_{\cup S} \triangleq 1 - p_{\overline{S}}.$$

which is the probability that *at least* one of the destination $k \in S$ receives the packet $Y(t)$.

We close this subsection by introducing a commonly used definition.

Definition 2.2.1. *A 1-to-K broadcast PEC is* spatially independent *if the distribution of $S_{\mathrm{rx}}(t)$ satisfies*

$$p_{S\overline{[K]\backslash S}} = \left(\prod_{\mathrm{i}\in S} p_{\mathrm{i}}\right)\left(\prod_{\mathrm{j}\in[K]\backslash S}(1-p_{\mathrm{j}})\right),\ \forall S \in 2^{[K]}. \tag{2.3}$$

The results of this work do not assume the underlying 1-to-K broadcast PEC being spatially independent, unless explicitly stated otherwise.

2.2.2 The Block-Coding Setting

Given any rate vector $\vec{R} \triangleq (R_1, \ldots, R_K)$, we now define the traditional block-coding setting with block length n. We denote $\mathbf{X}_k = \{X_{k,\mathrm{i}} \in \mathsf{GF}(q) : \mathrm{i} = 1, 2, \cdots, nR_k\}$ as the nR_k packets of stream k. For any time $t \in [n]$, source s sends a coded symbol

$$Y(t) = f_t^{(\mathrm{bl})}(\mathbf{X}_1, \ldots, \mathbf{X}_K, [S_{\mathrm{rx}}]_1^{t-1}) \in \mathsf{GF}(q)$$

where the encoding function $f_t^{(\mathrm{bl})}(\cdot)$ takes all information symbols $\{\mathbf{X}_k\}$ and the past reception sets $[S_{\mathrm{rx}}]_1^{t-1} \triangleq \{S_{\mathrm{rx}}(\tau) : \tau \in [t-1]\}$ as input and generates the coded symbol $Y(t) \in \mathsf{GF}(q)$. The knowledge of $[S_{\mathrm{rx}}]_1^{t-1}$ models the causal ACK/NACK fed back from the destinations to the source. The superscript (bl) emphasizes that it is the block-coding setting. In the end of time n, each d_k decodes

$$\hat{\mathbf{X}}_k = g_k^{(\mathrm{bl})}([Z_k]_1^n, [S_{\mathrm{rx}}]_1^n), \tag{2.4}$$

based on all the received packets $[Z_k]_1^n \triangleq \{Z_k(t) : t \in [n]\}$ and all the reception sets $[S_{\mathrm{rx}}]_1^n$. Here we assume $[S_{\mathrm{rx}}]_1^n$, the network-wide ACK/NACK information is known to all the destinations.[5]

A network block code of length n is defined by the corresponding n encoding functions $f_t^{(\mathrm{bl})}(\cdot)$, $t \in [n]$, and K decoding functions $g_k^{(\mathrm{bl})}(\cdot)$, $k \in [K]$. The achievable rates of a 1-to-K broadcast PEC with ACK/NACK are then defined by

Definition 2.2.2. *Given any fixed* $\mathsf{GF}(q)$, $(R_1, \cdots, R_K)$ *is achievable if for any* $\epsilon > 0$, *there exists a network block code of length* n *such that*

$$\Pr\left(\bigcup_{k\in[K]} \{\hat{\mathbf{X}}_k \neq \mathbf{X}_k\}\right) \leq \epsilon.$$

The capacity region is the closure of all achievable rate vectors $(R_1, \ldots, R_K)$.

The exact capacity region for general channel parameters $\{p_{S\overline{[K]\setminus S}} : \forall S \in 2^{[K]}\}$ remains an open problem. We close this subsection by restating the best known existing capacity outer bound results [5], [17].

Proposition 2.2.1 ([5], [17])**.** *A* K*-permutation (or simply permutation) is a bijective function from the set* $[K]$ *to itself. That is,* $\boldsymbol{\pi} : [K] \mapsto [K]$. *Given any permutation* $\boldsymbol{\pi}$, *for all*

[5]↑In practice, each d_k naturally knows whether itself has received the packet $Y(t)$ or not, but may not know the reception status of other $d_{\tilde{k}}$, $\tilde{k} \neq k$. Therefore, in practice, source s may need to *broadcast* the network-wide information $[S_{\mathrm{rx}}]_1^n$ to each individual d_k using additional $\frac{n \cdot K}{\log_2(q) \min_k p_k}$ packets. For sufficiently large q, the rate penalty of broadcasting $[S_{\mathrm{rx}}]_1^n$ is negligible and our model thus directly assumes that $[S_{\mathrm{rx}}]_1^n$ is known to all d_k.

$\mathrm{j} \in [K]$ *we define* $S_{\mathrm{j}}^{\boldsymbol{\pi}} \triangleq \{\boldsymbol{\pi}(l) : l \in [\mathrm{j}]\}$ *as the set of the first* j *elements according to* $\boldsymbol{\pi}$. *Any achievable rate vector* $(R_1, \ldots, R_K)$ *must satisfy the following* $K!$ *inequalities:*

$$\sum_{\mathrm{j}=1}^{K} \frac{R_{\boldsymbol{\pi}(\mathrm{j})}}{p_{\cup S_{\mathrm{j}}^{\boldsymbol{\pi}}}} \leq 1, \quad \forall \boldsymbol{\pi}. \tag{2.5}$$

For example, if $K = 3$, the above outer bound becomes:

$$\begin{cases} \frac{R_1}{p_1} + \frac{R_2}{p_{\cup\{1,2\}}} + \frac{R_3}{p_{\cup\{1,2,3\}}} \leq 1 \\ \frac{R_2}{p_2} + \frac{R_1}{p_{\cup\{1,2\}}} + \frac{R_3}{p_{\cup\{1,2,3\}}} \leq 1 \\ \frac{R_1}{p_1} + \frac{R_3}{p_{\cup\{1,3\}}} + \frac{R_2}{p_{\cup\{1,2,3\}}} \leq 1 \\ \frac{R_3}{p_3} + \frac{R_1}{p_{\cup\{1,3\}}} + \frac{R_2}{p_{\cup\{1,2,3\}}} \leq 1 \\ \frac{R_2}{p_2} + \frac{R_3}{p_{\cup\{2,3\}}} + \frac{R_1}{p_{\cup\{1,2,3\}}} \leq 1 \\ \frac{R_3}{p_3} + \frac{R_2}{p_{\cup\{2,3\}}} + \frac{R_1}{p_{\cup\{1,2,3\}}} \leq 1 \end{cases} \tag{2.6}$$

In [5], [17] it is proven that in the scenario of $K = 3$, this outer bound is indeed the capacity region.

2.2.3 The Sequential-Coding Setting

In this setting, the packets arrive sequentially for each time slot. We denote the number of user-k packets arriving at time slot t by $A_k(t)$ and assume $A_k(t)$ is an i.i.d. Poisson random process with mean $\mathsf{E}\{A_k(t)\} = R_k$. At time t, we denote $M_k(t) \triangleq \sum_{\tau=1}^{t} A_k(\tau)$ as the cumulative number of packet arrivals until time t and $\mathbf{X}_k(t) = \{X_{k,\mathrm{i}} : \mathrm{i} = 1, 2, \ldots, M_k(t)\}$ as the set of all user-k packets that have already arrived at source s by time t. Obviously, $\mathbf{X}_k(t) \backslash \mathbf{X}_k(t-1)$ represents the content of the user-k packets arriving during time t. We use $\vec{A}(t) = (A_1(t), \ldots, A_K(t))$ to denote the arrival patterns of all K destinations at time t.

Unlike the block-coding setting which is not worried about memory usage, any sequential network code has to carefully manage its memory. Specifically, the content of the memory at the *end* of time t is denoted by $Q(t) \in (\mathsf{GF}(q))^*$, a variable-length string of symbols in $\mathsf{GF}(q)$. In the networking terminology, $Q(t)$ represents the content of the "queue" at time t.

At the beginning of each time slot t, source s transmits a coded symbol $Y(t)$ using the broadcast PEC. That is,

$$Y(t) = f_t^{(\mathrm{sq})}\left(Q(t-1), \bigcup_{k\in[K]} \mathbf{X}_k(t)\backslash\mathbf{X}_k(t-1), [S_{\mathrm{rx}}]_1^{t-1}\right) \tag{2.7}$$

where $f_t^{(\mathrm{sq})}(\cdot)$ is the sequential encoding function that takes as input $Q(t-1)$, the content of the memory at the end of time $t-1$, the new arrivals of user-k packets at time t for all $k \in [K]$, and the past reception status $[S_{\mathrm{rx}}]_1^{t-1}$ obtained through causal ACK/NACK feedback. The superscript (sq) emphasizes we consider the setting of sequential coding.

After the transmission over the broadcast PEC, new ACK/NACK will be fed back to source s and in the end of time t the source updates its memory

$$Q(t) = f_{\mathrm{buff},t}^{(\mathrm{sq})}\left(Q(t-1), \bigcup_{k\in[K]} \mathbf{X}_k(t)\backslash\mathbf{X}_k(t-1), [S_{\mathrm{rx}}]_1^{t}\right). \tag{2.8}$$

Comparing to (2.7), the buffer management function $f_{\mathrm{buff},t}^{(\mathrm{sq})}$ has an additional input argument $S_{\mathrm{rx}}(t)$, the reception status of time t.

The definition of decodability of a sequential encoder/decoder pair needs special attention, and we define the sequential decoder $g_{k,t}^{(\mathrm{sq})}$ as follows.

$$\hat{\mathbf{X}}_k(t) = g_{k,t}^{(\mathrm{sq})}\left([Z_k]_1^t, Q(t), [S_{\mathrm{rx}}]_1^t, [\vec{A}]_1^t\right) \tag{2.9}$$

and we require the decoder function $g_{k,t}^{(\mathrm{sq})}$ to satisfy

$$\Pr\left(\hat{\mathbf{X}}_k(t) \neq \mathbf{X}_k(t)\right) = 0, \quad \forall k, t. \tag{2.10}$$

The rationale behind (2.9) and (2.10) is as follows. Consider the end of time t. It is likely that $\mathbf{X}_k(t)$ cannot be fully decoded by d_k at the end of the current time t. One reason is that some of them may have just arrived at s and are still waiting for their turn to be transmitted, the so-called *queueing delay*. Another reason is that they may be transmitted in a coded form and thus some *decoding delay* is needed before actual decoding.

That said, if at the end of time t a genie conveys to destination d_k all the (undelivered) entries/content in the current source queue $Q(t)$ and the past reception status $[S_{\text{rx}}]_1^t$, then d_k must be able to perfectly decode all $M_k(t)$ packets in $\mathbf{X}_k(t)$. Otherwise, some packets in $\mathbf{X}_k(t)$ are *permanently lost* and can no longer be decoded in the future since only the content in $Q(t)$ may have any impact on future encoding. Following this rationale, d_k, if equipped by a genie with the knowledge of all the received symbols $[Z_k]_1^t$, the queue content $Q(t)$, the past network-wide reception status $[S_{\text{rx}}]_1^t$, and the past arrival patterns $[\vec{A}]_1^t$, must be able to decode all packets in $\mathbf{X}_k(t)$ error-freely, which is captured by (2.9) and (2.10).

A sequential network coding scheme is thus described by the $K+2$ infinite series of the encoding functions $f_t^{(\text{sq})}$, the buffer management functions $f_{\text{buff},t}^{(\text{sq})}$, and the decoding functions $g_{k,t}^{(\text{sq})}$ in (2.7), (2.8), (2.9), and (2.10). We then define the stability region as follows.

Definition 2.2.3. *A rate $(R_1, \ldots, R_K)$ is stable if there exists a sequential network coding scheme such that*

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \mathsf{E}\left\{|Q(\tau)|\right\} < \infty \tag{2.11}$$

where $|Q(\tau)|$ is the length of the queue $Q(\tau)$.

Remark: The above definition is compatible with the traditional stability region definition for non-coded solutions. Specifically, a traditional scheme has a very simple buffer management function $f_{\text{buff},t}^{(\text{sq})}$ in (2.8), that stores the new packets $\mathbf{X}_k(t)\backslash\mathbf{X}_k(t-1)$ in the queue while removing those packets that have been successfully delivered (those being ACKed). The transmission function $f_t^{(\text{sq})}$ of the traditional non-coded scheme then picks one packet in the queue $Q(t-1)$ or the newly arrived packets $\mathbf{X}_k(t)\backslash\mathbf{X}_k(t-1)$ and transmits the chosen packet uncodedly. Which packet to choose is decided by a "network scheduler", which takes the past reception status $[S_{\text{rx}}]_1^{t-1}$ as input. The decodability conditions (2.9) and (2.10) hold naturally for the traditional uncoded scheme since this simple memory management scheme ensures that $Q(t)$ contains all the packets that have not been delivered. The goal of the network scheduler is then to stabilize the queue length $|Q(t)|$ as in (2.11) while using simple uncoded $f_t^{(\text{sq})}$ and $f_{\text{buff},t}^{(\text{sq})}$. Our new definitions can be viewed as a generalization of

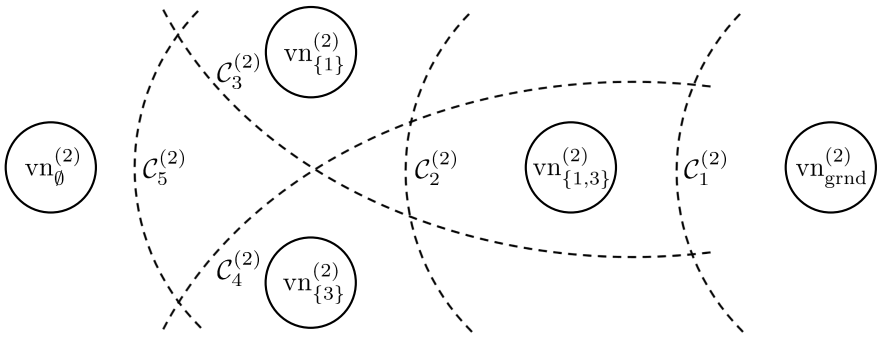


Figure 2.2. The virtual nodes and proper-cuts for $K = 3$ and $k = 2$.

the traditional ones by allowing both the transmission $Y(t)$ and the packets stored in the memory $Q(t)$ to be coded.

We close this subsection by establishing the relationship between the block-coding-based capacity region and the sequential-coding-based stability region.

Proposition 2.2.2. *If a rate vector* $(R_1, \ldots, R_K)$ *is stable, then it also belongs to the capacity region.*

The proof of Proposition 2.2.2 is provided in Appendix A.1.

Since the stability region is a subset of the capacity region, in the sequel we will describe a new stability region and the result naturally serves as a new inner bound of the capacity region.

2.3 A New Stability Region

To describe the new stability region, we first introduce the following definitions.

Definition 2.3.1. *For any fixed* $k \in [K]$*, we define* 2^{K-1} virtual nodes *and label each of them by a subset* $S \subseteq [K]\backslash\{k\}$*. It is thus convenient to simply denote each virtual node as* $\mathrm{vn}^{(k)}_S$ *for all* $S \in 2^{[K]\backslash\{k\}}$*. We then define another virtual node, called the* ground node*, and denoted it by* $\mathrm{vn}^{(k)}_{\mathrm{grnd}}$*. Totally there are* $2^{K-1}+1$ *virtual nodes for a given* k *and these virtual nodes form a* virtual sub-network *of destination* d_k*. The* K *virtual sub-networks (totally* $K(2^{K-1}+1)$ *virtual nodes) jointly form the overall virtual network.*

Definition 2.3.2. *For any fixed k, consider the k-th virtual sub-network. We say a subset of the $2^{K-1}+1$ virtual nodes $\{\mathrm{vn}_S^{(k)} : \forall S\} \cup \{\mathrm{vn}_{\mathrm{grnd}}^{(k)}\}$, denoted by $\mathcal{C}^{(k)}$, is a* cut *for d_k if $\mathrm{vn}_{\emptyset}^{(k)} \notin \mathcal{C}^{(k)}$ and $\mathrm{vn}_{\mathrm{grnd}}^{(k)} \in \mathcal{C}^{(k)}$.*

Definition 2.3.3. *For any fixed k, we say a cut $\mathcal{C}^{(k)}$, is a* proper-cut *if it also satisfies: If $\mathrm{vn}_{S_1}^{(k)} \in \mathcal{C}^{(k)}$ for some S_1, then for all $S_2 \supseteq S_1$ we must have $\mathrm{vn}_{S_2}^{(k)} \in \mathcal{C}^{(k)}$.*

An illustration of the above three definitions is provided in Fig. 2.2, for which we assume $K=3$ and plot the k-th virtual sub-network for $k=2$. The corresponding virtual nodes are $\left\{\mathrm{vn}_S^{(2)} : \forall S \subseteq [3]\backslash\{2\}\right\} = \left\{\mathrm{vn}_{\emptyset}^{(2)}, \mathrm{vn}_{\{1\}}^{(2)}, \mathrm{vn}_{\{3\}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\right\}$ plus the ground node $\mathrm{vn}_{\mathrm{grnd}}^{(2)}$. There are exactly 5 proper-cuts:

$$\mathcal{C}_1^{(2)} = \left\{\mathrm{vn}_{\mathrm{grnd}}^{(2)}\right\}, \tag{2.12}$$

$$\mathcal{C}_2^{(2)} = \left\{\mathrm{vn}_{\mathrm{grnd}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\right\}, \tag{2.13}$$

$$\mathcal{C}_3^{(2)} = \left\{\mathrm{vn}_{\mathrm{grnd}}^{(2)}, \mathrm{vn}_{\{1\}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\right\}, \tag{2.14}$$

$$\mathcal{C}_4^{(2)} = \left\{\mathrm{vn}_{\mathrm{grnd}}^{(2)}, \mathrm{vn}_{\{3\}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\right\}, \tag{2.15}$$

$$\mathcal{C}_5^{(2)} = \left\{\mathrm{vn}_{\mathrm{grnd}}^{(2)}, \mathrm{vn}_{\{1\}}^{(2)}, \mathrm{vn}_{\{3\}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\right\}. \tag{2.16}$$

A new stability region can then be described as follows.

Proposition 2.3.1. *A rate vector $\vec{R} \triangleq (R_1, \ldots, R_K)$ is* stable *if there exists $2^K - 1$ non-negative variables $\left\{x_T \geq 0 : \forall T \in 2^{[K]}\backslash\{\emptyset\}\right\}$ satisfying the following two groups of conditions.*

[Condition 1:] The time-sharing condition:

$$\sum_{T:T\in 2^{[K]}\backslash\{\emptyset\}} x_T < 1; \tag{2.17}$$

and [Condition 2:] The min-cut condition: For all k and for all $\mathcal{C}^{(k)}$ being a proper cut,

$$\sum_{\substack{S:S\in 2^{[K]\backslash\{k\}},\\ \mathrm{vn}_S^{(k)}\notin\mathcal{C}^{(k)}}} x_{S\cup\{k\}} \cdot \left(p_k + \sum_{\substack{S_X:\\ S_X\in\mathcal{F}(k,S,\mathcal{C}^{(k)})}} p_{S_X\overline{[K]\backslash(S_X\cup S)}} \right) \geq R_k, \tag{2.18}$$

where the set $\mathcal{F}(k, S, \mathcal{C}^{(k)})$ *is defined by*

$$\mathcal{F}(k, S, \mathcal{C}^{(k)}) \triangleq \Big\{ S_X \in 2^{[K]\setminus(S\cup\{k\})} : \exists \tilde{S} \subseteq (S_X \cup S) \\ s.t.\ \mathrm{vn}_{\tilde{S}}^{(k)} \in \mathcal{C}^{(k)} \Big\}. \tag{2.19}$$

Condition 1 contains a single time-sharing inequality. To illustrate Condition 2, we expand it for the case of $K = 3$ in the following. When $K = 3$, any fixed k has 5 distinct proper-cuts $\mathcal{C}^{(k)}$. There are thus $3 \cdot 5 = 15$ inequalities in Condition 2. Specifically, using the $\mathcal{C}_1^{(2)}$ to $\mathcal{C}_5^{(2)}$ defined in (2.12) to (2.16), Condition 2 becomes

$$\mathcal{C}_1^{(2)} : \left(x_{\{2\}} + x_{\{1,2\}} + x_{\{2,3\}} + x_{\{1,2,3\}}\right) \cdot p_2 \geq R_2, \tag{2.20}$$

$$\mathcal{C}_2^{(2)} : x_{\{2\}} \left(p_2 + p_{\{1,3\}\overline{2}}\right) + x_{\{1,2\}} \left(p_2 + p_{3\overline{2}}\right) \\ + x_{\{2,3\}} \left(p_2 + p_{1\overline{2}}\right) \geq R_2, \tag{2.21}$$

$$\mathcal{C}_3^{(2)} : x_{\{2\}} \left(p_2 + p_{1\overline{\{2,3\}}} + p_{\{1,3\}\overline{2}}\right) + x_{\{2,3\}} \left(p_2 + p_{1\overline{2}}\right) \\ \geq R_2, \tag{2.22}$$

$$\mathcal{C}_4^{(2)} : x_{\{2\}} \left(p_2 + p_{3\overline{\{1,2\}}} + p_{\{1,3\}\overline{2}}\right) + x_{\{1,2\}} \left(p_2 + p_{3\overline{2}}\right) \\ \geq R_2, \tag{2.23}$$

$$\mathcal{C}_5^{(2)} : x_{\{2\}} \cdot \left(p_2 + p_{1\overline{\{2,3\}}} + p_{3\overline{\{1,2\}}} + p_{\{1,3\}\overline{2}}\right) \geq R_2. \tag{2.24}$$

For example, consider proper-cut $\mathcal{C}_3^{(2)}$ and the corresponding (2.18) in Condition 2. By the definition of $\mathcal{C}_3^{(2)}$ in (2.14), there are exactly two virtual nodes $\mathrm{vn}_{\emptyset}^{(2)}$ and $\mathrm{vn}_{\{3\}}^{(2)}$ not in $\mathcal{C}_3^{(2)}$, i.e., the summation in (2.18) is over $S = \emptyset, \{3\}$.

For the case $S = \emptyset$, we have $\mathcal{F}(2, \emptyset, \mathcal{C}_3^{(2)}) = \{\{1\}, \{1,3\}\}$. The reason is that S_X in (2.19) is now chosen from $2^{[K]\setminus S\cup\{k\}} = \{\emptyset, \{1\}, \{3\}, \{1,3\}\}$. If $S_X = \{1\}$ or $\{1,3\}$, there exists $\tilde{S} = \{1\}$ satisfying $\tilde{S} \subseteq S_X \cup S$ and $\mathrm{vn}_{\tilde{S}}^{(k)} \in C_3^{(2)}$. Furthermore, no such $\tilde{S}$ exists if $S_X = \emptyset$ or $S_X = \{3\}$. As a result, $\mathcal{F}(2, \emptyset, \mathcal{C}_3^{(2)}) = \{\{1\}, \{1,3\}\}$, which then contributes to the term $x_{\{2\}} \left(p_2 + p_{1\overline{\{2,3\}}} + p_{\{1,3\}\overline{2}}\right)$ in (2.22).

For the case $S = \{3\}$, we have $\mathcal{F}(2, \{3\}, \mathcal{C}_3^{(2)}) = \{\{1\}\}$. The reason is that S_X in (2.19) is now chosen from $2^{[K]\setminus S\cup\{k\}} = \{\emptyset, \{1\}\}$. If $S_X = \{1\}$, there exists $\tilde{S} = \{1\}$ satisfying

$\tilde{S} \subseteq S_X \cup S$ and $\text{vn}_{\tilde{S}}^{(k)} \in \mathcal{C}_3^{(2)}$. Furthermore, no such $\tilde{S}$ exists if $S_X = \emptyset$. As a result, $\mathcal{F}(2, \{3\}, \mathcal{C}_3^{(2)}) = \{\{1\}\}$, which contributes to the term $x_{\{2,3\}}(p_2 + p_{1\bar{2}})$ in (2.22). The other four inequalities in (2.20) to (2.24) can be derived similarly.

Proposition 2.3.1 is a direct result of Proposition 2.4.2 that will be formally described in Section 2.4.

Corollary 1. *For all the scenarios in which the capacity region is known, i.e., either (i) $K \leq 3$, or (ii) the symmetric channels, or (iii) the channel is spatially independent and the rate vector is one-sided fair (see [5] for detailed definitions), the above stability region matches the capacity region.*

The proof of Corollary 1 is relegated to Appendix A.2.1.

It is worth noting that the stable rate region in Proposition 2.3.1 meets the existing outer bound in Proposition 2.2.1 for many other scenarios not specified in Corollary 1. For example, we can examine the solution space of the LP problem in Proposition 2.3.1 by a brute-force but computer-aided search, which leads to

Example 1. *For $K = 4$ and spatially independent broadcast PEC with marginal success probability $\vec{p} = (p_1, p_2, p_3, p_4) = (\frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{4}{7})$, the rate vector $\vec{R} = (R_1, R_2, R_3, R_4) = (\frac{96}{1193}, \frac{672}{5965}, \frac{288}{1193}, \frac{1952}{8351})$ does not belong to any of the scenarios in Corollary 1. However, it is in the boundary of the outer bound (2.5) and also lies within the stable rate region in Proposition 2.3.1. Therefore, it is an optimal capacity vector for the given $\vec{p}$.*

The proof of Example 1 is relegated to Appendix A.2.2.

Example 1 is not unique in any sense. In fact, for $K = 4$ and the *spatially independent* channels, we have run 10^5 numerical trials with different $\vec{p}$ and we have not found any $\vec{R}$ that is in the outer bound but not in Proposition 2.3.1. However, if $K = 4$ and the underlying channel is not spatially independent, we have found some rate vectors $\vec{R}$ that are in the outer bound but not in the new stability region. One such example is provided in Appendix A.2.3.

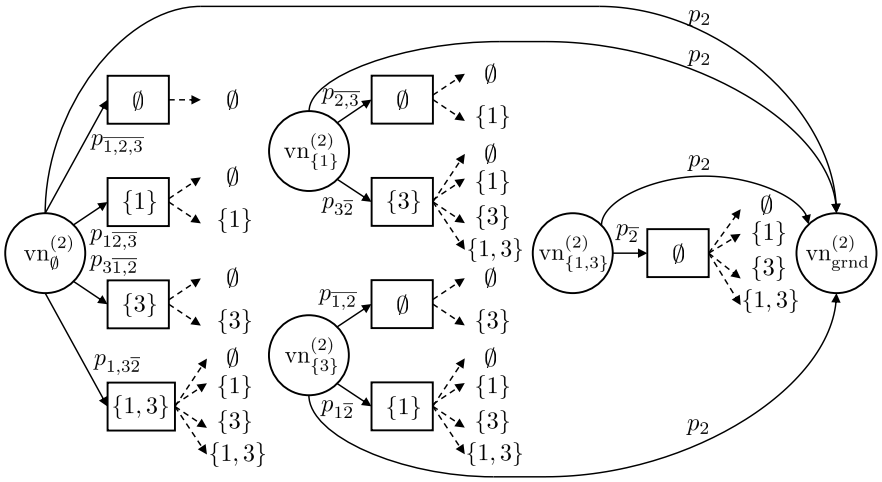


Figure 2.3. The virtual sub-network for $K = 3$ and $k = 2$, where the virtual nodes are represented by circles and the auxiliary nodes by squares. The dotted edge marked by a set S is actually connected to $\mathrm{vn}_S^{(2)}$ with ∞ capacity. For example, there are 4 outgoing edges of the auxiliary node (square) labeled by $\{1,3\}$. They are connected to $\mathrm{vn}_{\emptyset}^{(2)}$, $\mathrm{vn}_{\{1\}}^{(2)}$, $\mathrm{vn}_{\{3\}}^{(2)}$, and $\mathrm{vn}_{\{1,3\}}^{(2)}$, respectively, with each edge having infinite capacity. Note that the edge from auxiliary node $\{1,3\}$ to $\mathrm{vn}_{\emptyset}^{(2)}$ forms part of a *self loop* in the virtual network.

2.4 The New Achievability Scheme

In Section 2.4.1, we will provide a high-level max-flow/min-cut-based interpretation of the LP problem in Proposition 2.3.1. Later in Section 2.4.2 we will describe a sequential network coding scheme when assuming unlimited computing power. Finally, in Section 2.5 we will revise the scheme to take into account several practical considerations.

2.4.1 The Connection to the Virtual Network

We first introduce a virtual-network-based interpretation for the capacity inner bound in Proposition 2.3.1, which will shed some useful but very high-level intuition of the proposed design. It is worth noting that this virtual network representation is not needed when describing the proposed scheme. Therefore, some readers may choose to skip this subsection and directly start from the detailed scheme description in Section 2.4.2.

In Section 2.3, we have described the $2^{K-1}+1$ virtual nodes in the k-th virtual sub-network. We now describe its edges. See Fig. 2.3 for illustration.

For each $\text{vn}_S^{(k)}$ that is indexed by a subset $S \subseteq [K]\backslash\{k\}$, we add (i) an edge of capacity p_k that ends in $\text{vn}_{\text{grnd}}^{(k)}$; (ii) $2^{K-(|S|+1)}$ auxiliary nodes, each denoted by a subset $S_X \in 2^{[K]\backslash(S\cup\{k\})}$. For example suppose $K = 3$ and we focus on $\text{vn}_{\{1\}}^{(2)}$ with $S = \{1\}$. Since $[K]\backslash(S\cup\{k\}) = \{3\}$, we add two auxiliary nodes denoted by $S_X = \emptyset$ and $S_X = \{3\}$, respectively. To distinguish an auxiliary node from a regular virtual node, we represent the auxiliary nodes by squares in Fig. 2.3.

(iii) For each auxiliary nodes indexed by S_X, we add an edge connecting $\text{vn}_S^{(k)}$ and its auxiliary node S_X with capacity $p_{S_X\overline{[K]\backslash(S\cup S_X)}}$; (iv) Finally, for each auxiliary node S_X of $\text{vn}_S^{(k)}$, we add an edge of infinite capacity to virtual node $\text{vn}_{\tilde{S}}^{(k)}$ for all $\tilde{S} \subseteq (S\cup S_X)$. Take the virtual network in Fig. 2.3 for example, for which $K = 3$ and $k = 2$. Suppose we focus on $\text{vn}_{\{1\}}^{(2)}$ with $S = \{1\}$ and the corresponding auxiliary node indexed by $S_X = \emptyset$. Then we add an edge connecting $\text{vn}_{\{1\}}^{(2)}$ and its auxiliary node $S_X = \emptyset$ with capacity $p_{\overline{\{2,3\}}}$. Furthermore, we add two infinite-capacity edges from $S_X = \emptyset$ to $\text{vn}_{\emptyset}^{(2)}$ and $\text{vn}_{\{1\}}^{(2)}$, respectively. The description of the k-th virtual sub-network is complete.

In the following we describe how our new network coding scheme and its operations are mapped to the "packet movement" within the virtual network. For any subset $T \in 2^{[K]}$, it is well known that a single coded packet can deliver[6] $|T|$ different packets, one for each destination d_k, $k \in T$ if the following conditions hold: If for all $k \in T$ there exists a flow-k packet X_{k,i_k} that has previously been overheard by all other destinations $d_{\tilde{k}}$ for all $\tilde{k} \in T\backslash\{k\}$, then we can send a linear sum $\sum_{k\in T} X_{k,\text{i}_k}$ that benefits all d_k, $k \in T$, simultaneously. Roughly speaking, the value of x_T in (2.17) corresponds to the frequency of sending a linear sum of $|T|$ packets, one packet from each flow $k \in T$. Since each time slot can only choose one specific T, the frequencies $\{x_T\}$ must satisfy the time-sharing condition in (2.17).

The intuition behind Proposition 2.3.1 is that each virtual node $\text{vn}_S^{(k)}$ represents a queue that stores the flow-k packets that have been overheard by all $d_{\tilde{k}}$, $\tilde{k} \in S$. Since combining flows of $k \in T$ can simultaneously benefit all flows $k \in T$, a larger x_T value should help move more packets out of $\text{vn}_S^{(k)}$ where $S = T\backslash\{k\}$. That is why we scale the virtual network edge capacity of the k-th virtual sub-network, see (2.18), by $x_{S\cup\{k\}}$. A close look at (2.18) shows

[6]↑Obviously this statement assumes that the coded packet is received successfully for all d_k, $k \in T$.

that for any fixed T, x_T will appear in (2.18) for all $k \in T$, which reflects the fact that the coded transmission can simultaneously benefit all d_k with $k \in T$.

When a coded packet combined from the node (queue) $\mathrm{vn}_S^{(k)}$, $k \in T$, is received by destinations $\{d_{\mathrm{i}} : \mathrm{i} \neq k, \mathrm{i} \in S_{\mathrm{rx}}\}$, the coded packet can potentially move to the nodes (queues) $\mathrm{vn}_{\tilde{S}}^{(k)}$, $\tilde{S} \subseteq S \cup S_{\mathrm{rx}}$. Therefore we use the auxiliary node S_X to indicate the *additional overhearing set* $S_X = S_{\mathrm{rx}} \backslash T = S_{\mathrm{rx}} \backslash (S \cup \{k\})$ such that $S \cup S_{\mathrm{rx}} = S \cup S_X$. The probability that the additional overhearing set S_X is observed, computed by $\Pr(S_X \subseteq S_{\mathrm{rx}} \subseteq S_X \cup S) = p_{S_X \overline{[K] \backslash (S \cup S_X)}}$ in (2.2), is thus the capacity from the virtual node $\mathrm{vn}_S^{(k)}$ to its auxiliary node S_X. Since the flows out of the auxiliary nodes is already constrained by the flows into the auxiliary nodes, we place no capacity limit on the out links of the auxiliary nodes, thus the infinite capacity.

Before any further discussion, we provide the following lemmas.

Lemma 1. *For any cut $\mathcal{C}^{(k)}$, the left hand side of (2.18) represents the corresponding cut value in the k-th virtual sub-network, i.e., the summation of the capacity (scaled by the node scheduling frequency x_T) of the edges crossing the cut $\mathcal{C}^{(k)}$.*

Lemma 2. *The following two statements are equivalent. [Statement 1:] The minimum cut value of all cuts (not necessarily proper-cut) is no less than R_k; [Statement 2:] The minimum cut value of all proper cuts is no less than R_k.*

The proofs of Lemmas 1 and 2 are relegated to Appendix A.3.

By Lemma 1, (2.18) asserts that the minimum cut value of all proper cuts must be no less than R_k, the desired rate of the k-th session. Then by Lemma 2 it further implies that the minimum cut value of all cuts is no less than R_k. Finally, by the max-flow/min-cut theorem, we can find a max-flow that achieves R_k in the virtual sub-network. As a result, if we can establish the relationship between (i) the achievable rate region of a sequential network coding scheme and (ii) the max-flow values of the virtual network, then Proposition 2.3.1 can be used to characterize the achievable rate region of the given network coding scheme. This is the basic road map of our approach.

Remark: The concept of using virtual networks to represent network coding operations is not new. Nonetheless, our virtual network representation is significantly different from

existing ones, e.g., [18]. Firstly, the existing virtual network representation [18] uses 1-to-1 edges, i.e., each edge is from a single $\text{vn}_S^{(k)}$ to another $\text{vn}_{\tilde{S}}^{(k)}$. In contrast, by letting $\text{vn}_S^{(k)}$ first connect to an auxiliary node labeled by S_X and then adding edges of infinite capacity from S_X to many virtual nodes $\text{vn}_{\tilde{S}}^{(k)}$ for all $\tilde{S} \subseteq (S \cup S_X)$, we essentially create a 1-to-many *hyper-edge* from $\text{vn}_S^{(k)}$ to $\{\text{vn}_{\tilde{S}}^{(k)} : \tilde{S} \subseteq (S \cup S_X)\}$ for each S_X, see Fig. 2.3; Secondly, the virtual network in [18] is acyclic as the packets only move to queues of *higher levels*. Our construction contains many cycles and self-loops as illustrated in Fig. 2.3. This new cyclic, hyper-graph-based virtual network is a direct result of the new concept of opportunistic packet evolution.

2.4.2 A New Sequential Network Coding Scheme

We now present a new sequential coding scheme, the operation of which requires only the statistics $p_{S\overline{[K]\setminus S}}$ of the underlying PEC.

Our sequential network coding scheme has five components:

1. Maintain the queues and the global coding kernels.
2. Handle random sequential packet arrivals at time t.
3. Select a set $T^* \in 2^{[K]}$ of flows to be "added" together.
4. Generate the coded packet $Y(t)$.
5. Update the queues and the global coding kernels based on the reception status $S_{\text{rx}}(t)$ fed back from the destinations to the source.

Component 1: The queues and coding information at source. The discussion of this component can be further divided into two sub-components.

Component 1.1: Physical Memory Usage. Source s maintains $K \cdot 2^{K-1}$ queues, denoted by $Q_S^{(k)}$ for all $k \in [K]$ and $S \in 2^{[K]\setminus\{k\}}$. Each element of the queue $Q_S^{(k)}$ belongs to $\mathsf{GF}(q)$,

which represents the coded or uncoded entries (payload) for destination d_k that have been overheard[7] by destinations $\{d_{\mathrm{i}} : \mathrm{i} \in S\}$. Initially at time 0, all queues $Q_S^{(k)}$ are empty.

Component 1.2: Computation Task. Source s maintains $2K$ matrices: $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ for all $k \in [K]$. Unlike the queues described previously, these $2K$ matrices can be deterministically computed from the packet arrival processes $[A_k]_1^t$ and the past channel reception status $[S_{\mathrm{rx}}]_1^{t-1}$. Therefore, they can be computed *on-the-fly*, see (2.7), and no actual memory usage is needed. However, for the ease of exposition we describe these matrices as if they are being stored in memory.

Specifically, each $\mathbf{U}^{(k)}$ matrix stores the *global encoding kernels* [29] of the coded packets in the queues $\{Q_S^{(k)}\}$. The number of rows of $\mathbf{U}^{(k)}$ is thus equal to $\sum_S |Q_S^{(k)}|$. Each $\mathbf{V}^{(k)}$ matrix stores the global encoding kernels of all the packets that have been successfully delivered to d_k. The number of rows of $\mathbf{V}^{(k)}$ is equal to $\sum_{\tau=1}^{t} 1_{\{k \in S_{\mathrm{rx}}(\tau)\}}$.

The detailed update rules of $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ and $Q_S^{(k)}$ will be discussed in Components 2 and 5. Initially at time 0, these $2K$ matrices are 0×0 matrices.

Component 2: Sequential packet arrivals. Whenever the i-th flow-k information packet arrives at s, denoted by $X_{k,\mathrm{i}}$, we store it in the queue $Q_{\emptyset}^{(k)}$ by creating a new entry $W_{\mathsf{ptr}} = X_{k,\mathrm{i}} \in \mathsf{GF}(q)$ where ptr is the pointer of the new entry W_{ptr}. The queue update is now complete. We note that the pointer ptr can be of any arbitrary format. One convenient choice is to let $\mathsf{ptr} = (k, \mathrm{i})$ being a vector (k, i). Note that even though when a packet arrives, the newly queued element $W_{(k,\mathrm{i})}$ is identical to the newly arrived $X_{k,\mathrm{i}}$, the other queued element $W_{(k,\mathrm{i})}$ may not be equal to $X_{k,\mathrm{i}}$. *The pointer* $\mathsf{ptr} = (k, \mathrm{i})$ *(or* $\mathsf{ptr} = (k, \mathrm{i})$*) only serves as a unique pointer/index to the queued element and does not necessarily dictate the content of the queued element.* For example, we may have $W_{(2,3)} = X_{2,3} + X_{1,4}$, which means that the queue entry being referred by the pointer $(k, \mathrm{i}) = (2, 3)$ is a linear sum of $X_{2,3}$, the third packet of user d_2, and $X_{1,4}$, the fourth packet of user d_1.

We now describe how to update the matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$. We first add an all-zero column vector to the right of all $2K$ matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$. Namely, all $2K$ matrices are

[7]↑A more precise statement would be "*non-interfering* to destinations $\{d_{\mathrm{i}} : \mathrm{i} \in S\}$" rather than "overheard by $\{d_{\mathrm{i}} : \mathrm{i} \in S\}$". The subtle difference will be explained in Appendix A.5. For ease of exposition, we use the term "overheard" in this subsection.

getting wider by one column. The number of rows of the added column vector should be clear from the context. Intuitively, the new column corresponds to the newly arrived packet $X_{k,\mathrm{i}}$.

Let $\mathbf{e}_{(k,\mathrm{i})} = (0, \ldots, 0, 1)$ denote a row vector with value 1 in the last position, the position corresponding to the new column (k, i), and 0s otherwise. We then add $\mathbf{e}_{(k,\mathrm{i})}$ to the bottom of $\mathbf{U}^{(k)}$. Namely, $\mathbf{U}^{(k)}$ is now getting both taller and wider since one new packet $X_{k,\mathrm{i}}$ now appears in queue $Q_{\emptyset}^{(k)}$ of user-k. All the other $\mathbf{U}^{(k)}$ with $k \neq k$ just get wider (due to the added column) but the heights remain the same.

Note that for ease of exposition, we use the same pointer $\mathsf{ptr} = (k, \mathrm{i})$ to refer to either an entry in $Q_S^{(k)}$ or the corresponding row in $\mathbf{U}^{(k)}$. For example, the actual content of the W_{ptr} is equal to the vector product $W_{\mathsf{ptr}} = \mathsf{row}_{\mathsf{ptr}}(\mathbf{U}^{(k)}) * \vec{\mathbf{X}}_{\mathsf{arr}}$ where $\mathsf{row}_{\mathsf{ptr}}(\mathbf{U}^{(k)})$ is the row vector of $\mathbf{U}^{(k)}$ referred to by ptr and $\vec{\mathbf{X}}_{\mathsf{arr}}$ is a column vector that consists of all the packets that have arrived according to its arrival order (namely, according to the order when the columns were created.) Note that $\vec{\mathbf{X}}_{\mathsf{arr}}$ contains the packets of all K users, not just user d_k. The reason is that our scheme performs intersession coding over all K users' packets.

Component 3: Selecting a set $T^* \in 2^{[K]}$ of flows to be added together. For each time t, we select a new T^* as follows. For all $k \in [K]$ and all $\check{S} \in 2^{[K]\setminus\{k\}}$, we compute

$$q(k, \check{S}) \triangleq \min_{\tilde{S} \in 2^{\check{S}}} \left| Q_{\tilde{S}}^{(k)} \right|, \tag{2.25}$$

where $\left|Q_{\tilde{S}}^{(k)}\right|$ is the number of entries in $Q_{\tilde{S}}^{(k)}$. The minimum operation in (2.25) given k and $\check{S}$ is to compute the minimum queue length associated to the hyper edge that reaches all $\tilde{S} \in 2^{\check{S}}$ in the flow-k virtual sub-network. Also see the discussion in Section 2.4.1. Since a hyper edge originated from $\mathrm{vn}_S^{(k)}$ through auxiliary node S_X can reach all virtual nodes $\mathrm{vn}_{\tilde{S}}^{(k)}$ satisfying $\tilde{S} \subseteq \check{S} = S \cup S_X$, by setting $\check{S} = S \cup S_X$, the expression $q(k, S \cup S_X)$ computes the minimum[8] of the queue lengths of all the destinations of $\mathrm{vn}_S^{(k)}$ when the additional overhearing set is S_X. Note that the additional overhearing is a random set. Since the backpressure is defined

[8]↑The reason we choose the minimum is mostly based on the corresponding Lyapunov drift analysis in Appendix A.6. Intuitively, since we can opportunistically decide to which virtual node of the hyper edge should we move the packet, we should move the packet to the one with the smallest queue length (i.e., the destination node currently experiences the largest backpressure), and thus the minimum operation.

Table 2.2. A $K = 3$ example of computing $q(k, S)$ and the backpressure term $\mathrm{bp}(Q_S^{(k)})$ from $\left|Q_S^{(k)}\right|$.

(k, S)	$\left\|Q_S^{(k)}\right\|$	$q(k, S)$	$\mathrm{bp}(Q_S^{(k)})$
$(1, \emptyset)$	1	1	0.84
$(1, \{2\})$	2	1	1.90
$(1, \{3\})$	1	1	0.88
$(1, \{2, 3\})$	0	0	0
$(2, \emptyset)$	5	5	3.68
$(2, \{1\})$	3	3	1.80
$(2, \{3\})$	1	1	0.40
$(2, \{1, 3\})$	2	1	1.40
$(3, \emptyset)$	3	3	1.90
$(3, \{1\})$	2	2	1.00
$(3, \{2\})$	4	3	2.90
$(3, \{1, 2\})$	2	2	1.00

as the difference between the queue length of a given node and the (expected) queue lengths of all its next-hop neighbors, we compute the backpressure by

$$\mathrm{bp}\left(Q_S^{(k)}\right) \triangleq \left|Q_S^{(k)}\right| - \sum_{S_X \in 2^{[K]\setminus(S\cup\{k\})}} p_{S_X \overline{[K]\setminus(S\cup S_X)}} q(k, S \cup S_X). \tag{2.26}$$

The target set T^* is then selected as follows.

$$T^* = \underset{\text{non-empty } T\in 2^{[K]}}{\arg\max} \sum_{k\in T} \mathrm{bp}\left(Q_{T\setminus\{k\}}^{(k)}\right). \tag{2.27}$$

That is, if we choose to transmit the sum of the packets in queues $\{Q_{T\setminus\{k\}}^{(k)} : k \in T\}$ for some T, then such coded transmission can benefit all d_k with $k \in T$. Therefore, their individual backpressures are also summed together. Eq. (2.27) chooses the T^* with the largest sum.

For example, suppose $K = 3$ and the channel is spatially independent with marginal success probability $(p_1, p_2, p_3) = (0.8, 0.4, 0.5)$. At time t, suppose the lengths of $K \cdot 2^{K-1} = 12$

queues are listed as in the second column of Table 2.2. We can calculate the corresponding $q(k,S)$ values by (2.25) and the $\mathrm{bp}(Q_S^{(k)})$ by (2.26). For example

$$\begin{aligned} q(2,\{1,3\}) &= \min_{\tilde{S}\subseteq\{1,3\}} \left|Q_{\tilde{S}}^{(2)}\right| \\ &= \min\left\{\left|Q_{\emptyset}^{(2)}\right|, \left|Q_{\{1\}}^{(2)}\right|, \left|Q_{\{3\}}^{(2)}\right|, \left|Q_{\{1,3\}}^{(2)}\right|\right\} = 1 \end{aligned}$$

and

$$\mathrm{bp}\left(Q_{\{3\}}^{(2)}\right) = \left|Q_{\{3\}}^{(2)}\right| - p_{\overline{\{1,2\}}} q(2,\{3\}) - p_{1\bar{2}} q(2,\{1,3\}) = 0.4.$$

With the $\mathrm{bp}(Q_S^{(k)})$ values, we calculate $\sum_{k\in T}\mathrm{bp}(Q_{T\setminus\{k\}}^{(k)})$ for all $T=\{1\}$, $\{2\}$, $\{3\}$, $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, $\{1,2,3\}$ and they are $0.84, 3.68, 1.90, 3.70, 1.88, 3.30, 2.40$ respectively. Finally we choose the target set $T^*=\{1,2\}$, which has the maximum backpressure sum 3.70.

Component 4: Generating the coded packet. After the set T^* is decided, for each $k\in T^*$, we extract the head-of-line element $W_{(k,\mathrm{j}_k)}$ from queue $Q_{T^*\setminus\{k\}}^{(k)}$. If queue $Q_{T^*\setminus\{k\}}^{(k)}$ is empty, we simply set the head-of-line element to be $W_{(k,0)} \triangleq 0$, the zero/null packet. The to-be-transmitted coded packet $Y(t)$ is computed by

$$Y(t) = \sum_{k\in T^*} \beta_k \cdot W_{(k,\mathrm{j}_k)} \tag{2.28}$$

That is, the transmitting packet is the linear sum of the head-of-line packets multiplied by β_k. Define

$$\mathbf{y}(t) = \sum_{k\in T^*} \beta_k \cdot \mathsf{row}_{(k,\mathrm{j}_k)}(\mathbf{U}^{(k)}). \tag{2.29}$$

Recall that $\mathsf{row}_{(k,\mathrm{j}_k)}(\mathbf{U}^{(k)})$ is the global coding kernel of the stored packet $W_{(k,\mathrm{j}_k)}$. $\mathbf{y}(t)$ is thus the global coding kernel for the coded transmission $Y(t)$. Note that only $Y(t)$ is transmitted and we do not transmit $\mathbf{y}(t)$.

For ease of exposition, one may assume β_k is chosen uniformly randomly from a sufficiently large $\mathsf{GF}(q)$ and skip to Component 5 directly. This simple random coding scheme

will work if $q = \infty$. On the other hand, the following paragraphs present a deterministic construction that holds for any finite $q \geq K$.

For all $k \in T^*$, construct a matrix $\mathbf{U}_{\mathrm{j}_k}^{(k)}$ by removing $\mathsf{row}_{(k,\mathrm{j}_k)}(\mathbf{U}^{(k)})$ corresponding to $W_{(k,\mathrm{j}_k)}$ from $\mathbf{U}^{(k)}$. The coefficients $\{\beta_k : k \in T^*\}$ can be deterministically chosen by following two steps. Step 1: Check whether $\mathsf{row}_{(k,\mathrm{j}_k)}(\mathbf{U}^{(k)})$ is linearly dependent of the rows of $((\mathbf{U}_{\mathrm{j}_k}^{(k)})^{\mathsf{T}}, (\mathbf{V}^{(k)})^{\mathsf{T}})^{\mathsf{T}}$, where $((\mathbf{U}_{\mathrm{j}_k}^{(k)})^{\mathsf{T}}, (\mathbf{V}^{(k)})^{\mathsf{T}})^{\mathsf{T}}$ is the vertical concatenation of matrices $\mathbf{U}_{\mathrm{j}_k}^{(k)}$ and $\mathbf{V}^{(k)}$. If so, then set $\beta_k = 0$. Namely, this is a degenerate case and the corresponding $W_{(k,\mathrm{j}_k)}$ will no longer participate in the coding computation. Perform this dependency check for all $k \in T^*$ and denote by T those non-degenerate k for which the β_k values are still undecided.

Step 2: Find arbitrarily one set of $\{\beta_k : k \in T\}$ values satisfying the following property: Define $\mathbf{y}(t) \triangleq \sum_{k \in T} \beta_k \cdot \mathsf{row}_{(k,\mathrm{j}_k)}(\mathbf{U}^{(k)})$. For all $k \in T$, vector $\mathbf{y}(t)$ has to be linearly independent of the rows of $((\mathbf{U}_{\mathrm{j}_k}^{(k)})^{\mathsf{T}}, (\mathbf{V}^{(k)})^{\mathsf{T}})^{\mathsf{T}}$. Lemma 5 in Appendix A.4 guarantees the existence of such coefficients $\{\beta_k : k \in T\}$. The description of how to generate the coded packet $Y(t)$ is complete.

Component 5: Packet movement among the queues — The opportunistic packet evolution component. We describe the packet movement for one specific k_0 value while the same mechanism has to be applied to all $k \in [K]$.

If $k_0 \notin T^*$, then no entry in $\{Q_S^{(k_0)} : \forall S\}$ will move. If $k_0 \in T^*$, then we consider one and only one queue: $Q_{T^* \setminus \{k_0\}}^{(k_0)}$. Recall that $W_{(k_0,\mathrm{j}_{k_0})}$ is the head-of-line packet of $Q_S^{(k_0)}$ selected in Component 4. In the degenerate case $Q_{T^* \setminus \{k_0\}}^{(k_0)} = \emptyset$ or equivalently $W_{(k_0,\mathrm{j}_{k_0})} = 0$, then no further action will be needed. Component 5 is complete. We go back to Component 2 for time $t + 1$.

In a non-degenerate case, that is $Q_{T^* \setminus \{k_0\}}^{(k_0)} \neq \emptyset$, we first remove the entry $W_{(k_0,\mathrm{j}_{k_0})}$ of $Q_{T^* \setminus \{k_0\}}^{(k_0)}$ as well as the corresponding $\mathsf{row}_{(k_0,\mathrm{j}_{k_0})}(\mathbf{U}^{(k_0)})$. If d_{k_0} has received the coded transmission $Y(t)$, then insert the global coding kernel $\mathbf{y}(t)$ of $Y(t)$ as a new row to the matrix $\mathbf{V}^{(k_0)}$. If d_{k_0} did not receive $Y(t)$, then we insert row vector $\mathbf{y}(t)$ as a new row to matrix $\mathbf{U}^{(k_0)}$ and insert the coded content $Y(t)$ to a new destination queue $Q_{\tilde{S}^*}^{(k_0)}$. Effectively, the location of the old entry $W_{(k_0,\mathrm{j}_{k_0})}$ has been moved from the old queue $Q_{T^* \setminus \{k_0\}}^{(k_0)}$ to the new queue $Q_{\tilde{S}^*}^{(k_0)}$ while the content is updated from the old $W_{(k_0,\mathrm{j}_{k_0})}$ to the new $Y(t)$.

The new destination queue $Q^{(k_0)}_{\tilde{S}^*}$ is decided as follows. Define $\check{S} = (T^*\backslash\{k_0\}) \cup S_{\rm rx}(t)$ where the reception set $S_{\rm rx}(t)$ is obtained from ACK/NACK. Define $\tilde{S}^*$ as the $\tilde{S}$ that minimizes the expression $q(k_0, \check{S})$ in (2.25). Then we choose $Q^{(k_0)}_{\tilde{S}^*}$ as the destination queue.

Component 5 is the most distinct feature of our scheme. Unlike existing solutions that choose the destination queue as a deterministic function of the reception set $S_{\rm rx}(t)$ (usually in an acyclic fashion), Component 5 decides the movement *opportunistically* with the help of $S_{\rm rx}(t)$ and the neighbors' queue lengths, possibly in a cyclic way. That is, it is possible that $\tilde{S}^* = T^*\backslash\{k_0\}$ or even $\tilde{S}^* \subsetneq T^*\backslash\{k_0\}$. Since the newly stored entry $Y(t)$ in $Q^{(k_0)}_{\tilde{S}^*}$ differs from the older entry $W_{(k_0,{\rm j}_{k_0})}$, the stored packet *evolves* over time. Jointly these two features give the name of *opportunistic packet evolution* of our design.

We use the following example to illustrate the operations in Component 5. Consider $K = 3$ and suppose that at time t, the target set is $T^* = \{1, 2, 3\}$. Also suppose after transmission, the reception set is $S_{\rm rx}(t) = \{3\}$. We now discuss how the packets move within the queues.

Consider the packets for d_2, i.e., $k_0 = 2$. The head-of-line packet $W_{(2,{\rm j}_2)}$ is first removed from queue $Q^{(2)}_{\{1,3\}}$ and the corresponding row is removed from $\mathbf{U}^{(2)}$. Then, suppose the flow-2 queue lengths are $|Q^{(2)}_{\{\emptyset\}}| = 5$, $|Q^{(2)}_{\{1\}}| = 3$, $|Q^{(2)}_{\{3\}}| = 1$, and $|Q^{(2)}_{\{1,3\}}| = 2$. Since $\check{S} = (T^*\backslash\{k_0\}) \cup S_{\rm rx}(t) = \{1, 3\}$ we compute

$$q(2, \{1, 3\}) = \min\left\{|Q^{(2)}_{\{\emptyset\}}|, |Q^{(2)}_{\{1\}}|, |Q^{(2)}_{\{3\}}|, |Q^{(2)}_{\{1,3\}}|\right\} = 1$$

and the minimizer $\tilde{S}^* = \{3\}$. The new packet $Y(t)$ is then injected to queue $Q^{(k_0)}_{\tilde{S}^*} = Q^{(2)}_{\{3\}}$ and $\mathbf{y}(t)$ is inserted back to $\mathbf{U}^{(2)}$.

For the other destination d_3, i.e., $k_0 = 3$. the head-of-line entry $W_{(3,{\rm j}_3)}$ is first removed from $Q^{(3)}_{\{1,2\}}$ and the corresponding row is removed from $\mathbf{U}^{(3)}$. Since d_3 received the packet (because $S_{\rm rx}(t) = \{3\}$), we inject $\mathbf{y}(t)$ to matrix $\mathbf{V}^{(3)}$. The packet movement for destination d_1 is similar to that of d_2.

The proposed scheme operates by repeatedly executing Components 2 to 5 for each time slot. We can then prove the following results.

Proposition 2.4.1. *The above 5-component sequential coding scheme satisfies the decodability condition in* (2.9) *and* (2.10).

Proposition 2.4.2. *The proposed 5-component sequential coding scheme can stabilize any given rate vector* $\vec{R} = (R_1, \cdots, R_K)$ *if* $\vec{R}$ *satisfies Proposition 2.3.1.*

Proposition 2.4.3. *The proposed 5-component sequential coding scheme can stabilize* $\vec{R} = (R_1, \cdots, R_K)$ *only if* $\vec{R}$ *satisfies Proposition 2.3.1, provided we replace the strict inequality in* (2.17) *by* $\leq$.

The proof of Proposition 2.4.1 is provided in Appendix A.5. The proofs of Propositions 2.4.2 and 2.4.3 are relegated to Appendix A.6.

Remark: The proposed scheme allows cyclic packet movement and thus packets sometimes move from $Q_{S_1}^{(k)}$ to $Q_{S_2}^{(k)}$ with $S_1 \supsetneq S_2$. This is in sharp contrast with the schemes in [5], [18], which always move packets to queues of larger overhearing sets (thus being acyclic). The drawback of the strict acyclic packet movement is that when deciding to send a coding operation that combines all flows $k \in T$, the queue lengths of the $|T|$ participating virtual queues $\{Q_{T\backslash\{k\}}^{(k)} : k \in T\}$ may be highly unbalanced since many packets may have already moved out of $Q_{T\backslash\{k_0\}}^{(k_0)}$ for a specific k_0. To mitigate the unbalanced queue lengths, the schemes in [5], [18] searches for *collapsed overhearing set matching* (COSM) when the encoding set being T. That is, when a flow-k_0 queue $Q_{T\backslash\{k_0\}}^{(k_0)}$ is empty, COSM chooses the packet from a queue $Q_S^{(k_0)}$ with strictly larger overhearing set $S \supsetneq T\backslash\{k_0\}$. This substitution effectively moves some packets from $Q_S^{(k_0)}$ to $Q_{T\backslash\{k_0\}}^{(k_0)}$ to create more overhearing set matching opportunities.

The performance improvement of COSM is due to its exhaustive examination of all possible coding combinations by temporarily suppressing some of the overhearing sets. However, searching for COSM is of exceedingly high complexity. To mitigate the complexity, the proposed scheme uses backpressure to balance virtual queue lengths preemptively so that the packets are evenly distributed among all virtual queues. As a result, when choosing a coding decision T, we only need to combine packets from $\{Q_{T\backslash\{k\}}^{(k)} : k \in T\}$ without resorting to expensive search of COSM. Note that the proposed backpressure scheme continuously rebalances the queue lengths for each time slot based on the feedback $S_{\text{rx}}(t)$, and the rebalancing sometimes moves packets to queues of smaller overhearing subsets in order to keep those

queues from depletion. One contribution of this work is to prove that such a counterintuitive packet movement, once performed optimally, attains the same performance as the more complicated COSM-based schemes.

2.5 Practical Sequential Coding Scheme with Overhead

In Section 2.2, we assume *network-wide* channel output feedback $S_{\text{rx}}(t)$ and traffic pattern $\vec{A}(t)$ are causally available to all the destinations and we assume that source s is of unlimited computing power/memory so that the global encoding kernel matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ can either be computed on-the-fly or be completely stored with no penalty. In this section, we modify the previous scheme to handle the lack of the knowledge of $S_{\text{rx}}(t)$ and $\vec{A}(t)$ at d_k and the limited computing power and memory at s in practice. The modification incurs additional overhead that is absent in the previous theoretic setting and we examine the overhead by simulation in Section 2.6.

2.5.1 A Header-based Implementation

A standard approach in practical network coding protocols is to append a *header* to each of the packets in the queue $Q_S^{(k)}$, which contains the *global encoding kernel*. The global coding vector $\mathbf{y}(t)$ (the header) is transmitted together with the payload $Y(t)$. Destination d_k can then decode the original message packets $X_{k,\text{i}}$, $\text{i} = 1, 2, \cdots$ using the global coding vector $\mathbf{y}(t)$.

However, recall that the global encoding kernel of any entry $W_{(k,\text{i})}$ is $\mathsf{row}_{(k,\text{i})}(\mathbf{U}^{(k)})$, the row vector of the matrix $\mathbf{U}^{(k)}$, and per our construction its dimension grows every time there is a new packet arrival. The overhead of sending the header thus increases over time. Furthermore, to implement the above naive scheme, matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ need to be stored in s and again the increasing size of $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ requires memory usage that grows linearly with respect to time. In the sequel, we focus on how one can "prune" the matrices in practice.

Our modification contains two parts. First, we let each queuing entry $W_{(k,\text{i})} = (\mathsf{H}, \mathsf{PL})$ contain two parts, a header H and payload PL. The header H consists of the global encoding kernel $\mathsf{row}_{(k,\text{i})}(\mathbf{U}^{(k)})$. However, since the dimension of $\mathsf{row}_{(k,\text{i})}(\mathbf{U}^{(k)})$ grows with time, we take

advantage of the observation that the vector $\mathsf{row}_{(k,\mathrm{i})}(\mathbf{U}^{(k)})$ is sparse and let H store the lossless compressed version of $\mathsf{row}_{(k,\mathrm{i})}(\mathbf{U}^{(k)})$. Specifically, we let H contain a set of tuples $(k, \mathrm{i}, \beta_{k,\mathrm{i}})$, where $\beta_{k,\mathrm{i}}$ is the global encoding coefficient multiplied to the i-th packet of destination k. Namely, we have

$$\mathsf{PL} = \sum_{(k,\mathrm{i},\beta_{k,\mathrm{i}})\in\mathsf{H}} \beta_{k,\mathrm{i}} X_{k,\mathrm{i}}. \tag{2.30}$$

Since the header only records the coefficients of all participating packets, the header size is small for sparse coding vectors. Since in the original scheme, the $\mathbf{U}^{(k)}$ matrices store all the global coding vectors for all the entries of $Q_S^{(k)}$ for d_k, this lossless compression effectively replaces/subsumes the $\mathbf{U}^{(k)}$ matrices and we do not need to store $\mathbf{U}^{(k)}$ anymore.

Even though we do not need to store the $\mathbf{U}^{(k)}$ matrix, we still need to store and constantly update the $\mathbf{V}^{(k)}$ matrix for all $k \in [K]$. Define $\mathcal{HS}$ as the collection of the headers of all queueing entries stored in the $K \cdot 2^{K-1}$ virtual queues $\{Q_S^{(k)} : k \in [K], S \in 2^{[K]\setminus\{k\}}\}$. The pseudo code in Algorithm 1 describes the corresponding operations of updating and pruning the $\mathbf{V}^{(k)}$ matrix.

After pruning matrix $\mathbf{V}^{(k)}$, recall that when choosing the coding coefficients $\{\beta_k\}$ in Component 4, we have to find β_k such that the resulting $\mathbf{y}(t)$ in (2.29) is independent of the row of $((\mathbf{U}_{\mathrm{j}_k}^{(k)})^\mathsf{T}, (\mathbf{V}^{(k)})^\mathsf{T})^\mathsf{T}$ for all k. Let $\tilde{\mathbf{V}}^{(k)}$ denote the output matrix after applying the above pruning procedure to the original $\mathbf{V}^{(k)}$ matrix. Recall that $\mathbf{U}^{(k)}$, $\mathbf{U}^{(k)}$, and $\mathbf{V}^{(k)}$ in Component 4 have the same set of columns while $\tilde{\mathbf{V}}^{(k)}$ only keeps a subset of the columns of $\mathbf{V}^{(k)}$. If we keep the columns of the $\mathbf{U}^{(k)}$ and $\mathbf{U}^{(k)}$ matrices that also appear in $\tilde{\mathbf{V}}^{(k)}$ and denote the resulting matrices and global coding kernel as $\tilde{\mathbf{U}}^{(k)}$, $\tilde{\mathbf{U}}^{(k)}$, and $\tilde{\mathbf{y}}(t)$, respectively, then we have the following lemma.

Lemma 3. $\mathbf{y}(t) \in \langle \mathbf{U}^{(k)} \rangle \oplus \langle \mathbf{V}^{(k)} \rangle$ *if and only if* $\tilde{\mathbf{y}}(t) \in \langle \tilde{\mathbf{U}}^{(k)} \rangle \oplus \langle \tilde{\mathbf{V}}^{(k)} \rangle$.

The proof of Lemma 3 is relegated to Appendix A.7.

That is, instead of checking whether the coding vector belongs to the row space of the three original matrices $\mathbf{U}^{(k)}$, $\mathbf{U}^{(k)}$, and $\mathbf{V}^{(k)}$ of which the size grows linearly with respect to time, now we only need to generate the coding vector $\tilde{\mathbf{y}}(t)$ according to the pruned versions

Algorithm 1 Pruning the $\mathbf{V}^{(k)}$ Matrix at time t

input $\{\mathbf{V}^{(k)} : k \in [K]\}$ and $\mathcal{HS}$.
Recall that each header $\mathsf{H} \in \mathcal{HS}$ contains a variable number of tuples $(k, \mathrm{i}, \beta_{k,\mathrm{i}})$. Temporarily ignore the last coordinate $\beta_{k,\mathrm{i}}$ in the tuple and define $\mathcal{J}_U \triangleq \{(k, \mathrm{i}) : \forall (k, \mathrm{i}) \in \mathsf{H}, \mathsf{H} \in \mathcal{HS}\}$ as the collection of all (k, i) that appear in the stored headers.
Define $\mathcal{N}_0$ as the collection of (k, i) simultaneously satisfying (i) packet $X_{k,\mathrm{i}}$ has already arrived, and (ii) $(k, \mathrm{i}) \notin \mathcal{J}_U$. Namely, $\mathcal{N}_0$ contains those (k, i) that no longer appears in any of the headers.
for $(k_0, \mathrm{i}_0) \in \mathcal{N}_0$ **do**
  for $k \in [K]$ **do**
    if the (k_0, i_0) column has not been pruned from $\mathbf{V}^{(k)}$ before **then**
      Swap the (k_0, i_0) column of $\mathbf{V}^{(k)}$ with its leftmost column. Namely, shift the (k_0, i_0) column to the left as the first column of $\mathbf{V}^{(k)}$.
      if the first column of the new $\mathbf{V}^{(k)}$ is not a zero column **then**
        Apply Gaussian elimination to $\mathbf{V}^{(k)}$ so that the first column becomes $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$.
        Remove the first row of the resulting $\mathbf{V}^{(k)}$.
      end if
      Remove the first column of $\mathbf{V}^{(k)}$.
    end if
    Apply Gaussian elimination to $\mathbf{V}^{(k)}$ so that the resulting $\mathbf{V}^{(k)}$ is of the row-echelon form. Remove all the zero rows of $\mathbf{V}^{(k)}$.
  end for
end for
return the final matrices $\{\mathbf{V}^{(k)} : k \in [K]\}$.

$\tilde{\mathbf{U}}^{(k)}$, $\tilde{\mathbf{U}}^{(k)}$, and $\tilde{\mathbf{V}}^{(k)}$. (Matrices $\tilde{\mathbf{U}}^{(k)}$ and $\tilde{\mathbf{U}}^{(k)}$ are derived implicitly from the header of the entries in $Q_S^{(k)}$.)

2.5.2 Other Issues for Practical Implementation

The benefits of the modified scheme in Section 2.5.1 are two-fold. Firstly, the source node uses packet headers to reduce its encoding complexity and memory usage. Secondly, the destination d_k does not need the knowledge of the network-wide reception status $S_{\text{rx}}(t)$ and only needs the coding vectors in the headers during decoding.

In this subsection, we discuss some other practical issues that may arise during implementation. They are beyond the scope of this theoretical study and we thus only outline some possible solutions. The designs discussed in this subsection will not be used during the simulation in Section 2.6.

Unknown Channel Statistic

All our results assume that the channel statistics $p_{S\overline{[K]\setminus S}}$ is stationary and known to the source. When $p_{S\overline{[K]\setminus S}}$ is unknown, we can estimate $p_{S\overline{[K]\setminus S}}$ on-the-fly via the history of the ACK/NACK feedback. The estimation $\hat{p}_{S\overline{[K]\setminus S}}$ can then be used as a substitute of the true $p_{S\overline{[K]\setminus S}}$ during the backpressure computation. If the channel statistics are indeed stationary, then the empirical estimation will eventually converge to the true distribution.

Variable Header Length

The header-based scheme in Section 2.5.1 allows for variable header lengths, which may grow unboundedly as time progresses, even though in simulation it grows quite slowly. To address this potential unboundedness, during implementation we may set an upper limit such that whenever a queueing entry has header length exceeding that limit, the source node would stop the normal coding operations and transmit that entry uncodedly until it is received by its target destination. In this way, the header length can be kept strictly bounded away from infinity.

Delay

As will be verified by simulation in Section 2.6, the proposed scheme has reasonable delay and queue length performance when operating away from the capacity. However, when we are getting closer and closer to the capacity, inevitably the queue lengths will grow to infinity just like any queueing system that operates closely to its theoretic limit. Furthermore, our scheme also exhibits the so-called *decoding delay*, which again grows unboundedly when operating arbitrarily close to capacity.

One solution that mitigates this fundamental phenomenon is *admission control*. Periodically, we could stop admitting new packets and focus on sending all packets in the current queues and emptying the queues completely. By periodically suspending the incoming packets and emptying the queues, we can ensure that the decoding and queueing delay remain bounded all the time. Like all other admission control schemes, the cost of this scheme is that the achievable throughput decreases since no new packets are admitted during suspension. There are many variants how the admission control can be better implemented, e.g., statistically or deterministically, which is beyond the scope of this work.

Deadline

It is possible that each individual packet $X_{k,\mathrm{i}}$ has some deadline, i.e., it needs to be delivered by a certain amount of time. One solution to improve the deadline performance is that when a packet has been in the queues for too long and is about to expire, we can give it higher scheduling priority and temporarily overwrite the throughput-optimal backpressure scheduler described in Section 2.4.2. In this way, we can control the deadline at the cost of reduced throughput.

2.6 Simulation Results

In this section, we assume that the arrival of the flow-k packets is i.i.d. Poisson with rate R_k. Assume $K = 4$ and spatially independent PEC with marginal success probability $(p_1, p_2, p_3, p_4) = (\frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{4}{7})$. Example 1 shows that $\vec{R}^* \triangleq (R_1^*, R_2^*, R_3^*, R_4^*) = (\frac{96}{1193}, \frac{672}{5965}, \frac{288}{1193}, \frac{1952}{8351})$

is at the boundary of the true capacity region. We then set the actual arrival rate to be $\vec{R}_\alpha = (R_1, R_2, R_3, R_4) = \alpha\vec{R}^*$ where $0 < \alpha \leq 1.03$ measures how closely we are operating at the capacity. Each simulation trial lasts for 10^5 time slots. For reference, we use the same (p_1, p_2, p_3, p_4) value and compute the largest α that can possibly be achieved by a traditional non-network-coded scheme. That is, the largest α satisfying

$$\frac{\alpha R_1^*}{p_1} + \frac{\alpha R_2^*}{p_2} + \frac{\alpha R_3^*}{p_3} + \frac{\alpha R_4^*}{p_4} \leq 1 \tag{2.31}$$

is $\alpha^*_{\text{time.sharing}} = \frac{1193}{1688} \simeq 0.71$.

To measure the sum of queue lengths $\sum_{k,S} |Q_S^{(k)}|$, we first fix the α value being considered. Then for each trial we compute the time average of $\sum_{k,S} |Q_S^{(k)}|$ over the 10^5 time slots. We repeat 100 trials for any given α value and Fig. 2.4 reports the average queue length for different α values with the corresponding 95% confidence intervals. For any $\alpha < 1$, the sum of queue lengths remains very small even if we continue the simulation trial beyond 10^5 time slots. If $1 < \alpha$, then the queue lengths grow linearly with respect to the number of time slots. Note that the sum of queue lengths corresponds to how many packets the source s needs to store in the memory during sequential encoding, which is around 54.9 packets when $\alpha = 0.95$. It is worth noting that the queue lengths of any uncoded scheme will become unbounded whenever $\alpha > \alpha^*_{\text{time.sharing}} = 0.71$.

The control overhead is measured by the length of the header of each transmission. We noticed that in our header-based scheme in Section 2.5.1, the header length grows whenever we mix multiple packets together. Recall that the virtual packet movement is originated from the queue $Q_S^{(k)}$ and ends in the destination queue $Q_{\tilde{S}^*}^{(k)}$. To further control the growth rate of the header length, we notice that if $\tilde{S}^* \subseteq S$, then instead of injecting a new packet $Y(t)$ with coding vector $\mathbf{y}(t)$ into the destination queue $Q_{\tilde{S}^*}^{(k)}$, we can simply inject the original packet $W_{(k,\mathrm{i})}$ packet with coding vector $\mathsf{row}_{(k,\mathrm{i})}(\mathbf{U}^{(k)})$ into the same destination queue $Q_{\tilde{S}^*}^{(k)}$. The reason is that the overhearing set of the original queue being S is larger than the new overhearing set $\tilde{S}^*$ and thus we can reuse the same packet $W_{(k,\mathrm{i})}$ instead of the new packet $Y(t)$. Since the original coding vector $\mathsf{row}(k,\mathrm{i})(\mathbf{U}^{(k)})$ generally has a shorter expression than the new coding vector $\mathbf{y}(t)$, this modification reduces the growth rate of the header length.

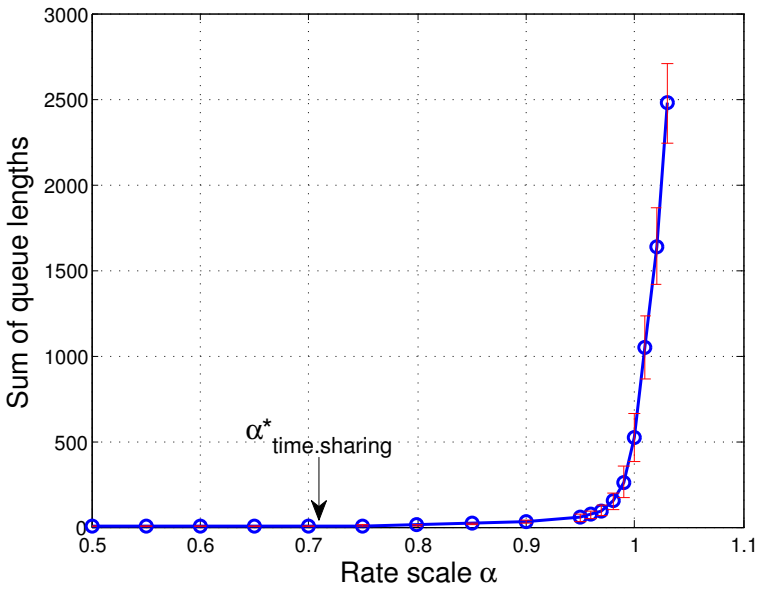


Figure 2.4. The sum of queue lengths $\sum_{k,S} |Q_S^{(k)}|$ for different α.

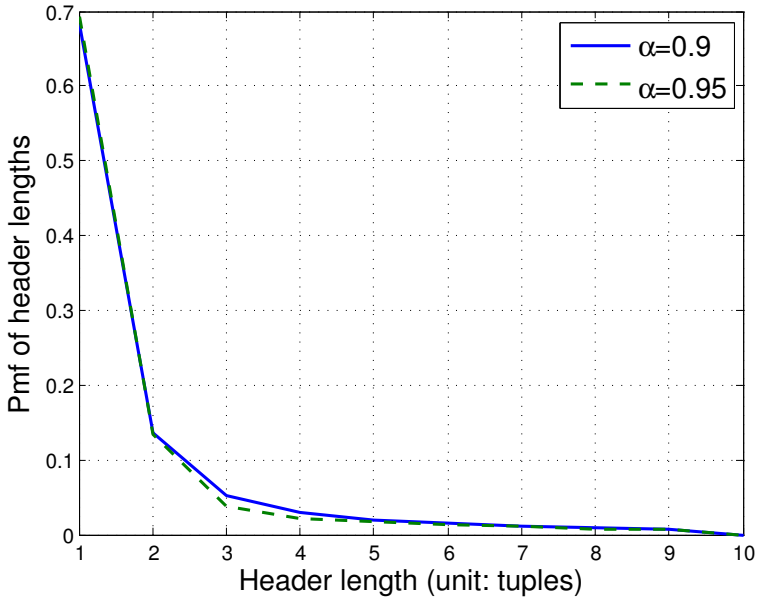


Figure 2.5. Pmf of the header length: When $\alpha = 0.9$, the largest header length in the simulation of 10^5 time slots is 116 and the average is 2.5. When $\alpha = 0.95$, the largest and the average header lengths become 135 and 2.7.

Using this modified scheme, we run 5 trials with $\alpha = 0.95$ and the empirical pmf of header length (number of triples), computed over 10^5 time slots and 5 trials, is shown in Fig. 2.5. The average header length per transmission is about 2.7, which reflects the average number of combined packets in a coded packet. In fact the maximum header length in 10^5 time slots is about 135 (unit: tuples) and 95% packets has header length no larger than 10 (unit: tuples). That is, if each $(k, \mathrm{i}, \beta_{k,\mathrm{i}})$ tuple in the header takes 4 bytes, then 95% packets use no more than 40 bytes for headers, or 4% of a regular 1000 bytes packet, and the average size of header is $2.7 \cdot 4 = 10.8$ bytes, which is 1.1% of the payload length.

Finally, we compare the per-packet total delay, which is measured by the time span between a packet first arriving at source s until it being fully decoded at destination d_k (queueing plus decoding delay). We run 5 trials with $\alpha = 0.95$ and Fig. 2.6 shows the empirical pmf of the total delay over 10^5 time slots and 5 trials. The average per-packet total delay is 513.6 time slots. Specifically the average per-packet total delay are 882.8, 774.3, 406.6, and 370.2 for each destination d_1, d_2, d_3, and d_4, respectively. Due to the

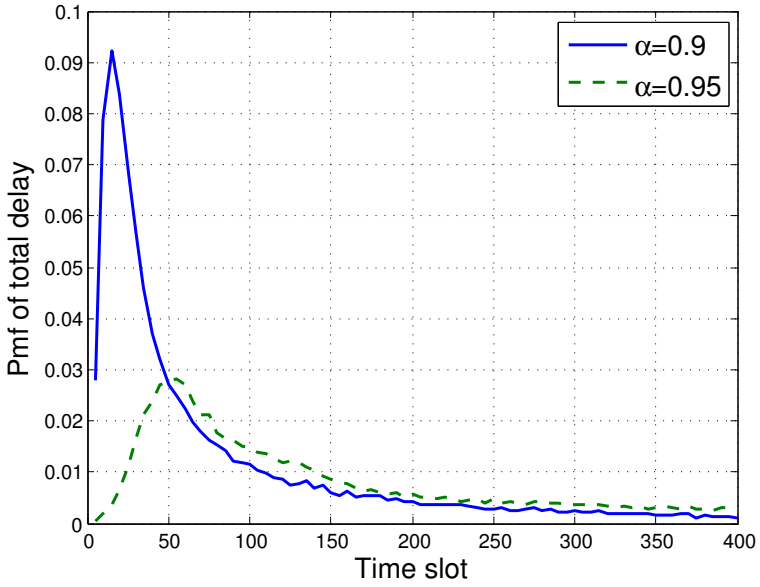


Figure 2.6. Pmf of the total delay: When $\alpha = 0.9$, the largest total delay in the simulation of 10^5 time slots is 3332 and the average is 108.7. 94% of the packets have total delay ≤ 400 time slots. When $\alpha = 0.95$, the largest and the average delay become 13382 and 513.6. 67% of the packets have total delay ≤ 400 time slots.

spatial independent PEC in the simulation, the delay is roughly inversely proportional to marginal success probability (p_1, p_2, p_3, p_4).

Figs. 2.5 and 2.6 also plot the empirical pmfs of the header length and the total delay when operated at $\alpha = 0.9$. Comparing the case of $\alpha = 0.9$ and the previous, more demanding case of $\alpha = 0.95$, one can see that the average header length decreases slightly from 2.7 to 2.5; but the average total decoding delay decreases 5 fold from 513.6 to 108.7. The substantial delay reduction is as expected since the closer we are operating to the capacity, the higher number of packets will be queued (longer queueing delay) and their mixture will lead to much longer decoding delay at the receiver.

2.7 Summary

We investigate the stability region of 1-to-K broadcast packet erasure channels with ACK/NACK. The schemes in prior works has some drawbacks that is difficult to implement in practice. We have presented a new network coding protocol for 1-to-K broadcast packet

erasure channels with causal ACK/NACK to address the drawbacks. The achievable rate region matches the capacity region for all the scenarios in which the capacity is known. The proposed scheme has many desirable features, including sequential encoding and being adaptive to unknown arrival rates.

3. CODED CACHING SYSTEM OF TWO USERS AND TWO FILES

In this chapter, we focus on the coded caching system of one server and K users, each user having its cache memory and demanding a file among the N files at server. The system consists of two phases: The placement phase: Each user accesses the N files and fills its cache memory; and the delivery phase: Each user submits its own file request and the server broadcasts a signal to deliver the desired packets to each user. In this part, we characterizes the *exact capacity* of the smallest 2-user/2-file ($N = K = 2$) coded caching problem but under the most general setting that simultaneously allows for (i) heterogeneous files sizes, (ii) heterogeneous cache sizes, (iii) user-dependent file popularity, and (iv) average-rate analysis. Solving completely the case of $N = K = 2$ could shed further insights on the performance and complexity of optimal coded caching with full heterogeneity for arbitrary N and K.

3.1 Introduction

Nowadays high-definition video streaming motivates the demand of high-throughput Internet traffic with small delay. One way to contain the peak load within the underlying communication channel capacity is to use caching to re-distribute some of the peak traffic to off-peak hours by prefetching (some of) the content in advance. The design tasks of a caching scheme consist of two parts: how to place the content during off-peak hours and how to satisfy the requests by delivering the additional packets during peak hours. Caching is especially attractive under the model of *broadcast channels* for which a single packet transmission could simultaneously benefit/reach multiple destinations.

Content caching has been studied in various settings [30], such as exploiting the opportunities of user population, file correlation, and time correlation. These traditional techniques usually divide each file into multiple (uncoded) pieces, prefetch some of them, and transmit the rest when needed. Recently, coded caching was proposed [31], which reduces delivery time by substituting the uncoded pieces with a coded version and taking advantage of multicasting capabilities. The framework of coded caching consider one server of N files and K

users over an error-free broadcast link. The system is performed in two stages: In the *placement phase*, each user is able to encode the information of all the N files as the content in its cache memory. In the *delivery phase*, each user demands one file and the server broadcasts a signal to all the users such that all the users are able to decode their requests. Consider an example of 3-user/3-file ($N = K = 3$) coded caching system. The server has 3 files A, B, and C, each of file size F; and each user has cache memory of size $M = F$. A coded caching scheme as shown in Fig. 3.1 is described in the following. We divide each file into three subfiles of equal size, i.e., $A = (A_1, A_2, A_3)$, $B = (B_1, B_2, B_3)$, and $C = (C_1, C_2, C_3)$. In the placement phase, each user k stores (A_k, B_k, C_k) to fill its memory. In the delivery phase, suppose user 1 requests file B, user 2 requests file C, and user 3 requests file A, then the server can transmit the signal $(A_1 \oplus B_3, B_2 \oplus C_1, A_2 \oplus C_3)$ of rate $R = F$ such that all the three users are able to decode their requested files. That is, user 1 uses $(A_1 \oplus B_3, B_2 \oplus C_1)$ to decode (B_2, B_3), user 2 uses $(B_2 \oplus C_1, A_2 \oplus C_3)$ to decode (C_1, C_3), and user 3 use $(A_1 \oplus B_3, A_2 \oplus C_3)$ to decode (A_1, A_2). Comparing to the uncoded signal in delivery phase that requires rate $R = 2F$, the coded caching scheme under such file request reduces half of the throughput.

By exploiting the coded caching with broadcast channel, [31] show that show that coded caching can shorten the *worst-case* delivery time by a factor of $(\frac{1}{1+KM/FN})$ when compared to the traditional uncoded caching schemes, where N is the number of files, K is the number of users, M is the individual cache size and F is the individual file size. While the capacity of the general coded caching problem remains an open problem, the optimal coded caching scheme (exact capacity) has been characterized for some special cases [31]–[33] and order-optimal capacity characterization has been found for several more general scenarios [31], [34]–[39].

Most existing results are based on the settings of (i) homogeneous file sizes, (ii) homogeneous cache sizes, (iii) user-independent homogeneous file popularity, and (iv) worst-case analysis. These settings are not 100% compatible with the traditional uncoded caching solutions. Specifically, the basic design principle of traditional uncoded schemes is to first predict the likelihood of the next file request for each individual user separately (i.e., user-dependent heterogeneous file popularity), and then let each user store the most likely file(s) until his/her cache is full (which is naturally applicable to heterogeneous file and cache sizes). The ratio-

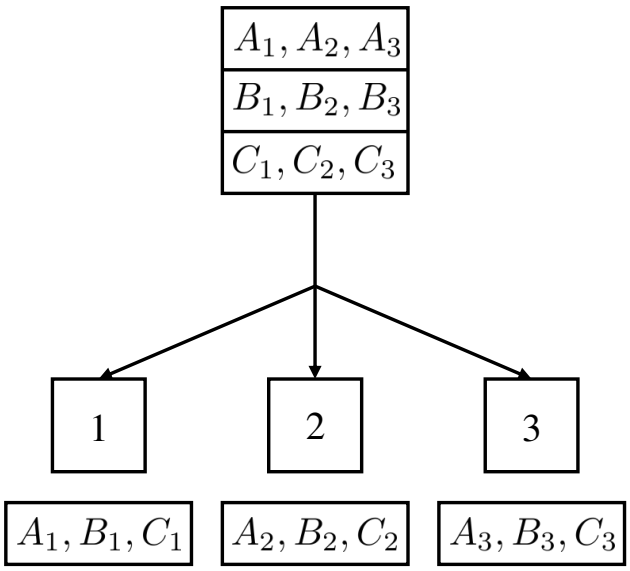


Figure 3.1. The 3-user/3-file ($N = K = 3$) coded caching example. The sever has three files (A, B, C) of same size F and each file is split into three equal subfiles: $A = (A_1, A_2, A_3)$, $B = (B_1, B_2, B_3)$, and $C = (C_1, C_2, C_3)$. Each user k has same cache memory size F and store cache content (A_k, B_k, C_k).

nale behind this simple design is that such a probability-based greedy solution would reduce the *average rate* during delivery, even though there is no information-theoretic optimality guarantee.

Because of the aforementioned differences between their settings, a coded scheme designed for the homogeneous, worst-case setting could have significantly worse average-rate performance in practice when compared to a traditional scheme, especially for the scenarios in which the individualized file request prediction is very effective and the file and cache sizes are highly heterogeneous. In principle, since coded caching is a strict generalization of any uncoded solution, an optimal coded caching solution should outperform its non-coded counterpart under *any setting*. This potential loss of performance[1] is mainly due to the mismatch between practical scenarios and the homogeneous and worst case settings for which existing coded caching schemes are optimized.

Motivated by this observation, this work studies the exact capacity region and the corresponding optimal coded caching schemes under (i) heterogeneous file sizes, (ii) heterogeneous cache sizes, (iii) user-dependent heterogeneous file popularity, and (iv) average-rate analysis. Such results could allow the system designers to accurately assess the performance gain of coded caching (the ultimate capacity minus the achievable rate of traditional uncoded schemes) in a practical heterogeneous setting. While the problem remains open for general N and K values, we characterize the exact capacity for $N = K = 2$. The results could shed further insights for general N and K.

3.1.1 Comparison to Existing Results

Several existing works relaxed parts of the above conditions (i) to (iv). Table 3.1 provides a non-comprehensive list of several related results. For example, the authors in [31] assume homogeneous file and cache size and, under those conditions, characterize the exact worst-case capacity when $N = K = 2$ and propose a scheme that achieves order-optimal worst-case rate for arbitrary N and K. [37] provides a new information-theoretic lower bound and a

[1]↑In practice, there are other issues that need to be addressed, e.g., synchronization [40]. Our statement disregards the implementation overhead and focuses exclusively on the theoretic performance under heterogeneous settings.

Table 3.1. Comparisons of existing results

	Worst-case rate	Average rate
Homo. file homo. cache	• Arbitrary K and N, order-optimal rate [31], [34], [39] • $K = 2$ and arbitrary N, exact capacity [31], [32] • $K = 3$ and $N = 2$, exact capacity [32]	• Arbitrary K and N, order-optimal rate [36]–[39], [43], [44] • Arbitrary K and N, achievable rate only [45], [46] • Arbitrary K and $N = 2$, uncoded placement, exact cap. [47]
Homo. file heter. cache	• $K = 2$ and arbitrary N, exact capacity [33] • Arbitrary K and N, achievable rate only [48]	• Arbitrary K and N, achievable rate only [42]
Heter. file homo. cache	• Arbitrary K and N, order-optimal rate [49]–[51]	• Arbitrary K and N, achievable rate only [42]
Heter. file heter. cache	• $N = K = 2$, exact capacity [41]	• Arbitrary K and N, achievable rate only [42] • $N = K = 2$, exact capacity [52]

corresponding order-optimal scheme of average rate with homogeneous file size, homogeneous cache size, and user-independent popularity.

As can be seen in Table 3.1, finding the exact capacity of coded caching remains a difficult task. Most existing exact capacity results [31]–[33], [41] are based on small K (i.e., $K = 2$ or $K = 3$) and focus on the *worst-case* rate rather than a general probabilistic average-rate model. One of the most general heterogeneous setting results is [42], which uses linear programming results to search for better achievable rates without deriving any converse bounds, and is not focused on the general user-dependent file popularity setting. By focusing exclusively on the average-rate setting with heterogeneous file and cache sizes as well as user-dependent file popularity, our $N = K = 2$ results represent the first step toward fully characterizing the capacity of coded caching with full heterogeneity.

3.2 General Coded Caching Model

We consider the simplest non-trivial coded caching system with $N = 2$ files and $K = 2$ users. A central server has access to two files W_1 and W_2 of file sizes F_1 and F_2 bits,

respectively. We sometimes write F_1 and F_2 as some non-integer values, e.g., $F_1 = 1.5$ and $F_2 = \frac{1}{3}$. One way to interpret this real-valued file-size expression is to assume F_1 and F_2 are sufficiently large so that we can express F_1 and F_2 by their normalized values instead. The cache content of user k is denoted by Z_k and is of size M_k bits for $k \in \{1, 2\}$. Without loss of generality, we assume real-valued $M_k \in [0, F_1 + F_2]$ for all k.

The operation of the system contains two phases, the *placement phase* and the *delivery phase*. In the *placement phase*, user k populates its cache by

$$Z_k = \phi_k(W_1, W_2), \ \forall k \in \{1, 2\}, \tag{3.1}$$

where ϕ_k is the caching function of user k. In the *delivery phase*, the two users send a demand request $\vec{d} \triangleq (d_1, d_2) \in \{1, 2\}^2$ to the server, i.e., user k demands file W_{d_k}. The probability mass function of the demand request $\vec{d}$ is denoted by $p_{\vec{d}}$, which satisfies $\sum_{\vec{d} \in \{1,2\}^2} p_{\vec{d}} = 1$. We assume $\{p_{\vec{d}} : \vec{d} \in \{1, 2\}^2\}$ is known to the server.

One popular choice of $p_{\vec{d}}$ is to assume that the demands of user-1 and user-2 are *statistically independent*, i.e., $p_{(d_1,d_2)} = p_{d_1}^{[1]} p_{d_2}^{[2]}$ where $p_d^{[k]}$ is the marginal probability that user-k requests file W_d. In this work, we allow for arbitrary $p_{\vec{d}}$ that can be statistically independent or not.

After receiving $\vec{d}$, the server *broadcasts* an encoded signal

$$X_{\vec{d}} = \psi(\vec{d}, W_1, W_2) \tag{3.2}$$

of $R_{\vec{d}}$ bits with encoding function ψ through an *error-free* broadcast channel. Each user k then uses its cache content Z_k and the received signal $X_{\vec{d}}$ to decode his/her desired file

$$\hat{W}_{d_k} = \mu_k(\vec{d}, X_{\vec{d}}, Z_k), \ \forall k \in \{1, 2\}, \tag{3.3}$$

where μ_k is the decoding function of user k. Herein we assume that user k knows the network-wide request pattern $\vec{d}$, which can be easily achieved by piggybacking the 2-bit vector $\vec{d}$ to the encoded symbol $X_{\vec{d}}$.

Definition 3.2.1. *A coded caching scheme for $N = K = 2$ is specified completely by its five functions $\{\phi_1, \phi_2, \psi, \mu_1, \mu_2\}$. The scheme is* zero-error feasible *if $\hat{W}_{d_k} = W_{d_k}$ for all $\vec{d} \in \{1,2\}^2$, all $k \in \{1,2\}$, and all $W_k \in \{0,1\}^{F_k}$.*

Definition 3.2.2. *The worst-case rate of a zero-error coded caching scheme is*

$$R^* = \max_{\vec{d} \in \{1,2\}^2} R_{\vec{d}}. \tag{3.4}$$

The worst-case capacity is the infimum of the worst-case rates of all zero-error schemes.

Definition 3.2.3. *The average rate of a zero-error coded caching scheme is*

$$\bar{R} = \sum_{\vec{d} \in \{1,2\}^2} p_{\vec{d}} R_{\vec{d}}. \tag{3.5}$$

The average-rate capacity is the infimum of the average rates of all zero-error schemes.

For simplicity, we slightly abuse the above notation and directly use R^* and $\bar{R}$ to denote the worst-case and the average-rate capacities, respectively, even though their original definitions in (3.4) and (3.5) are for the achievable rates instead.

3.3 The Two-User/Two-File Coded Caching Capacity

To solve the worst-case and the average-rate capacities R^* and $\bar{R}$, we first define the following strictly more general concept.

Definition 3.3.1. *The* per-request capacity region *(PRCR) is the closure of the rate vectors $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$ of all zero-error coded caching schemes.*

The PRCR is the most fundamental performance limit of coded caching since it captures the optimal trade-off needed to simultaneously satisfy different request patterns.

In Section 3.3.1 we describe 7 basic coded caching schemes for the 2-file/2-user setting ($N = K = 2$). Section 3.3.2 then provides the basic lower bounds of 4-dimensional coded caching rate $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$. Finally, Section 3.3.3 shows that the 7 basic schemes can achieve the 4-dimensional rate lower bounds. The end result is thus a complete characterization of the PRCR for arbitrary (F_1, F_2, M_1, M_2) values. The exact characterization of

Table 3.2. Basic coded caching schemes for two files of size (f_1, f_2) and two users of memory (m_1, m_2). It is possible to have $f_1 \geq f_2$, or $f_1 < f_2$, or $m_1 \geq m_2$, or $m_1 < m_2$. The intuition of the schemes are Mix.Emp for premixing at d_1, Emp.Mix for premixing at d_2, Ha.Fi for splitting files in halves, and a.b.Cov for covering $\vec{d} = (a, b)$.

Scheme	Feasibility Condition	Achievable Rate Vector $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$
Mix.Emp	$f{=}f_1{=}f_2{=}m_1$; $m_2{=}0$	(f, f, f, f)
Emp.Mix	$f{=}f_1{=}f_2{=}m_2$; $m_1{=}0$	(f, f, f, f)
Ha.Fi	$f{=}f_1{=}f_2{=}m_1{=}m_2$	$(f/2, f/2, f/2, f/2)$
1.1.Cov	$\max(m_1, m_2) \leq f_1$	$(f_1{-}\min(m_1, m_2), f_1{+}f_2{-}m_1, f_1{+}f_2{-}m_2, f_2)$
1.2.Cov	$m_1 \leq f_1$, $m_2 \leq f_2$	$(f_1, f_1{+}f_2{-}m_1{-}m_2, f_1{+}f_2{-}\min(m_1, m_2), f_2)$
2.1.Cov	$m_1 \leq f_2$, $m_2 \leq f_1$	$(f_1, f_1{+}f_2{-}\min(m_1, m_2), f_1{+}f_2{-}m_1{-}m_2, f_2)$
2.2.Cov	$\max(m_1, m_2) \leq f_2$	$(f_1, f_1{+}f_2{-}m_2, f_1{+}f_2{-}m_1, f_2{-}\min(m_1, m_2))$

PRCR will naturally lead to new closed form expressions of the capacities R^* and $\bar{R}$ under any arbitrary file popularity distribution $p_{\vec{d}}$. Further discussion on how to use the new PRCR characterization to derive the average-rate capacity $\bar{R}$ is provided at the end of Section 3.3.3.

3.3.1 Basic Zero-Error Coded Caching Schemes

We first describe 7 coded caching schemes for the 2-file/2-user setting ($N = K = 2$), which later serve as the basis for all our achievability proofs when characterizing the 4-dimensional PRCR. Consider user 1 and 2 of cache memory size m_1 and m_2 with two files w_1 and w_2 of sizes f_1 and f_2, respectively, the 7 basic schemes of parameters (f_1, f_2, m_1, m_2) are listed in Table 3.2 and described as follows.

1) Mix.Emp: Consider two files of equal size $f_1 = f_2 = f$, and two users of memory sizes $m_1 = f$ and $m_2 = 0$. In the placement phase, user 1 caches $w_1 \oplus w_2$ and user 2 caches none. In the delivery phase, the transmitted signals for the demands are $X_{(1,1)} = w_1$, $X_{(1,2)} = w_2$, $X_{(2,1)} = w_1$, and $X_{(2,2)} = w_2$. One can easily verify that for any $\vec{d}$, the transmitted symbol $X_{\vec{d}}$ satisfies the demands of both users. Since $X_{\vec{d}}$ is of size f for all $\vec{d}$, the corresponding achievable rate vector is $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f, f, f, f)$. The first row of Table 3.2 summarizes the achievable rate vector and the condition $f = f_1 = f_2 = m_1$, $m_2 = 0$ for

this scheme to be zero-error feasible. Since user 1 stores an XORed packet and user 2 stores none, we call this scheme Mix.Emp.

2) Emp.Mix: The scheme is user-symmetric to Mix.Emp by swapping the roles of users 1 and 2. Since this time user 1 stores none and user 2 stores an XORed packet, we call this scheme Emp.Mix.

3) Ha.Fi: Consider two files of equal size $f_1 = f_2 = f$, and two users of equal memory size $m_1 = m_2 = f$. We divide the file $w_1 = (u_1, u_2)$ into two subfiles of size $(f/2, f/2)$ and divide the file $w_2 = (v_1, v_2)$ into two subfiles of size $(f/2, f/2)$. In the placement phase, user 1 caches (u_1, v_1) and user 2 caches (u_2, v_2). In the delivery phase, the transmitted signals for the demands are $X_{(1,1)} = u_1 \oplus u_2$, $X_{(1,2)} = u_2 \oplus v_1$, $X_{(2,1)} = u_1 \oplus v_2$, and $X_{(2,2)} = v_1 \oplus v_2$. Since $X_{\vec{d}}$ is of size $f/2$ for all $\vec{d}$, the achievable rates are $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f/2, f/2, f/2, f/2)$. Since each user stores half of file k for all k, we call this scheme Ha.Fi.

4) 1.1.Cov: In this scheme both users cache as much as possible from file 1. Consider two users of cache memory size $\max(m_1, m_2) \leq f_1$ and arbitrary f_2. If $m_1 \geq m_2$, we divide $w_1 = (u_1, u_2, u_3)$ into three subfiles of file size $(m_2, m_1 - m_2, f_1 - m_1)$. In the placement phase, user 1 caches (u_1, u_2) and user 2 caches u_1. In the delivery phase, the transmitted signals for the different demands $\vec{d}$ are $X_{(1,1)} = (u_2, u_3)$, $X_{(1,2)} = (u_3, w_2)$, $X_{(2,1)} = (u_2, u_3, w_2)$, and $X_{(2,2)} = w_2$. One can easily verify that both users can decode their desired files under any demand $\vec{d}$. By quantifying the size of $X_{\vec{d}}$ for all $\vec{d}$, the achievable rates are $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f_1 - m_2, f_1 + f_2 - m_1, f_1 + f_2 - m_2, f_2)$.

If $m_1 < m_2$, we can implement the same scheme by swapping the roles of users 1 and 2. By taking into account both scenarios ($m_1 \geq m_2$ and $m_1 < m_2$), we can write the achievable rate vector in the following more general form:

$$\begin{aligned}(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (&f_1 - \min(m_1, m_2),\\ &f_1 + f_2 - m_1, f_1 + f_2 - m_2, f_2).\end{aligned} \tag{3.6}$$

Since the strategy of both users is "to cover as much file 1 as possible", we call this scheme 1.1.Cov.

5) 1.2.Cov: In this scheme user 1 caches file 1 and user 2 caches file 2. Consider two users of memory size $m_1 \leq f_1$ and $m_2 \leq f_2$. If $m_1 \geq m_2$, we divide $w_1 = (u_1, u_2, u_3)$ into three subfiles of size $(m_2, m_1 - m_2, f_1 - m_1)$ and divide $w_2 = (v_1, v_2)$ into two subfiles of file size $(m_2, f_2 - m_2)$. In the placement phase, user 1 caches (u_1, u_2) and user 2 caches v_1. In the delivery phase, the transmitted signals for the different demands are $X_{(1,1)} = w_1$, $X_{(1,2)} = (u_2, v_2)$, $X_{(2,1)} = (u_1 \oplus v_1, u_2, u_3, v_2)$, and $X_{(2,2)} = w_2$, which results in the achievable rate vector being $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f_1, f_1 + f_2 - m_1 - m_2, f_1 + f_2 - m_2, f_2)$.

If $m_1 < m_2$, a symmetric scheme can be implemented by dividing w_1 into two subfiles of size $(m_1, f_1 - m_1)$ and w_2 into three subfiles of size $(m_1, m_2 - m_1, f_2 - m_2)$. By taking into account both scenarios ($m_1 \geq m_2$ and $m_1 < m_2$), we can write the achievable rate vector in the following more general form:

$$\begin{aligned}(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (&f_1, \\ &f_1 + f_2 - m_1 - m_2, f_1 + f_2 - \min(m_1, m_2), f_2).\end{aligned} \tag{3.7}$$

Since the strategy of user 1 is "to cover as much file 1 as possible" and the strategy of user 2 is "to cover as much file 2 as possible", we call this scheme 1.2.Cov.

6) 2.1.Cov: The scheme is user-symmetric to 1.2.Cov by swapping the roles of users 1 and 2. Since the strategy of user 1 is "to cover as much file 2 as possible" and the strategy of user 2 is "to cover as much file 1 as possible", we call this scheme 2.1.Cov.

7) 2.2.Cov: The scheme is file-symmetric to 1.1.Cov by swapping the roles of files 1 and 2. Since the strategy of user 1 is "to cover as much file 2 as possible" and so is user 2's strategy, we call this scheme 2.2.Cov.

It is worth noting that none of the 7 basic schemes can be achieved by space-sharing the rest of 6 schemes and they thus will serve as the basis of our achievability proofs [53].

3.3.2 Lower Bounds of the PRCR

We derive the following lower bounds for arbitrary file and cache sizes (F_1, F_2, M_1, M_2).

Instance 0: Nonnegative rates:

$$R_{\vec{d}} \geq 0, \quad \forall \vec{d} \in \{1,2\}^2.$$

By varying $\vec{d}$, there are totally 4 inequalities in Instance 0.

$$R_{(1,1)} \geq 0 \qquad \text{(O-1)} \qquad\qquad R_{(1,2)} \geq 0 \qquad \text{(O-2)}$$

$$R_{(2,1)} \geq 0 \qquad \text{(O-3)} \qquad\qquad R_{(2,2)} \geq 0 \qquad \text{(O-4)}$$

Instance 1: For any $\mathrm{i}, \mathrm{j} \in \{1,2\}$, there are two inequalities:

$$\begin{aligned} R_{(\mathrm{i},\mathrm{j})} + M_1 \geq & H(X_{(\mathrm{i},\mathrm{j})}, Z_1) = H(X_{(\mathrm{i},\mathrm{j})}, Z_1, W_{\mathrm{i}}) \geq F_{\mathrm{i}}, \\ R_{(\mathrm{i},\mathrm{j})} + M_2 \geq & H(X_{(\mathrm{i},\mathrm{j})}, Z_2) = H(X_{(\mathrm{i},\mathrm{j})}, Z_2, W_{\mathrm{j}}) \geq F_{\mathrm{j}}. \end{aligned}$$

By varying i, j, there are totally 8 inequalities in Instance 1.

$$R_{(1,1)} + M_1 \geq F_1 \qquad \text{(I-1)} \qquad\qquad R_{(1,1)} + M_2 \geq F_1 \qquad \text{(I-2)}$$

$$R_{(1,2)} + M_1 \geq F_1 \qquad \text{(I-3)} \qquad\qquad R_{(1,2)} + M_2 \geq F_2 \qquad \text{(I-4)}$$

$$R_{(2,1)} + M_1 \geq F_2 \qquad \text{(I-5)} \qquad\qquad R_{(2,1)} + M_2 \geq F_1 \qquad \text{(I-6)}$$

$$R_{(2,2)} + M_1 \geq F_2 \qquad \text{(I-7)} \qquad\qquad R_{(2,2)} + M_2 \geq F_2 \qquad \text{(I-8)}$$

Instance 2: For any $(\mathrm{i}, \mathrm{j}) = (1,2)$ or $(2,1)$,

$$R_{(\mathrm{i},\mathrm{j})} + M_1 + M_2 \geq H(X_{(\mathrm{i},\mathrm{j})}, Z_1, Z_2) \geq F_1 + F_2.$$

By varying (i, j), there are totally 2 inequalities in Instance 2.

$$R_{(1,2)} + M_1 + M_2 \geq F_1 + F_2 \qquad \text{(II-1)}$$

$$R_{(2,1)} + M_1 + M_2 \geq F_1 + F_2. \qquad \text{(II-2)}$$

Instance 3: For any $\mathrm{i},\mathrm{j} \in \{1,2\}$, there are two inequalities:

$$R_{(\mathrm{i},1)}+R_{(\mathrm{j},2)} + M_2 \geq H(X_{(\mathrm{i},1)}, X_{(\mathrm{j},2)}, Z_2) \geq F_1 + F_2,$$
$$R_{(1,\mathrm{i})}+R_{(2,\mathrm{j})} + M_1 \geq H(X_{(1,\mathrm{i})}, X_{(2,\mathrm{j})}, Z_1) \geq F_1 + F_2.$$

By varying i,j, there are totally 8 inequalities in Instance 3.

$$R_{(1,1)} + R_{(1,2)} + M_2 \geq F_1 + F_2, \tag{III-1}$$
$$R_{(1,1)} + R_{(2,1)} + M_1 \geq F_1 + F_2, \tag{III-2}$$
$$R_{(1,1)} + R_{(2,2)} + M_1 \geq F_1 + F_2, \tag{III-3}$$
$$R_{(1,1)} + R_{(2,2)} + M_2 \geq F_1 + F_2, \tag{III-4}$$
$$R_{(1,2)} + R_{(2,1)} + M_1 \geq F_1 + F_2, \tag{III-5}$$
$$R_{(1,2)} + R_{(2,1)} + M_2 \geq F_1 + F_2, \tag{III-6}$$
$$R_{(1,2)} + R_{(2,2)} + M_1 \geq F_1 + F_2, \tag{III-7}$$
$$R_{(2,1)} + R_{(2,2)} + M_2 \geq F_1 + F_2. \tag{III-8}$$

Instance 4 uses a more refined technique[2] and we thus provide the detailed derivation.

Instance 4: For any $(\mathrm{i},\mathrm{j}) = (1,2), (2,1)$, or $(2,2)$,

$$R_{(\mathrm{i},1)} + R_{(1,\mathrm{j})} + M_1 + M_2 \tag{3.8}$$
$$\geq H(X_{(\mathrm{i},1)}) + H(Z_2) + H(X_{(1,\mathrm{j})}) + H(Z_1) \tag{3.9}$$
$$\geq H(X_{(\mathrm{i},1)}, Z_2) + H(X_{(1,\mathrm{j})}, Z_1) \tag{3.10}$$
$$\geq H(X_{(\mathrm{i},1)}, Z_2, W_1) + H(X_{(1,\mathrm{j})}, Z_1, W_1) \tag{3.11}$$
$$\geq H(X_{(\mathrm{i},1)}, X_{(1,\mathrm{j})}, Z_1, Z_2, W_1) + H(W_1) \tag{3.12}$$
$$\geq H(X_{(\mathrm{i},1)}, X_{(1,\mathrm{j})}, Z_1, Z_2, W_1, W_2) + H(W_1) \tag{3.13}$$
$$= H(W_1, W_2) + H(W_1) = 2F_1 + F_2 \tag{3.14}$$

[2]↑A more general version of the techniques can be found in [31], [32], [54], [55].

where (3.10) follows from that the sum of marginal entropies is no less than the joint entropy; (3.11) follows from that user 2 can decode W_1 based on $X_{(\mathrm{i},1)}$ and Z_2, and user 1 can decode W_1 based on $X_{(1,\mathrm{j})}$ and Z_1; (3.12) follows from the Shannon-type inequality; (3.13) follows from that we can decode W_2 from $X_{(\mathrm{i},1)}$, $X_{(1,\mathrm{j})}$, Z_1, and Z_2 since we choose $(\mathrm{i},\mathrm{j}) \in \{(1,2),(2,1),(2,2)\}$ to begin with; and (3.14) follows from that X's and Z's are functions of (W_1, W_2).

Symmetrically for any $(\mathrm{i},\mathrm{j}) = (1,2),(2,1)$, or $(1,1)$

$$R_{(\mathrm{i},2)} + R_{(2,\mathrm{j})} + M_1 + M_2 \geq F_1 + 2F_2.$$

Varying (i,j), there are totally 6 inequalities in Instance 4.

$$R_{(1,1)} + R_{(1,2)} + M_1 + M_2 \geq 2F_1 + F_2, \tag{IV-1}$$
$$R_{(1,1)} + R_{(2,1)} + M_1 + M_2 \geq 2F_1 + F_2, \tag{IV-2}$$
$$R_{(1,2)} + R_{(2,1)} + M_1 + M_2 \geq 2F_1 + F_2, \tag{IV-3}$$
$$R_{(1,2)} + R_{(2,1)} + M_1 + M_2 \geq F_1 + 2F_2, \tag{IV-4}$$
$$R_{(1,2)} + R_{(2,2)} + M_1 + M_2 \geq F_1 + 2F_2, \tag{IV-5}$$
$$R_{(2,1)} + R_{(2,2)} + M_1 + M_2 \geq F_1 + 2F_2. \tag{IV-6}$$

Totally, there are 28 linear inequalities in Instances 0 to 4.

3.3.3 Coded Caching Capacity for $N = K = 2$

The derivation of the aforementioned lower bounds is relatively straightforward, see [31], [32], [41], [54] for similar derivations. A more important contribution of this work is to show that these lower bounds indeed characterize the exact 4-dimensional PRCR.

Proposition 3.3.1. *Consider arbitrary (F_1, F_2, M_1, M_2). For any $\vec{R}$ that satisfies the 28 lower bounds in Section 3.3.2 simultaneously, we can find a zero-error scheme attaining such $\vec{R}$.*

Proposition 3.3.1 leads to the following self-explanatory corollary:

Corollary 2. *Given arbitrary (F_1, F_2, M_1, M_2) values, the average-rate capacity can be characterized by solving a linear programming (LP) problem using the 28 lower bounds in Section 3.3.2.*

Further discussion of Corollary 2 will be provided in the remark after Proposition 3.3.3.

Proposition 3.3.1 follows directly from the following propositions.

Proposition 3.3.2. *The 4-dimensional polytope formed by the 28 linear inequalities has either 2 or 4 or 6 distinct corner points. The actual number depends on the underlying (F_1, F_2, M_1, M_2) values. An exhaustive list of all the corner points under arbitrary (F_1, F_2, M_1, M_2) is provided jointly in Fig. 3.2 and Table 3.3.*

Proposition 3.3.3. *All 28 corner points listed in Fig. 3.2 and Table 3.3 can be achieved by space-sharing the 7 basic schemes described in Section 3.3.1.*

The proofs of Propositions 3.3.2 and 3.3.3 are relegated to Appendices B.1 and B.2, respectively. In the proof of Proposition 3.3.3, we explicitly find 28 constructions that achieve the 28 corner points, respectively.

Remark: The statement in Proposition 3.3.1 has already cast the coded caching capacity problem as an LP problem involving 4 variables $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$ and 28 inequalities, which can be solved numerically. Nonetheless, the constructive and explicit statements in Propositions 3.3.2 and 3.3.3 go one step further. By exhaustively characterizing all corner points of the lower bounds and then proving their achievability, one can use Proposition 3.3.3 to devise the coded caching scheme of any feasible $\vec{R}$ and the formulas in Proposition 3.3.2, i.e., the expressions listed in Table 3.3, can be used to derive the closed-form expression of the capacity without any numerical solver. Compared to the implicit statement in Proposition 3.3.1, Propositions 3.3.2 and 3.3.3 uncover new, cleaner results that shed further insight to the problem at hand.

For example, since the average capacity $\bar{R}$ is achieved by the vector $\vec{R}$ in the PRCR that has the smallest linear objective value $\sum p_{\vec{d}} R_{\vec{d}}$ and since the minimum of a linear programming problem can only happen at the corner points, we can easily use the corner points in Fig. 3.2 and Table 3.3 to characterize the average-rate capacity of arbitrary popularity

Table 3.3. The expressions of all 28 possible corner points.

Vertex	Corresponding rate vector $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$
1	$(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$
2	$(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$
3	$(F_1, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2 - M_2)$
4	$(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$
5	$(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$
6	$(F_1, F_1, F_1 + F_2 - M_1 - M_2, F_2)$
7	$(F_1 + \frac{F_2 - M_1 - M_2}{2}, F_1 + \frac{F_2 - M_1 - M_2}{2}, F_1 + \frac{F_2 - M_1 - M_2}{2}, F_2 + \frac{F_2 - M_1 - M_2}{2})$
8	$(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$
9	$(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$
10	$(F_1, F_1 + F_2 - M_1, F_1, F_2 - M_2)$
11	$(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$
12	$(F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_1 - M_2, F_2)$
13	$(F_1 + \frac{F_2 - M_1 - M_2}{2}, F_1 + \frac{F_2 - M_1 - M_2}{2}, F_1 + \frac{F_2 - M_1 - M_2}{2}, \frac{F_2 + M_1 - M_2}{2})$
14	$(F_1 - M_2, F_2, F_1 - M_2, F_2)$
15	$(F_1, F_2 - M_2, F_1, F_2 - M_2)$
16	$(F_1, F_2 - M_2, F_1, F_1 + F_2 - M_1)$
17	$(\frac{F_1 + M_1 - M_2}{2}, F_2 + \frac{F_1 - M_1 - M_2}{2}, \frac{F_1 + M_1 - M_2}{2}, F_2 + \frac{F_1 - M_1 - M_2}{2})$
18	$(F_1 - M_2, F_2, F_1 + F_2 - M_1, F_2)$
19	$(F_1 + F_2 - M_1, F_2, F_1 - M_2, F_2)$
20	$(F_1 + \frac{F_2 - M_1 - M_2}{2}, \frac{F_2 + M_1 - M_2}{2}, F_1 + \frac{F_2 - M_1 - M_2}{2}, \frac{F_2 + M_1 - M_2}{2})$
21	$(F_1 + F_2 - M_2, F_1 - M_1, F_1 + F_2 - M_2, F_2)$
22	$(F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0)$
23	$(F_1 + \frac{F_2}{2} - M_2, F_1 + \frac{F_2}{2} - M_1, F_1 + \frac{F_2}{2} - M_2, \frac{F_2}{2})$
24	$(F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2, F_1 + F_2 - M_1)$
25	$(\frac{F_1 + F_2 + M_1}{2} - M_2, \frac{F_1 + F_2 - M_1}{2}, \frac{F_1 + F_2 + M_1}{2} - M_2, \frac{F_1 + F_2 - M_1}{2})$
26	$(0, F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2)$
27	$(F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2)$
28	$(\frac{F_1 + F_2 - M_1}{2}, \frac{F_1 + F_2 + M_1}{2} - M_2, \frac{F_1 + F_2 - M_1}{2}, \frac{F_1 + F_2 + M_1}{2} - M_2)$

vector $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)})$. Namely, given any (F_1, F_2, M_1, M_2), we first use Fig. 3.2 to find all the corner points in the PRCR (at most 6 of them). Then for each corner point, we plug in the closed-form expression in Table 3.3 to the objective function $\sum p_{\vec{d}} R_{\vec{d}}$. Repeat this process for each corner point. Finally the smallest objective value must be the average-rate capacity under the given (F_1, F_2, M_1, M_2) and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)})$. Two example results of this general procedure are provided as follows.

Example 2. *Suppose* $(F_1, F_2) = (1.5, 1)$ *and the demands of the users are statistically independent, with user 1 demanding files 1 and 2 with probability 2/3 and 1/3, respectively, and user 2 demanding files 1 and 2 with probability 2/5 and 3/5, respectively. The corresponding average-rate capacity for arbitrary* (M_1, M_2) *is described in Fig. 3.3.*

As discussed in the introduction, the main motivation of our study is to compare the optimal coded caching capacity with the performance of the naïve likelihood-based uncoded caching solution. For this particular example, we thus compare in Fig. 3.4 the optimal average-rate coded caching capacity with the performance of (i) the naïve likelihood-based uncoded caching, and (ii) the coded caching scheme in [41] that is optimized for the worst-case performance. As expected, the optimal coded caching capacity is always the smallest of the three and the largest rate reduction over the uncoded scheme is at v_{13} for which the optimal coded caching scheme uses only $\frac{1/2}{11/15} \simeq 68.2\%$ of the bandwidth of the uncoded solution.

It is also worth noting that at the corner point v_3, the worst-case-optimal coded caching scheme[3] actually performs worse than the uncoded scheme (5% worse) while the optimal coded scheme still exhibits 10% improvement over the uncoded solution.

Example 3. *Suppose* $(F_1, F_2) = (1.5, 1)$ *and the demands of the users are dependent with popularity* $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (\frac{2}{15}, \frac{8}{15}, \frac{4}{15}, \frac{1}{15})$. *Namely, user 1 demanding files 1 and 2 with probabilities 2/3 and 1/3, user 2 demanding files 1 and 2 with probability 2/5 and 3/5 but their demands are no longer statistically independent. Instead, the demands are*

[3]↑In general, the optimal scheme for the worst-case capacity may not be unique. A more precise statement should thus be "one worst-case optimal coded scheme actually performs ...". It is worth mentioning that it is an open problem how a system designer should choose from the set of optimal worst-case coded scheme since currently there is little study about what is the set of optimal worst-case coded schemes.

negatively correlated with correlation coefficient $-2\sqrt{5}/15$. *The corresponding average-rate capacity for arbitrary* (M_1, M_2) *is described in Fig. 3.5.*

Comparing Example 2 and 3, one can see that even with the same marginal distribution, the optimal coded caching can take into account the negative correlation, which results in a different capacity region.

We also compare the average rates of the optimal coded solution, the uncoded solution, and the worst-case-optimal coded solution with the setting of Example 3 in Fig. 3.6. The largest rate reduction over the uncoded scheme happens at v_{12} for which the optimal coded caching scheme uses only $\frac{7}{15}/\frac{11}{15} \simeq 63.6\%$ of the bandwidth of the uncoded solution.

The above examples consider user-dependent file popularity. If we relax that constraint and consider only uniform file popularity, we can derive a closed form capacity expression for any arbitrary (F_1, F_2, M_1, M_2).

Corollary 3. *For arbitrary* (F_1, F_2) *satisfying* $F_1 \geq F_2$ *and uniform file popularity (i.e.,* $p_{\vec{d}} = 0.25$, $\forall \vec{d}$*), the average-rate capacity for arbitrary* (M_1, M_2) *is described in Fig. 3.7, which contains exactly 5 facets.*

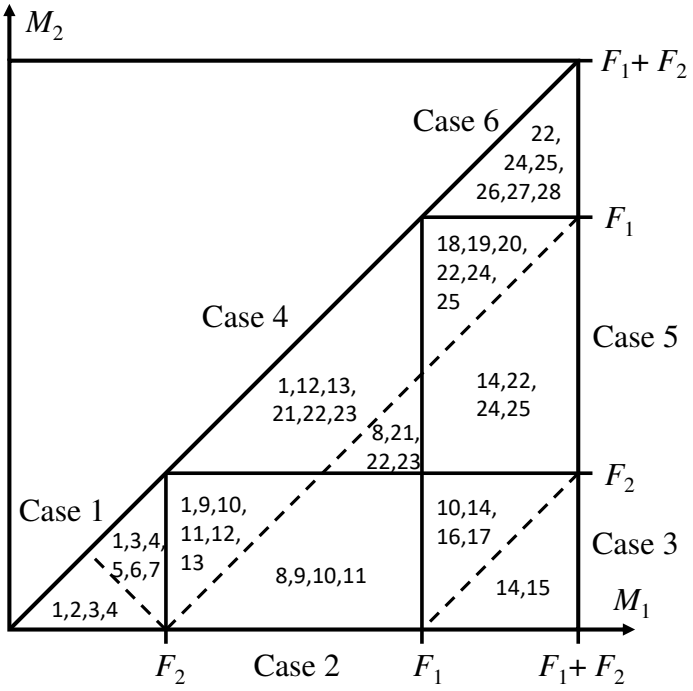
The proof of Corollary 3 is relegated to Appendix B.3.

The exact PRCR characterization can also be used to easily rederive the worst-case capacity R^* with arbitrary (F_1, F_2, M_1, M_2), previously found by examining the outer bounds of entropic cones [41]. See Appendix B.4 for details.

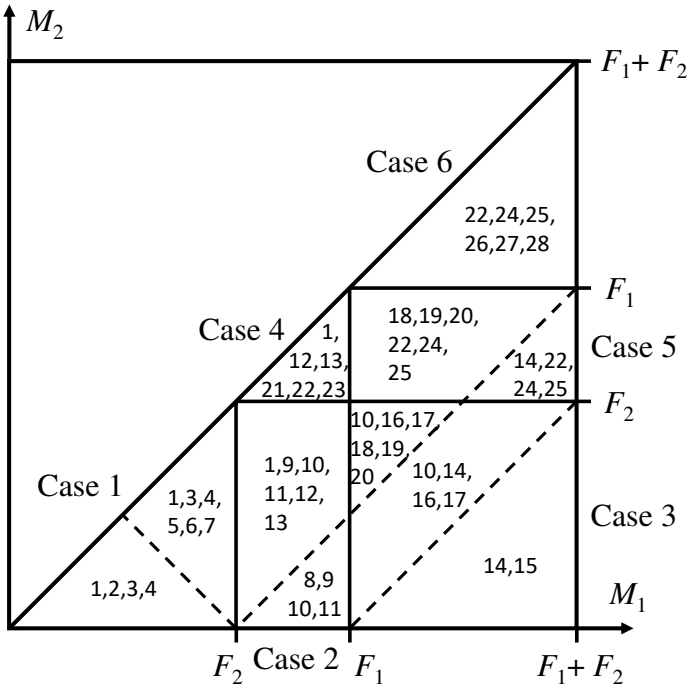
The closed form expressions of $\bar{R}$ and R^* as functions of (F_1, F_2, M_1, M_2) and $\{p_{\vec{d}}\}$, i.e., Corollary 3 and Corollary 7 in Appendix B.4, can be used to solve other design optimization problems. For example, we can solve the 2-user/2-file *memory allocation problem* [56] optimally by finding the (M_1^*, M_2^*) that minimizes $\bar{R}$ (or R^*) subject to the total memory constraint $M_1 + M_2 \leq M_{\text{total}}$. That is, we evaluate the coded caching capacity over the line $(M_1, M_2) = (m, M_{\text{total}} - m)$ for all $m \in [0, M_{\text{total}}]$. Then, the optimal allocation is simply $(M_1, M_2) = (m^*, M_{\text{total}} - m^*)$, where m^* denotes the value that leads to the smallest capacity rate.

3.4 Summary

The per-request capacity region (PRCR) is the most fundamental performance metric in the information-theoretic studies of coded caching. Given the PRCR of a coded caching problem, we can find the optimal coded caching schemes for any convex objective function. In this work, we have characterized the exact PRCR of the 2-user/2-file setting with full heterogeneity and used it to derive the average-rate capacity with heterogeneous demand popularity, file sizes, and cache sizes, and to re-derive the worst-case capacity previously found in [41]. By explicitly charactering the capacity and finding the capacity-achieving schemes, the results in this work allow the system designer to accurately evaluate the gain that optimal coded caching offers over naïve uncoded solutions under any general scenarios. The $N = K = 2$ results also represent the first step toward fully characterizing the average-rate/worst-case capacity of coded caching with full heterogeneity. Similar concepts and procedures may be applied to characterize/bracket the capacity for larger N and K: (i) Derive all the converse bounds of the N^K-dimensional PRCR, (ii) Find all the vertices of the PRCR polytope formed by the converse bounds for all $\{F_1, \ldots, F_N\}$ and $\{M_1, \ldots, M_K\}$ values, (iii) Find the achievable schemes for each of the vertices on the polytope. Such a process can be further simplified if there is any symmetry/homogeneity that can be exploited in the process. In addition, the result of $N = K = 2$ could shed new insights for deriving good/optimal achievability results for general N and K values. For example, it has been shown that the capacity-achieving schemes of the $N = K = 2$ case are based exclusively on space-sharing among 7 basic schemes. This observation is likely to be useful for the case of general N and K as well.



(a) $F_1 \geq 2F_2$



(b) $2F_2 > F_1 \geq F_2$

Figure 3.2. Description of the regions of (M_1, M_2) and the corresponding corner points. The x-axis (resp. y-axis) is for the M_1 (resp. M_2) value. In this figure we assume $F_1 \geq F_2$ and only describe the cases when $M_1 \geq M_2$, thus the lower-half of the line $M_1 = M_2$. The cases of $F_1 < F_2$ and $M_1 < M_2$ can be obtained by swapping the file and user indices, respectively. Two scenarios are considered: (a) $F_1 \geq 2F_2$; (b) $2F_2 > F_1 \geq F_2$.

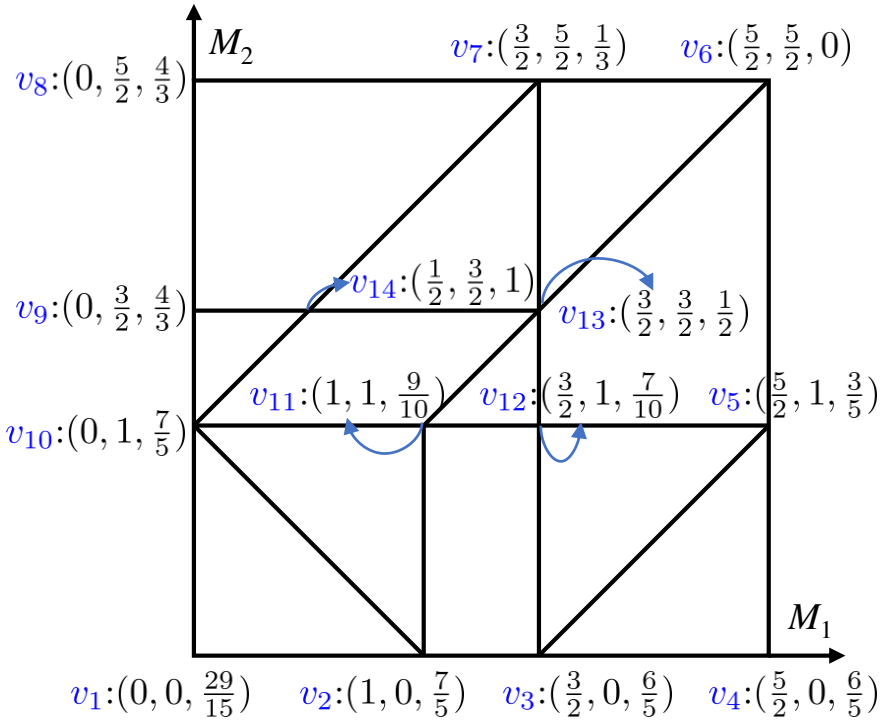


Figure 3.3. The average-rate capacity with $(F_1, F_2) = (1.5, 1)$ and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (4/15, 2/5, 2/15, 1/5)$. There are 12 facets and 14 corner points. Each corner point is labeled by a tuple $(M_1, M_2, \bar{R})$, where (M_1, M_2) give the location and the third coordinate specifies the corresponding exact average-rate capacity $\bar{R}$. The capacity is asymmetric with respect to (M_1, M_2) due to the heterogeneous file popularity.

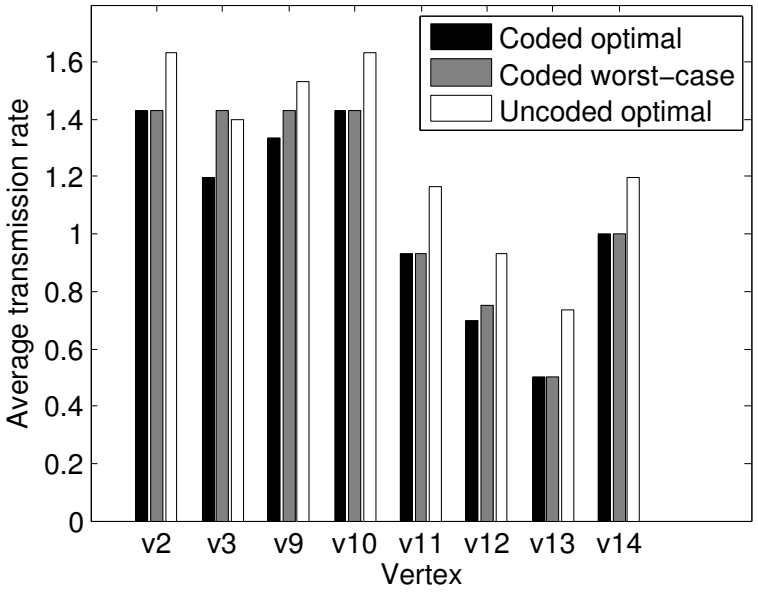


Figure 3.4. Comparison of the average-rate capacity with the average rate of naïve likelihood-based uncoded caching, and the coded caching scheme in [41] that is optimized for the worst-case performance on some of the vertices in Fig. 3.3.

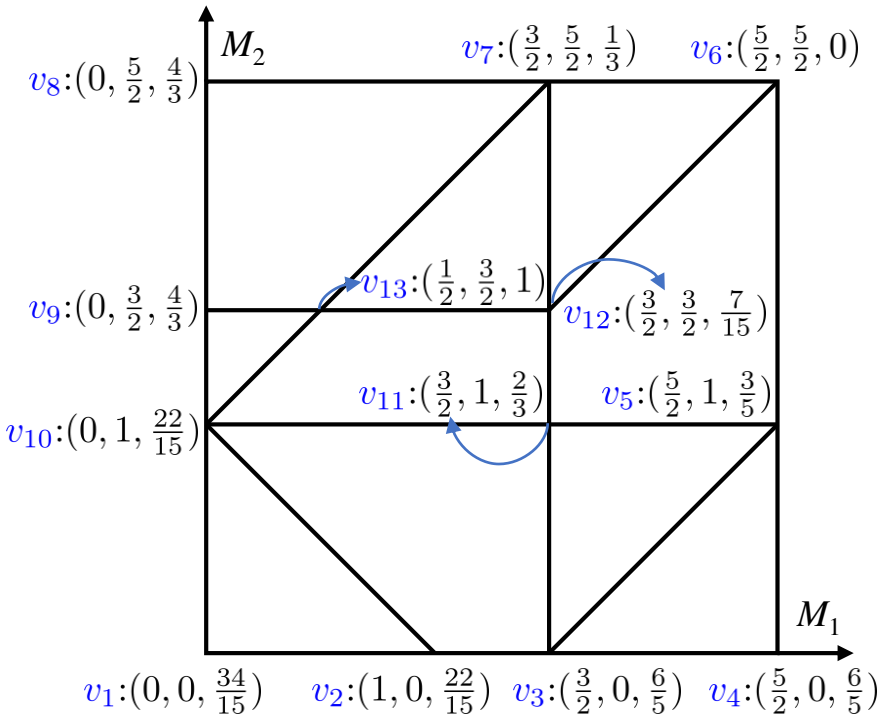


Figure 3.5. The average-rate capacity with $(F_1, F_2) = (1.5, 1)$ and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (2/15, 8/15, 4/15, 1/15)$. There are 10 facets and 13 corner points. Each corner point is labeled by a tuple $(M_1, M_2, \bar{R})$, where (M_1, M_2) give the location and the third coordinate specifies the corresponding exact average-rate capacity $\bar{R}$. The capacity is asymmetric with respect to (M_1, M_2) due to the heterogeneous file popularity.

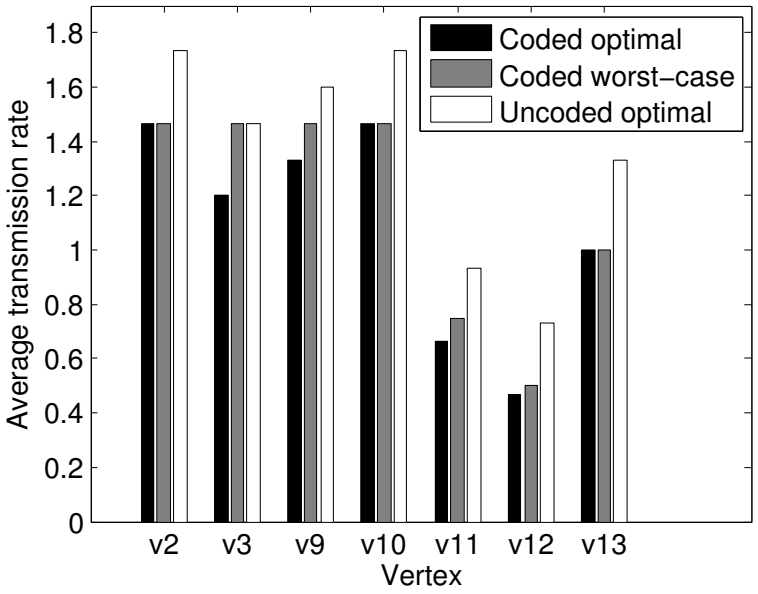


Figure 3.6. Comparison of the average-rate capacity with the average rate of naïve likelihood-based uncoded caching, and the coded caching scheme in [41] that is optimized for the worst-case performance on some of the vertices in Fig. 3.5.

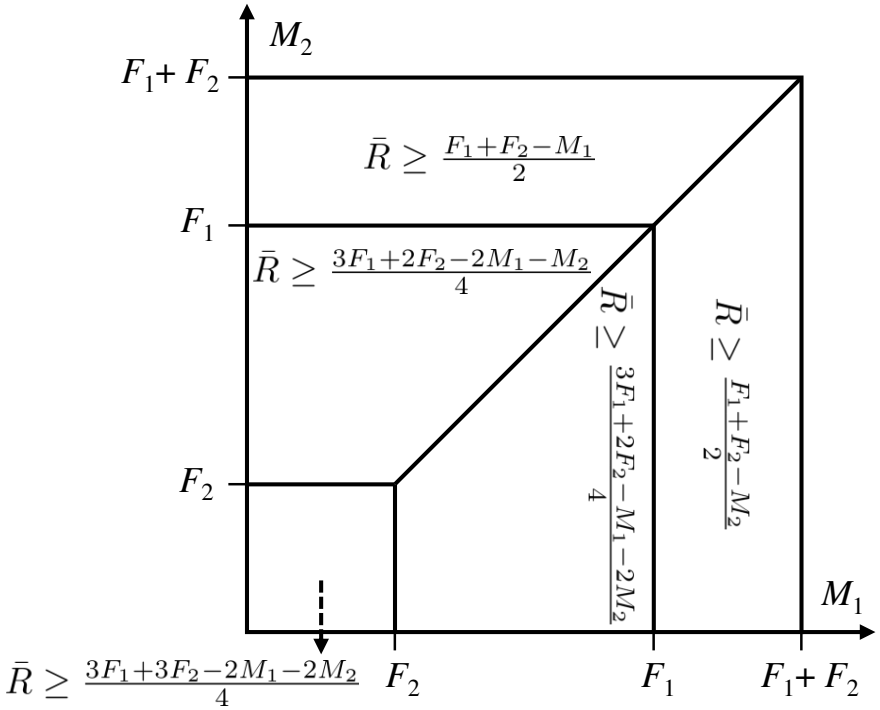


Figure 3.7. The average-rate capacity of uniform popularity, described for the case of $F_1 \geq F_2$.

4. CODED CACHING SYSTEM FOR TWO USERS WITH HETEROGENEOUS FILE DEMAND SETS

In this chapter, we consider the same coded caching system described in Chapter 3. The coded caching system consists of one server and K users, and operates in two phases: In placement phase, each user accesses the N files and fills its cache memory; and in delivery phase, each user submits its own file request and the server broadcasts a signal to deliver the desired packets to each user. Though we have discussed the exact capacity of the two-user/two-file ($K = N = 2$) coded caching system with full heterogeneous setting in Chapter 3. The average rate regions for general $K \geq 2$ and $N \geq 2$ coded caching are still open. In this part, we presents the first steps towards solving the problem by analyzing the case of two users $K = 2$ with distinct but dependent file popularity.

4.1 Introduction

The increasing demand of video streaming data has led to a significant challenge in content distribution over communication networks. Coded caching has recently received great success in reducing the peak transmission rate in some networks by exploiting multicasting opportunities in the underlying broadcast channels. A coded caching scheme has two phases: placement and delivery. In the placement phase, the users can access the files at the server to fill their cache memories during the off-peak hours. In the delivery phase, the users announce their requests and the server, with full knowledge of the users' cache contents, transmits the information required to satisfy such requests for all the users.

Caching has been extensively studied in content distribution networks with different objectives such as access latency and transmission rate [57], [58]. Traditional "uncoded caching" schemes focus on caching the content most likely to be requested and, when the content absent from the users' caches is requested, they deliver it in uncoded plain form. The most commonly used placement algorithms for uncoded caching are *least frequently used* (LFU) and *least recently used* (LRU), where the former retains the most frequently accessed subfiles in the past and the latter keeps the most recently used subfiles.

Coded caching, initially proposed in [31], can reduce the *worst-case* delivery time by a factor of $(\frac{1}{1+KM/FN})$ respect to traditional uncoded caching schemes, where N is the number of files, K is the number of users, F is the individual file size, and M is the individual cache size normalized by F. Existing works have characterized the coded caching capacity for some special N and K values [31]–[33], [52] and derived order-optimal capacity expressions for general N and K [31], [34], [39], [59], [60]. The worst-case setting is analytically appealing but it is not throughput optimal in practice with a time sequence of requests. Some recent results have focused on the *average-rate capacity* [36], [37], [39], [45], [61], evaluating the transmission rate over a distribution of demands. The order-optimal average-rate capacity results of *user-independent file popularity*, i.e., all users having the same file popularity profile has been investigated in [36], [37], [39], [45], [59], [61]. Nevertheless, the assumption of user-independent file popularity in these works is not compatible with traditional uncoded caching schemes, since the number of users with similar preferences can be small and each user often operates in a distributed manner with its own file popularity and prediction algorithm.

The rate improvement of coded caching primarily comes from the broadcasting nature and the simultaneous users' demands. However, the latter will not occur frequently and will require synchronization among the users. When the demands of users are served sequentially, coded caching increases the system complexity without providing any rate reduction. Clearly, the system should apply coded caching when users' demands are synchronized, and employ traditional uncoded caching when they come sparsely in time. One way to cover both scenarios is to enforce the so-called *uncoded prefetching* condition, which ensures that cached content is always of an uncoded form so that the content is useful to both the coded and uncoded delivery. It is obvious that such flexibility is at the expense of degrading the overall capacity since uncoded prefetching is a subclass of general prefetching schemes.

Motivated by the above discussion on limitations of user-independent file popularity and flexibility of uncoded prefetching, this work studies the average-rate capacity of coded caching with selfish and uncoded prefetching. Specifically, we assume that each user k is associated with a file demand set (FDS) Θ_k and requests each of those files with probability $\frac{1}{|\Theta_k|}$ (see Section 4.2 for details). Users will only cache segments from files in their demand set, hence the term *selfish* prefetching. The FDS setting reflects user-dependent file popularity

by considering different users k_1 and k_2 with $\Theta_{k_1} \neq \Theta_{k_2}$. Average-rate capacity results for disjoint FDS ($\Theta_{k_1} \cap \Theta_{k_2} = \emptyset$) and dominant FDS ($\Theta_{k_1} \subset \Theta_{k_2}$) were derived in [52], [62]. This work studies the case of two users ($K = 2$) and arbitrary number of files N, where the FDSs Θ_1 of user 1 and Θ_2 of user 2 overlap in α common files.

Under the coded caching model of file demand set, the coded caching problem for $K = 2$ users, arbitrary N files, and identical file demand sets $\Theta_1 = \Theta_2$ is the same as the original problem with homogeneous file popularity. We then focus on the coded caching problem with heterogeneous file demand sets $\Theta_1 \neq \Theta_2$. We first show the average-rate capacity when the two file demand sets Θ_1 and Θ_2 share a single common file, i.e., $\alpha = 1$, where the capacity can be achieved by selfish and uncoded prefetching. The we show the average rate capacity for the case $K = 2$, $N = 3$, and $\alpha = 2$, where the achievable scheme requires unselfish and coded prefetching. Finally we find the average-rate capacity of selfish and uncoded prefetching for an arbitrary number α of common files. From a practical perspective, this paper answers the following question: Suppose there are two users with significantly different file preferences (as is common in practice). However, there are α files that are simultaneously desired by both users, where α can be any number. What is the best coded caching scheme in this scenario?

4.2 Coded Caching Model with File Demand Set

We use the same model and notations of the coded caching system with one server and K users described in Section 3.2. The operation of the system consists of the *placement phase* and the *delivery phase*. In the placement phase, each user k populates its cache content by the caching function ϕ_k and in the delivery phase, each user k sends a request $d_k \in [N]$ to the server, i.e., user k demands file W_{d_k}. We denote the probability mass function (pmf) of the random request d_k by $p_{d_k}^{[k]}$. The joint pmf of the demand pattern of K users $\vec{d} \triangleq (d_1, \ldots, d_K) \in [N]^K$ is then $p_{\vec{d}} = p_{d_1}^{[1]} \cdots p_{d_K}^{[K]}$.

After receiving the demand index vector $\vec{d}$, the server broadcasts an encoded signal $X_{\vec{d}}$ of $R_{\vec{d}}$ bits with encoding function ψ using an error-free link to all K users. Each user k then uses $X_{\vec{d}}$ as well as his/her cache content Z_k to decode the requested file with the decoding

function μ_k. A coded caching scheme is completely specified by K caching functions $\{\phi_k\}$, one encoding function ψ, and K decoding functions $\{\mu_k\}$. Throughout this chapter, we consider exclusively the zero-error feasible schemes described in Definition 3.2.1.

Definition 4.2.1. *The* file demand set *(FDS) of user k is defined as $\Theta_k \triangleq \{n \in [N] : p_n^{[k]} > 0\}$, which is the set of files that user k desires with a strictly positive probability.*

Definition 4.2.2. *A coded caching scheme uses* uncoded prefetching *if the cache content of user k is*

$$Z_k = (w_1, \ldots, w_N) = \phi_k(W_1, \ldots, W_N), \ \forall k \in [K] \tag{4.1}$$

where each w_{i} is an uncoded subfile of W_{i}, $\mathrm{i} \in [N]$. That is, each user k stores uncoded fractions of the N files.

Definition 4.2.3. *A coded caching scheme uses* selfish prefetching *if all K caching functions ϕ_k in (3.1) can be replaced by*

$$Z_k = \phi_k(\{W_n : n \in \Theta_k\}), \ \forall k \in [K]. \tag{4.2}$$

Namely, each user k only stores the files that he/she is interested in, thus the name selfish. *In contrast, the more general design using (3.1) is referred to as an* unselfish *scheme.*

The general definitions of worst-case rate and average rate are provided in Definition 3.2.2 and Definition 3.2.3, respectively. Here we define the worst-case rate and average rate of coded caching schemes with file demand sets.

Definition 4.2.4. *The worst-case rate of a coded caching scheme is defined as*

$$R^* = \max_{\forall \vec{d}: d_k \in \Theta_k} R_{\vec{d}}. \tag{4.3}$$

Definition 4.2.5. *The average-rate of a coded caching scheme is defined as*

$$\bar{R} = \sum_{\forall \vec{d}, d_k \in \Theta_k} p_{\vec{d}} R_{\vec{d}}. \tag{4.4}$$

The uniform-average-rate of a scheme is defined as

$$\tilde{R} = \frac{1}{\prod_{k=1}^{K} |\Theta_k|} \sum_{\forall \vec{d}, d_k \in \Theta_k} R_{\vec{d}}. \tag{4.5}$$

$\tilde{R}$ can be viewed as a first-order approximation of the average-rate $\bar{R}$ that replaces the joint distribution $p_{\vec{d}}$ with a uniform distribution over the FDS $\prod_{k=1}^{K} \Theta_k$ (rather the simplest, uniform distribution over $[N]^K$ [39].). In [52], an exact characterization of $\bar{R}$ has been provided for the 2-user/2-file setting, which involves detailed discussion of up to 28 different cases that depends on the underlying values of (M_1, M_2) and $p_{\vec{d}}$. Instead of focusing on the exact $\bar{R}$, in this work we focus on the simplified, more tractable quantities $\tilde{R}$ and R^* but allow the N value to be ≥ 2.

For notational simplicity, we slightly abuse the above notation and directly use $\tilde{R}$ to denote the uniform-average-rate capacities even though the original notation $\tilde{R}$ in (4.5) represent the achievable rates instead. To investigate the capacity of a variety of schemes, we further denote the following four uniform-average-rate capacities: the uniform-average-rate capacity of unselfish and coded schemes $\tilde{R}_{\text{gc}}$ (or equivalently $\tilde{R}$); the uniform-average-rate capacity of unselfish and uncoded schemes $\tilde{R}_{\text{gu}}$; the uniform-average-rate capacity of selfish and coded schemes $\tilde{R}_{\text{sc}}$; and the uniform-average-rate capacity of selfish and uncoded schemes $\tilde{R}_{\text{su}}$. The relationship among the four capacities show the trade-off between the transmission rate and the complexity of achievable scheme designs.

4.3 Homogeneous File Demand Sets

When all the users have homogeneous (or identical) FDS, i.e., $\Theta_k = \Theta^*$ for all $k \in [K]$, the coded caching problem with FDS then degenerate to traditional N-file/K-user coded caching problem with the metric of uniform average transmission rate. According to Definition 4.2.3, there is no difference between selfish and unselfish designs, i.e., $\tilde{R}_{\text{gc}} = \tilde{R}_{\text{sc}}$ and $\tilde{R}_{\text{gu}} = \tilde{R}_{\text{su}}$. The capacity of uncoded prefetching schemes $\tilde{R}_{\text{gu}}$ (or $\tilde{R}_{\text{su}}$) can be solved by linear programming; however, the capacity of coded schemes $\tilde{R}_{\text{gc}}$ (or $\tilde{R}_{\text{su}}$) for arbitrary N and K are still open,

many order-optimal schemes have been proposed in 3.1. Specifically, the capacity result for $K = 2$ is provided as follows.

Proposition 4.3.1 ($\Theta_1 = \Theta_2$). *Consider $K = 2$ users, $N \geq 2$ files, and $\Theta_1 = \Theta_2 = [N]$. Then the uniform-average-rate $\tilde{R} = \tilde{R}_{\text{gc}} = \tilde{R}_{\text{sc}}$ is tightly characterized*[1] *by*

$$\tilde{R} \geq F - (M_1/N) \tag{P1}$$

$$\tilde{R} \geq F - (M_2/N) \tag{P2}$$

$$\tilde{R} \geq \frac{2N-1}{N}F - \frac{2N-2}{N^2}M_1 - \frac{1}{N}M_2 \tag{P3}$$

$$\tilde{R} \geq \frac{2N-1}{N}F - \frac{1}{N}M_1 - \frac{2N-2}{N^2}M_2 \tag{P4}$$

Corollary 4. *Under the restriction of uncoded prefetching, the capacity $\tilde{R}_{\text{gu}} = \tilde{R}_{\text{su}}$ is tightly characterized by [33].*

This proposition is the average-rate counterpart of the worst-case setting in [33] for the $K = 2$ and arbitrary $N \geq 2$ case. The relationship of $\tilde{R}$ versus (M_1, M_2) is illustrated in Fig. 4.1. The x-axis (resp. y-axis) is for the M_1 (resp. M_2) value. The inequalities (P1) to (P4) are marked in the corresponding regions. There are seven vertices s_1 to s_7 and each vertex is labeled by a tuple $(M_1, M_2, \tilde{R})$, where (M_1, M_2) describe the location and the third coordinate describe the corresponding exact uniform-average-rate capacity $\tilde{R}$.

4.4 Heterogeneous File Demand Sets

In this section, we show some coded caching results under heterogeneous FDS settings. Specifically, we consider arbitrary N files and K users with disjoint FDSs, i.e., $\Theta_{\text{i}} \cap \Theta_{\text{j}} = \emptyset$ for all $\text{i} \neq \text{j} \in [K]$, in Section 4.4.1; arbitrary N files and $K = 2$ users with one-overlapping FDSs, i.e., $|\Theta_1 \cap \Theta_2| = 1$, in Section 4.4.2; $N = 3$ files and $K = 2$ users with two-overlapping FDS i.e., $\Theta_1 = \{1, 2\}$, $\Theta_2 = \{1, 2, 3\}$, $|\Theta_1 \cap \Theta_2| = 2$ in Section 4.4.3, and arbitrary N_1, N_2 with FDS overlapped by ≥ 2 files in Section 4.4.4.

[1]↑We use the statement *tightly characterized* when we can derive a matching pair of the converse and achievability results, i.e., it characterizs capacity.

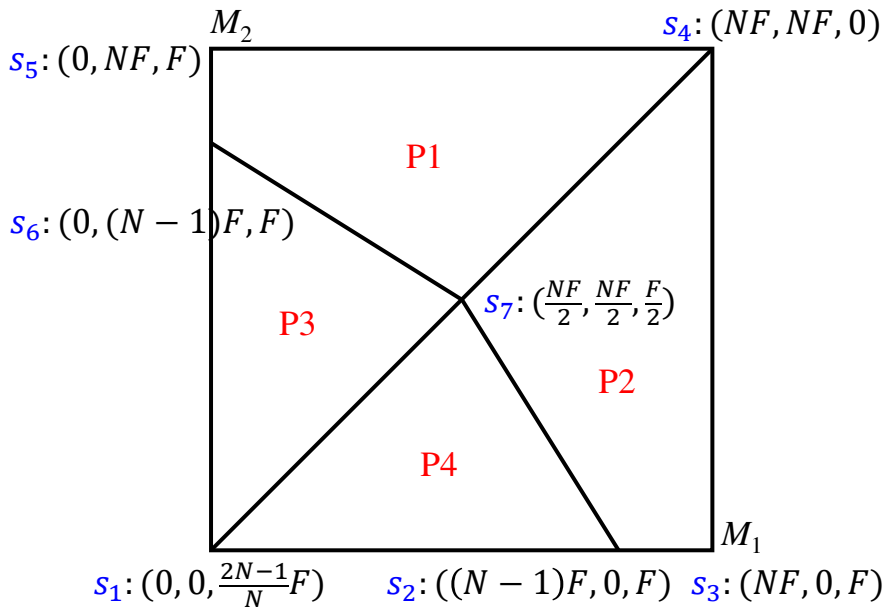


Figure 4.1. The capacity $\tilde{R}$ of both the selfish and unselfish designs with $\Theta_1 = \Theta_2 = [N]$.

4.4.1 Disjoint File Demand Sets for arbitrary K

We consider the coded caching of arbitrary N files and arbitrary $K \geq 2$ user, the uniform-average-rate capacity for disjoint FDSs is as follows.

Proposition 4.4.1. *If $\Theta_{k_1} \cap \Theta_{k_2} = \emptyset$ for all distinct $k_1, k_2 \in [K]$, then the uniform-average-rate $\tilde{R} = \tilde{R}_{\text{gc}} = \tilde{R}_{\text{sc}} = \tilde{R}_{\text{gu}} = \tilde{R}_{\text{su}}$ is tightly characterized by*

$$\tilde{R} = \sum_{k=1}^{K} \left(F - \frac{M_k}{|\Theta_k|} \right)^+ \tag{4.6}$$

The proof of Proposition 4.4.1 is relegated in Appendix C.1. This proposition shows if no two users are interested in a common file, each user can act as if he/she is the sole user in the system. Since $\tilde{R} = \tilde{R}_{\text{su}}$, the uniform-average-rate capacity (4.6) can be achieved by a simple selfish and uncoded prefeching scheme that user k caches $\min(F, M_k/|\Theta_k|)$ part of each file in Θ_k.

4.4.2 One-Overlapping File Demand Sets for $K = 2$

We consider the coded caching of arbitrary N files and $K = 2$ user, the uniform-average-rate capacity for one-overlapping FDSs is as follows.

Proposition 4.4.2 ($|\Theta_1| = N_1$, $|\Theta_2| = N_2$, $|\Theta_1 \cap \Theta_2| = 1$)**.** *Consider $K = 2$ users and $N = N_1 + N_2 - 1$ files with FDSs $\Theta_1 = \{1, \ldots, N_1\}$, $\Theta_2 = \{N_1, \ldots, N_1 + N_2 - 1\}$ such*

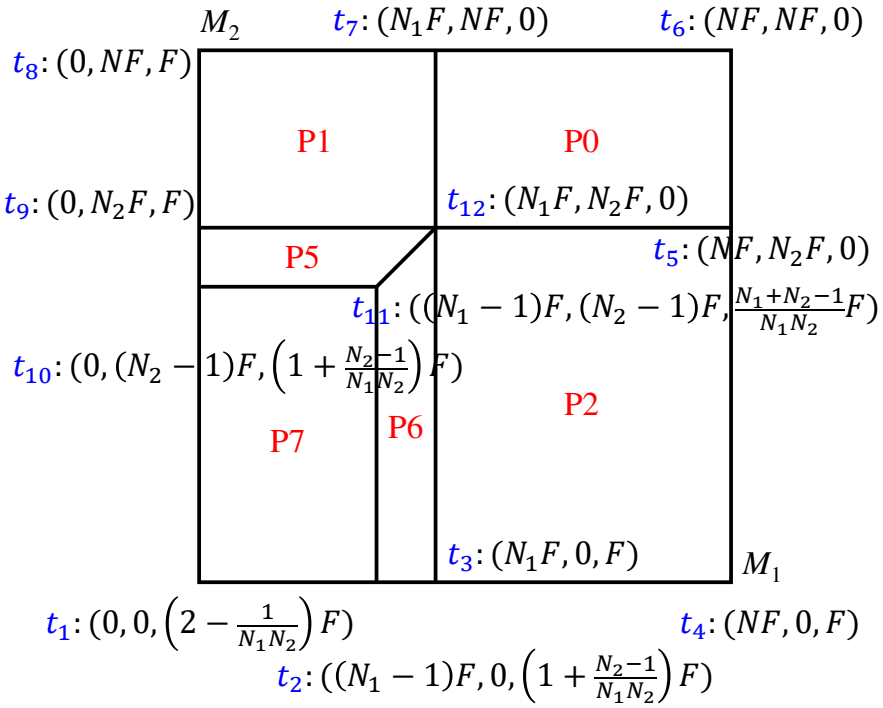


Figure 4.2. The minimum average rate $\tilde{R}$ of coded caching for $|\Theta_1| = N_1$, $|\Theta_2| = N_2$, and $\alpha = 1$. For any (M_1, M_2) inside each subregion, the rate $\tilde{R}$ is characterized by the corresponding equation marked in that region.

that $\Theta_1 \cap \Theta_2 = \{N_1\}$. *The uniform-average-rate* $\tilde{R} = \tilde{R}_{\text{gc}} = \tilde{R}_{\text{sc}} = \tilde{R}_{\text{gu}} = \tilde{R}_{\text{su}}$ *is tightly characterized by*

$$\tilde{R} \geq 0 \tag{P0}$$

$$\tilde{R} \geq F - M_1/N_1 \tag{P1}$$

$$\tilde{R} \geq F - M_2/N_2 \tag{P2}$$

$$\tilde{R} \geq \left(2 - \frac{1}{N_1}\right) F - \frac{M_1}{N_1} - \frac{N_1 - 1}{N_1 N_2} M_2 \tag{P5}$$

$$\tilde{R} \geq \left(2 - \frac{1}{N_2}\right) F - \frac{N_2 - 1}{N_1 N_2} M_1 - \frac{M_2}{N_2} \tag{P6}$$

$$\tilde{R} \geq \left(2 - \frac{1}{N_1 N_2}\right) F - \frac{M_1}{N_1} - \frac{M_2}{N_2} \tag{P7}$$

The relationship of $\tilde{R}$ *versus* (M_1, M_2) *is illustrated in Fig. 4.2.*

The proof of Proposition 4.4.2 is relegated in Appendix C.2. Since $\tilde{R} = \tilde{R}_{\text{su}}$, the rate $\tilde{R}$ in Prop. 4.4.2 can be achieved by selfish and uncoded schemes.

4.4.3 Two-Overlapping File Demand Sets for $N = 3$ and $K = 2$

Proposition 4.4.3 ($\Theta_1 = \{1, 2\}, \Theta_2 = \{1, 2, 3\}, \tilde{R}_{\text{gc}}$)**.** *Consider $K = 2$ users and $\Theta_1 = \{1, 2\}$ and $\Theta_2 = \{1, 2, 3\}$. The uniform-average-rate $\tilde{R} = \tilde{R}_{\text{gc}}$ is tightly characterized by*

$$\tilde{R} \geq F - M_1/2 \tag{Q1}$$

$$\tilde{R} \geq F - M_2/3 \tag{Q2}$$

$$\tilde{R} \geq \frac{5F}{4} - \frac{M_1}{4} - \frac{M_2}{4} \tag{Q3}$$

$$\tilde{R} \geq \frac{3F}{2} - \frac{M_1}{3} - \frac{M_2}{3} \tag{Q4}$$

$$\tilde{R} \geq \frac{5F}{3} - \frac{M_1}{2} - \frac{M_2}{3} \tag{Q5}$$

$$\tilde{R} \geq \frac{5F}{3} - \frac{M_1}{3} - \frac{M_2}{2} \tag{Q6}$$

The relationship of the capacity $\tilde{R}$ versus (M_1, M_2) is illustrated in Fig. 4.3.

The proof of Proposition 4.4.3 is provided in Appendix C.3.

Proposition 4.4.4 ($\Theta_1 = \{1, 2\}, \Theta_2 = \{1, 2, 3\}, \tilde{R}_{\text{sc}}$)**.** *Continue from Proposition 4.4.3. The rate $\tilde{R}_{\text{sc}}$ is tightly characterized by (Q1) to (Q6) plus an additional inequality:*

$$\tilde{R}_{\text{sc}} \geq \frac{4F}{3} - \frac{M_1}{6} - \frac{M_2}{3}. \tag{Q7}$$

The relationship of $\tilde{R}_{\text{sc}}$ versus (M_1, M_2) is illustrated in Fig. 4.4.

The proof of Proposition 4.4.4 is provided in Appendix C.4.

Note that one can prove that if $\tilde{R}_{\text{sc}}$ satisfies inequality (Q7), then it automatically satisfies (Q3) and (Q4). That is why in Fig. 4.4 there are only 5 subregions and the regions of (Q3) and (Q4) no longer appear.

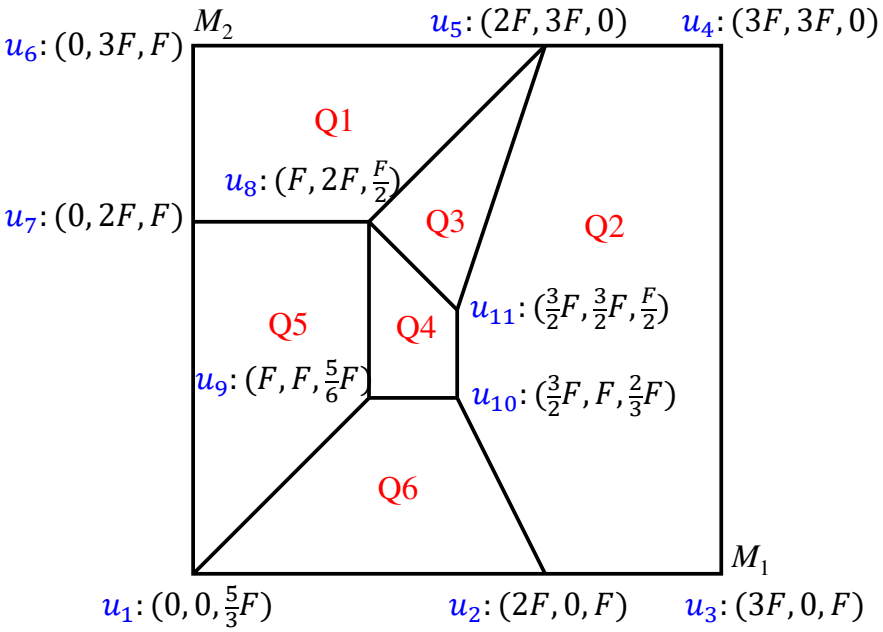


Figure 4.3. The uniform-average-rate capacity $\tilde{R} = \tilde{R}_{\text{gc}}$ with $\Theta_1 = \{1, 2\}$ and $\Theta_2 = \{1, 2, 3\}$.

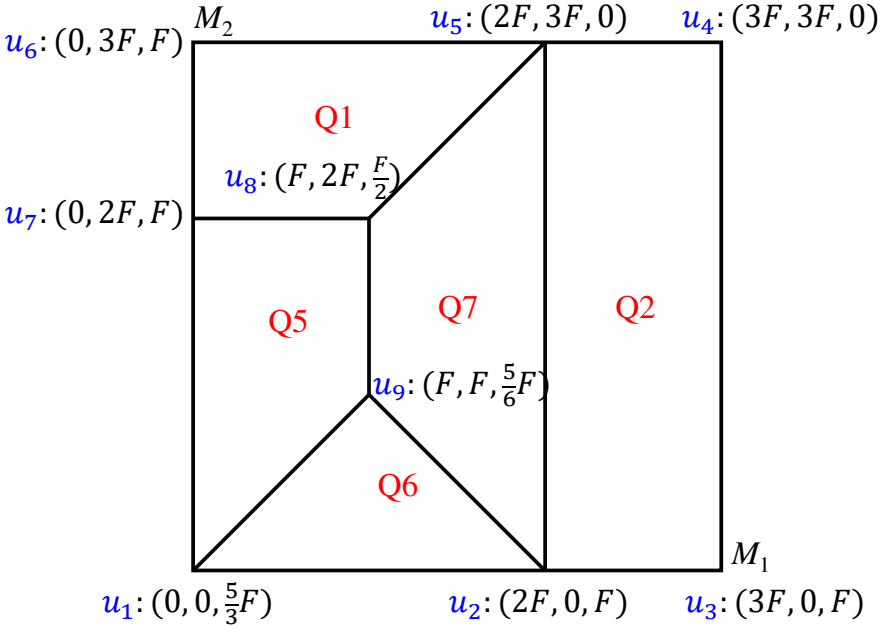


Figure 4.4. The uniform-average-rate capacity $\tilde{R}_{\text{sc}}$ for selfish and coded prefetching schemes with $\Theta_1 = \{1, 2\}$ and $\Theta_2 = \{1, 2, 3\}$.

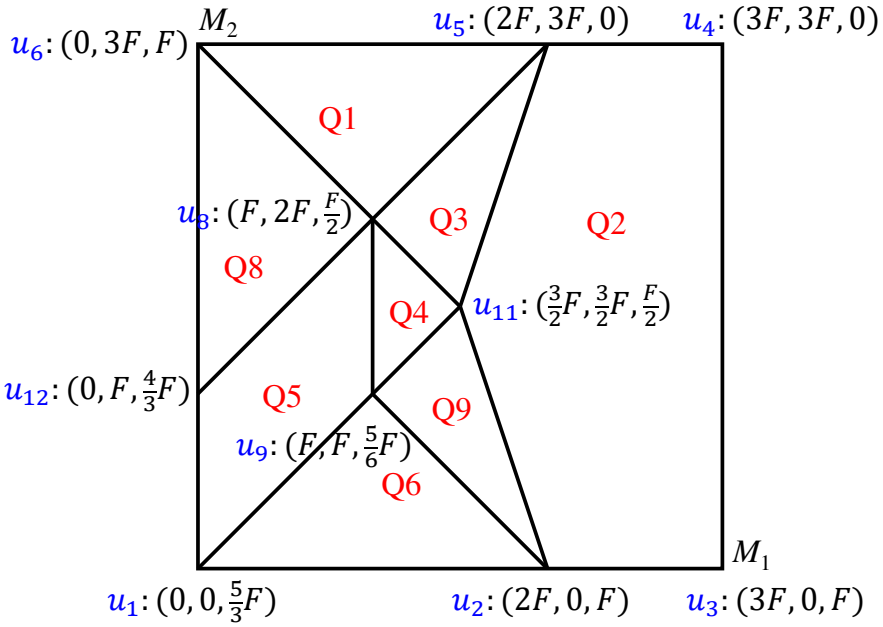


Figure 4.5. The uniform-average-rate capacity $\tilde{R}_{\text{gu}}$ for unselfish and uncoded prefetching schemes with $\Theta_1 = \{1, 2\}$ and $\Theta_2 = \{1, 2, 3\}$.

Proposition 4.4.5 ($\Theta_1 = \{1, 2\}, \Theta_2 = \{1, 2, 3\}, \tilde{R}_{\text{gu}}$)**.** *Continue from Proposition 4.4.3. The rate $\tilde{R}_{\text{gu}}$ is tightly characterized by (Q1) to (Q6) plus two additional inequalities:*

$$\tilde{R}_{\text{gu}} \geq \frac{3F}{2} - \frac{2}{3}M_1 - \frac{M_2}{6}. \tag{Q8}$$

$$\tilde{R}_{\text{gu}} \geq \frac{3F}{2} - \frac{M_1}{4} - \frac{5}{12}M_2. \tag{Q9}$$

The relationship of $\tilde{R}_{\text{gu}}$ versus (M_1, M_2) is illustrated in Fig. 4.5.

The proof of Proposition 4.4.5 is provided in Appendix C.5.

Proposition 4.4.6 ($\Theta_1 = \{1, 2\}, \Theta_2 = \{1, 2, 3\}, \tilde{R}_{\text{su}}$)**.** *Continue from Proposition 4.4.3. The rate $\tilde{R}_{\text{su}}$ is tightly characterized by (Q1), (Q2), (Q5) to (Q8).*

When viewed separately, Propositions 4.4.3 and 4.4.4 describe the fundamental limits of unselfish and selfish coded caching when two users, with arbitrary cache sizes (M_1, M_2), share concentrated[2], similar, but not identical interests, which alone are of important analytical

[2]↑We say a user is of *concentrated interest* if the corresponding FDS Θ_k is small, e.g., $|\Theta_1| = 2$ and $|\Theta_2| = 3$ in Propositions 4.4.3 and 4.4.4. This is usually a result of highly effective next-file prediction.

value. Jointly, they provide the first proof that selfish coded caching is strictly suboptimal, e.g., the two points u_{10} and u_{11} in Fig. 4.3 can only be achieved by an unselfish design.

It is worth pointing out that the insufficiency of selfish coded caching is not due to the use of the average rate $\tilde{R}$ as the performance metric. Even when using the worst-case rate R^* in (4.3), selfish designs are still insufficient.

Corollary 5. *Continue from Proposition 4.4.3. When $(M_1, M_2) = (1.5F, 1.5F)$, i.e., v_{11} in Fig. 4.3, the worst-case capacity R^* of the unselfish and selfish schemes are $0.5F$ and $\frac{7}{12}F$, respectively.*

4.4.4 Large-Overlapping File Demand Sets for $K = 2$ with Selfish and Uncoded Prefetching

Selfish and uncoded prefetching is not optimal in general, since it imposes unnecessary restrictions on the content of the caches. However, there are some cases in which it can proved that selfish and/or uncoded prefetching is optimal. One such example is Prop. 4.4.2 above and another example is provided in [52] for dominant $\Theta_1 = \{1\}$ and $\Theta_2 = \{1, \ldots, N\}$, $N \geq 2$. Furthermore, as discussed in Section 4.1 the simplicity of selfish and uncoded prefetching is a significant advantage when files, users, or links change dynamically.

This section provides an exact characterization of how the caches should be populated for a general M_1, M_2, N_1, N_2, and α (number of common files) in the case of $K = 2$ users with selfish and uncoded prefetching.

Proposition 4.4.7 ($|\Theta_1| = N_1$, $|\Theta_2| = N_2$, $|\Theta_1 \cap \Theta_2| \geq 2$, $\tilde{R}_{\text{su}}$)**.** *Consider $K = 2$ users and N files with file demand sets $\Theta_1 = \{1, \ldots, N_1\}$, $\Theta_2 = \{N_1 - \alpha + 1, \ldots, N_1 + N_2 - \alpha\}$ such that*

$|\Theta_1 \cap \Theta_2| = \alpha \geq 2$. *Without lost of generality, let* $M_2 \geq M_1$, *the uniform-average capacity* $\tilde{R}_{\text{su}}$ *for selfish and uncoded prefetching schemes is tightly characterized by*

$$\tilde{R}_{\text{su}} \geq 0 \tag{P0}$$

$$\tilde{R}_{\text{su}} \geq F - M_1/N_1 \tag{P1}$$

$$\tilde{R}_{\text{su}} \geq \left(2 - \frac{\alpha}{N_1}\right) F - \frac{M_1}{N_1} - \frac{N_1 - \alpha}{N_1 N_2} M_2 \tag{P8}$$

$$\tilde{R}_{\text{su}} \geq \left(2 - \frac{\alpha^2}{2N_1 N_2}\right) F - \frac{M_1}{N_1} - \frac{M_2}{N_2} \tag{P10}$$

$$\tilde{R}_{\text{su}} \geq \left(2 - \frac{1}{N_1}\right) F - \frac{N_2 + \alpha - 1}{N_1 N_2} M_1 - \frac{N_1 - 1}{N_1 N_2} M_2 \tag{P11}$$

$$\tilde{R}_{\text{su}} \geq \left(2 - \frac{\alpha}{N_1 N_2}\right) F - \frac{N_2 + \alpha - 2}{N_1 N_2} M_1 - \frac{M_2}{N_2} \tag{P13}$$

for the regions illustrated in Fig. 4.6. The equations for the remaining regions (i.e., $M_1 > M_2$*) can be found by symmetry, exchanging the roles of users 1 and 2.*

The proof of Proposition 4.4.7 is relegated in Appendix C.6.

Corollary 6. *When* $\alpha \geq 2$*, the minimum average rate in Proposition 4.4.7 can be achieved by the placement Algorithm 2.*

Algorithm 2 Optimal selfish and uncoded prefeching placement scheme for $\alpha \geq 2$

1: Input: $(N_1, N_2, \alpha, M_1, M_2)$
2: $\gamma \leftarrow \min(M_1, M_2, \frac{\alpha}{2} F)$
3: $M_1^u \leftarrow \min(0, M_1 - \gamma, (N_1 - \alpha)F)$
4: $M_2^u \leftarrow \min(0, M_2 - \gamma, (N_2 - \alpha)F)$
5: $M_1^c \leftarrow M_1 - M_1^u$
6: $M_2^c \leftarrow M_2 - M_2^u$
7: User 1 caches $[0, \frac{M_1^c}{\alpha}]$ part of each common file in $\mathcal{C}$ and caches $[0, \frac{M_1^u}{N_1 - \alpha}]$ part of each unique file in $\mathcal{U}_1$
8: User 2 caches $[F - \frac{M_2^c}{\alpha}, F]$ part of each common file in $\mathcal{C}$ and caches $[0, \frac{M_2^u}{N_2 - \alpha}]$ part of each unique file in $\mathcal{U}_2$

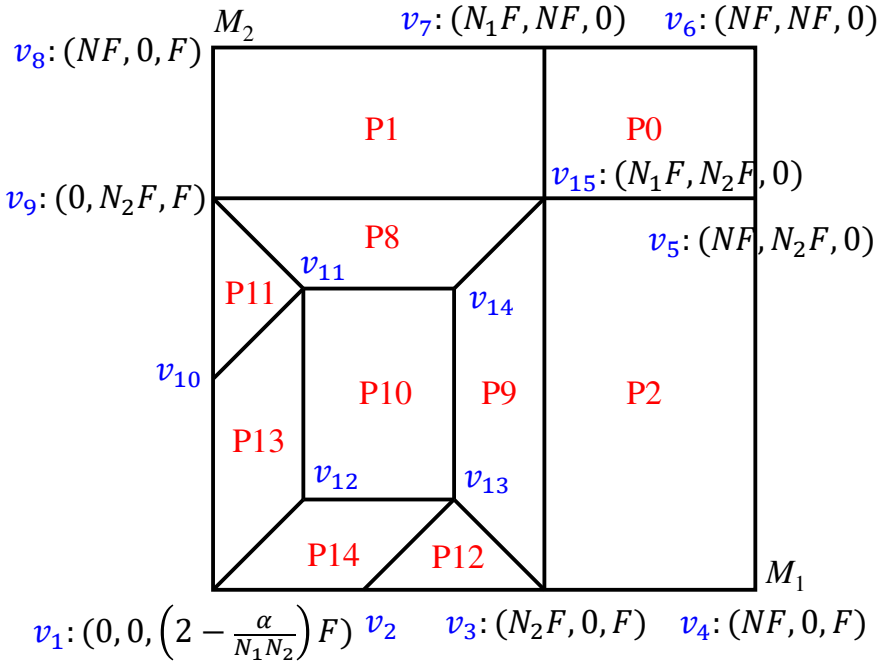


Figure 4.6. The minimum average rate $\tilde{R}$ for $|\Theta_1| = N_1$, $|\Theta_2| = N_2$, and $\alpha \geq 2$ with selfish and uncoded prefetching, where $v_2 : ((N_1 - \alpha)F, 0, (1 + (N_1 - 1)\alpha/(N_1N_2))F)$, $v_{10} : (0, (N_2 - \alpha)F, (1 + (N_2 - 1)\alpha/(N_1N_2))F)$, $v_{11} : (\alpha F/2, (N_2 - \alpha/2)F, (1 + (N_1 - N_2 - \alpha)\alpha/(2N_1N_2))F)$, $v_{12} : (\alpha F/2, \alpha F/2, (2 - (N_1 + N_2 + \alpha)\alpha/(2N_1N_2))F)$, $v_{13} : ((N_1 - \alpha/2)F, \alpha F/2, (1 + (N_2 - N_1 - \alpha)\alpha/(2N_1N_2))F)$, and $v_{14} : ((N_1 - \alpha/2)F, (N_2 - \alpha/2)F, (N_2 + N_1 - \alpha)\alpha F/(2N_1N_2))$.

4.5 Numerical Evaluations

This section illustrates the average rate capacity expressions in Fig. 4.6 for the particular case of $N_1 = N_2 = 128$ and $M_1 = M_2 = M$. It will compare the average rate with the optimal selfish and uncoded prefetching scheme described in Prop. 4.4.7 with two others:

- Uncoded transmission (UT): during the delivery phase, transmissions cannot encode multiple file segments into a single message. The server will send to each user whatever file segments it is missing from its cache, uncoded. There is no multicasting gain except when both users demand the same file, hence selfish prefetching is optimal. The minimum uniform-average rate is

$$\tilde{R}_{\text{ut}} \geq \left(2 - \frac{\alpha}{N_x^2}\right)\left(1 - \frac{M}{N_x}\right), \tag{4.7}$$

  where N_x denotes the value of $N_1 = N_2$.

- Shared caches (SC): users can share the content of their caches with each other. The two caches can therefore be treated as a single memory of size $2M$ that both users can access, without restricting the prefetching to be selfish or uncoded. The minimum uniform-average rate is

$$\tilde{R}_{\text{sc}} \geq \left(2 - \frac{\alpha}{N_x^2}\right)\left(1 - \frac{2M}{2N_x - \alpha}\right). \tag{4.8}$$

The UT scheme adds constraints to our system and therefore provides an upper bound to the general uniform-average rate capacity, while the SC scheme adds capabilities that our system did not have and therefore provides a lower bound to the general (including unselfish and coded prefetching) coded caching capacity.

Fig. 4.7 presents the uniform-average rates (normalized by the file size) for the three schemes when $\alpha = 96$ and the cache size per user grows from nothing ($M = 0$) to being able to cache all the files in the FDS ($M = 128$). The rate with our scheme falls between the other two, as expected, but it is interesting to note that for small M it is a lot closer

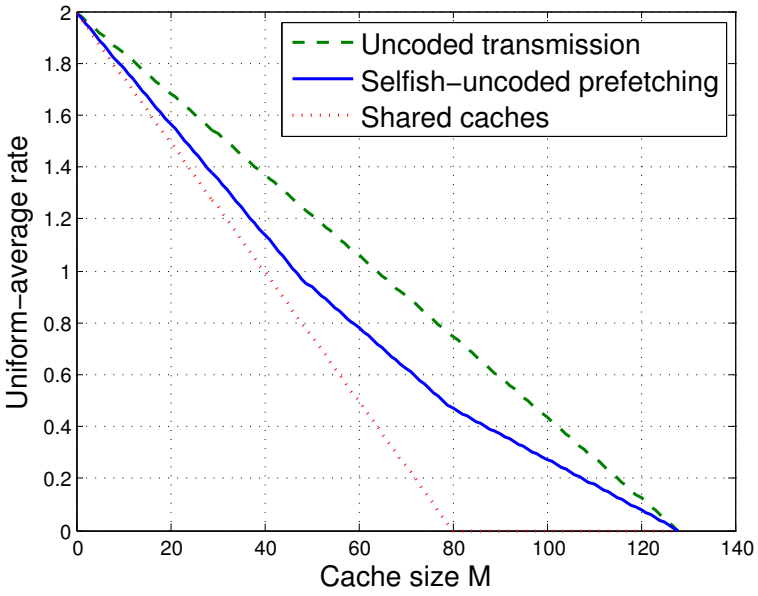


Figure 4.7. Uniform-average rate as a function of the cache size M when $N_1 = N_2 = 128$ and $\alpha = 96$.

to the SC than to the UT. This tells us that, even if we removed the selfish and uncoded prefetching restrictions, we would not be able to achieve a significant reduction in the rate.

Fig. 4.8 presents the uniform-average rates (normalized by the file size) for the three schemes when $M = 16$ and the intersection between the file demand sets (FDS) grows from empty ($\alpha = 0$) to a complete overlap ($\alpha = 128$). It can be observed that the UT scheme yields nearly constant rate regardless of the number of common files. This can be explained by the fact that, with $N_x = 128$ files in each demand set, the probability of both users requesting the same file is fairly low, even with a complete overlap. The uniform-average rate of our scheme is again relatively close to the capacity lower bound, specially when the overlap between the users' interests is small.

4.6 Summary

This chapter studies the average rate capacity of a coded caching system with two users having arbitrary cache capacities and demanding files from different but overlapping demand sets. Specifically, it analyzes the case of user 1 and user 2 with file demand sets Θ_1 and Θ_2, respectively. The chapter first provides the exact capacity results of one overlapped file

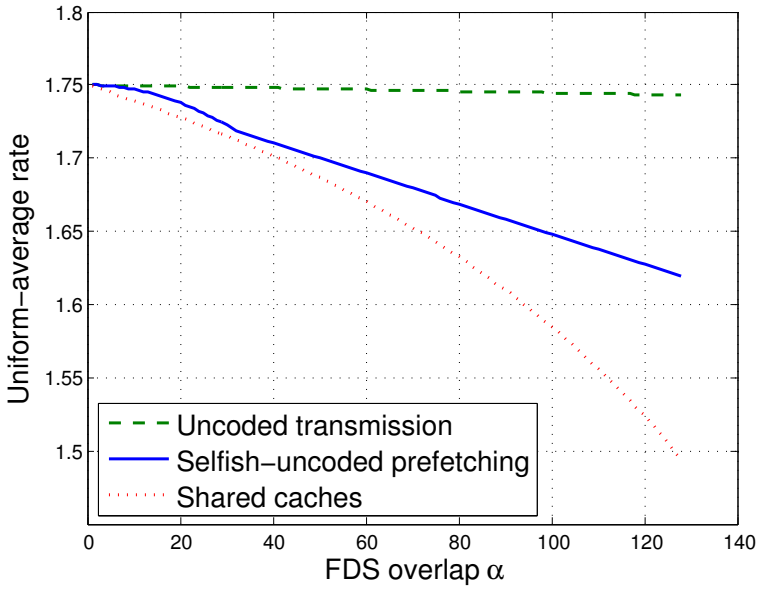


Figure 4.8. Uniform-average rate as a function of the FDS overlap α when $N_1 = N_2 = 128$ and $M = 16$.

demand sets $|\Theta_1 \cap \Theta_2| = 1$ for arbitrary N and two overlapped file demand sets $|\Theta_1 \cap \Theta_2| = 2$ for $N = 3$. To investigate the performance of large overlapped files, the chapter then focuses on this scenario of selfish and uncoded prefetching and derived explicit expressions for the minimal achievable average rate when the file demand sets have arbitrary size and number of overlap. Finally, we provide detailed placement and delivery schemes capable of achieving the derived capacities. Simulation results are used to illustrate our results and bound their gap to an optimal unselfish and coded prefetching scheme.

5. CONCLUSION AND FUTURE WORK

The thesis discusses the network coding applications in some less restrictive network models, including the 1-to-K broadcast packet erasure channel and the coded caching system with N files and K users. In this chapter, we summarize the thesis and provide some thoughts for future research directions.

5.1 Capacity of 1-to-K Spatial-Independent Broadcast Packet Erasure Channels with ACK/NACK

In Chapter 2, we propose the sequential coding protocol for 1-to-K broadcast packet erasure channels that attains the capacity region of known scenarios in Proposition 2.3.1. However, from Section 2.3 and Appendix A.2.3, we observe that the proposed scheme can achieve the outer bound for *spatially independent* channels (through examining over 10^5 numerical trials with different $\vec{p}$). However, if $K = 4$ and the underlying channel is not spatially independent, we have found some rate vectors $\vec{R}$ that are in the outer bound but not in the proposed stability region.

Fortunately, in most of wireless broadcast scenarios, the channels from the server to users has very small correlation among each other and hence can be viewed as independent. As the result, a scheme that achieves the capacity of spatially-independent broadcast PECs is useful in real design. Based on our examination, we conjecture that the proposed scheme can achieve the capacity of 1-to-4 spatially-independent broadcast PECs, i.e., the rate region obtained by substituting $K = 4$ and $p_{\cup S_j^\pi} = 1 - \prod_{k \in S_j^\pi}(1 - p_k)$ in (2.5). For best of our knowledge, the achievable scheme for $K = 4$ and spatially independent channels is still open without a complete proof. Moreover, due to the systematic structure of proposed inner bound in Proposition 2.3.1, a more interesting and ambitious research direction is to prove if the proposed scheme can achieve the capacity of the 1-to-K spatially independent broadcast PECs for all $K \geq 4$.

5.2 Linear Network Coding for Achieving Capacity of 1-to-K Broadcast Packet Erasure Channels with ACK/NACK

In the literature of network coding, most of the achievable schemes apply only the linear coding technique to approach the capacity; however, existing results [63] have shown the insufficiency of linear coding in some similar broadcasting models. In Chapter 2, we show that the proposed linear network coding scheme successfully achieves the capacity under the scenarios in Proposition 2.3.1 but fails under some scenarios of $K = 4$, e.g., the one described in Appendix A.2.3. In order to close the gap between the outer bound and proposed inner bound for $K = 4$, we need to prove one of the following assumptions.

First, under the assumption that the linear network coding is sufficient to achieve the capacity of 1-to-4 broadcast PECs, we need to construct a tighter outer bound to match the network coding scheme or to proposed a more advanced linear network coding scheme as a tighter inner bound. In this situation, as linear coding is well-investigated in the literature, we may want to actually propose an explicit linear network coding scheme and close the gap between the capacity outer and inner bound. Second, under the assumption that the linear network coding is not sufficient to achieve the capacity of 1-to-4 broadcast PECs, we need to construct a scenario of 1-to-4 broadcast PEC where all the linear coding schemes can not achieve the complete capacity region. As a negative proof in this situation, we may not need to construct a specific achievable non-linear network coding scheme.

5.3 General Lower Bounds for Coded Caching of Arbitrary N Files and K Users

In order to investigate the coded caching capacity or optimal rate for arbitrary number of files N and number of users K with full heterogeneity, we intend to achieve the outer bound with coded caching schemes. In the coded caching model, since each user demands one file and the server broadcasts a signal once, designing the achievability schemes or inner bound is much easier than designing the inner bound of 1-to-K broadcast packet erasure channels. However, on the other hand, constructing tight outer bounds for the coded caching system requires much more efforts. In general coded caching models, the per-request capacity region is determined by parameters file size F_{i} of file i and cache memory size M_k of user k, which

indicates that the information-theoretic outer bounds only involve the variables of file content W_{i} of file i and cache memory content Z_k of user k without any jointed or conditioned terms among the variables. Therefore when constructing the information-theoretic outer bounds, we should carefully eliminate these undesired jointed or conditioned terms in the equations. Prior studies [32], [33] exploit the file and user symmetry to develop new techniques for constructing the outer bounds with homogeneity and worst-case rate settings. To form the outer bound for heterogeneity and average rate settings, without the file and user symmetry, we should develop more general and advanced techniques for deriving tighter outer bounds.

5.4 Coded Caching with $K = 2$ user and Heterogeneous User File Popularity

In order to reduce the high complexity of coded caching problem with heterogeneous settings, we propose a simplified model in Chapter 4 with binary request probabilities. Specifically, we consider a coded caching system that each user demands a file with uniform probability among the files in his/her file demand set and with zero probability for the files outside the file demand set. We then propose the capacity results for the scenario of $K = 2$ users, arbitrary N files, and the two users' file demand sets Θ_1 and Θ_2 overlapping by one file. However, the coded caching problem for $K = 2$ users, arbitrary N files, and arbitrary file demand sets with arbitrary overlapped files, is still open.

Solving the coded caching problem for the scenario of $K = 2$ users, arbitrary N files, and arbitrary file demand sets, requires further derivation and verification of the outer and inner bounds. Under the assumption that linear coding is sufficient for achieving the capacity, we can straightforwardly obtain the linear achievable scheme as an inner bound. However, we may need more sophisticated techniques to derive tight outer bounds to match the linear coding achievable scheme. On the other hand, if linear network coding is not sufficient to achieve the capacity of the coded caching system, we may need to illustrate a scenario as a negative proof that all the linear coding schemes can not achieve the capacity rate region.

REFERENCES

[1] T. M. Cover, "Comments on broadcast channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2524–2530, Oct. 1998, ISSN: 0018-9448. DOI: 10.1109/18.720547.

[2] P. Bergmans, "Random coding theorem for broadcast channels with degraded components," *IEEE Trans. Inf. Theory*, vol. 19, no. 2, pp. 197–207, Mar. 1973, ISSN: 0018-9448. DOI: 10.1109/TIT.1973.1054980.

[3] Y. Wu, "Broadcasting when receivers know some messages a priori," in *Proc. IEEE Int. Symp. Inform. Theory*, Jun. 2007, pp. 1141–1145. DOI: 10.1109/ISIT.2007.4557377.

[4] M. A. Maddah-Ali and D. Tse, "Completely stale transmitter channel state information is still very useful," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4418–4431, Jul. 2012, ISSN: 0018-9448. DOI: 10.1109/TIT.2012.2193116.

[5] C.-C. Wang, "On the capacity of 1-to-K broadcast packet erasure channels with channel output feedback," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 931–956, Feb. 2012, ISSN: 0018-9448. DOI: 10.1109/TIT.2011.2173723.

[6] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback - capacity and algorithms," in *Proc. Workshop on Network Coding, Theory, and Applications (Netcod)*, Jun. 2009, pp. 54–61. DOI: 10.1109/NETCOD.2009.5191394.

[7] C.-C. Wang and J. Han, "The capacity region of two-receiver multiple-input broadcast packet erasure channels with channel output feedback," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5597–5626, Sep. 2014, ISSN: 0018-9448. DOI: 10.1109/TIT.2014.2334299.

[8] J. Han and C.-C. Wang, "General capacity region for the fully connected three-node packet erasure network," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5503–5523, Oct. 2016, ISSN: 0018-9448. DOI: 10.1109/TIT.2016.2600578.

[9] R. Y. Chang, S.-J. Lin, and W.-H. Chung, "Symbol and bit mapping optimization for physical-layer network coding with pulse amplitude modulation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 3956–3967, Aug. 2013, ISSN: 1536-1276. DOI: 10.1109/TWC.2013.071613.121520.

[10] C.-H. Chang, R. Y. Chang, and Y.-C. Huang, "A comparative analysis of secrecy rates of wireless two-way relay systems," in *Proc. IEEE GLOBECOM*, Dec. 2015, pp. 1–6. DOI: 10.1109/GLOCOM.2015.7417092.

[11] I.-H. Hou, V. Borkar, and P. R. Kumar, "A theory of qos for wireless," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 486–494. DOI: 10.1109/INFCOM.2009.5061954.

[12] I.-H. Hou and P. R. Kumar, "Utility maximization for delay constrained qos in wireless," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9. DOI: 10.1109/INFCOM.2010.5462070.

[13] I.-H. Hou, "Scheduling heterogeneous real-time traffic over fading wireless channels," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1631–1644, Oct. 2014, ISSN: 1063-6692. DOI: 10.1109/TNET.2013.2280846.

[14] W.-C. Kuo and C.-C. Wang, "Robust and optimal opportunistic scheduling for downlink two-flow network coding with varying channel quality and rate adaptation," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 465–479, Feb. 2017, ISSN: 1063-6692. DOI: 10.1109/TNET.2016.2583488.

[15] X. Li, C.-C. Wang, and X. Lin, "Inter-session network coding schemes for 1-to-2 downlink access-point networks with sequential hard deadline constraints," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 624–638, Feb. 2017, ISSN: 1063-6692. DOI: 10.1109/TNET.2016.2599116.

[16] L. Deng, C.-C. Wang, M. Chen, and S. Zhao, "Timely wireless flows with general traffic patterns: Capacity region and scheduling algorithms," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3473–3486, Dec. 2017, ISSN: 1063-6692. DOI: 10.1109/TNET.2017.2749513.

[17] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Multiuser broadcast erasure channel with feedback–capacity and algorithms," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 5779–5804, Sep. 2013, ISSN: 0018-9448. DOI: 10.1109/TIT.2013.2265692.

[18] S. Athanasiadou, M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Stable XOR-based policies for the broadcast erasure channel with feedback," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 476–491, Feb. 2016, ISSN: 1063-6692. DOI: 10.1109/TNET.2014.2366435.

[19] W. Nam, S.-Y. Chung, and Y. H. Lee, "Capacity of the gaussian two-way relay channel to within $\frac{1}{2}$ bit," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5488–5494, Nov. 2010, ISSN: 0018-9448. DOI: 10.1109/TIT.2010.2069150.

[20] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proc. Mobile Computing and Networking (MobiCom)*, ser. MobiCom '06, New York, NY, USA: ACM, 2006, pp. 358–365, ISBN: 1-59593-286-0. DOI: 10.1145/1161089.1161129.

[21] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011, ISSN: 0018-9448. DOI: 10.1109/TIT.2011.2165816.

[22] A. Eryilmaz, A. Ozdaglar, M. Medard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5511–5524, Dec. 2008, ISSN: 0018-9448. DOI: 10.1109/TIT.2008.2006454.

[23] L. Yang, Y. E. Sagduyu, J. Zhang, and J. H. Li, "Deadline-aware scheduling with adaptive network coding for real-time traffic," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1430–1443, Oct. 2015, ISSN: 1063-6692. DOI: 10.1109/TNET.2014.2331018.

[24] Y. E. Sagduyu, L. Georgiadis, L. Tassiulas, and A. Ephremides, "Capacity and stable throughput regions for the broadcast erasure channel with feedback: An unusual union," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2841–2862, May 2013, ISSN: 0018-9448. DOI: 10.1109/TIT.2011.2171180.

[25] A. Papadopoulos and L. Georgiadis, "Broadcast erasure channel with feedback and message side information, and related index coding result," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3161–3180, May 2017, ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2668386.

[26] Z. Li, C. He, and S. Yang, "On the capacity of the two-user erasure broadcast channel with mixed csit," in *Proc. IEEE Information Theory Workshop (ITW)*, Nov. 2017, pp. 499–503. DOI: 10.1109/ITW.2017.8277984.

[27] C. Chang and C. Wang, "A new capacity-approaching scheme for general 1-to-k broadcast packet erasure channels with ack/nack," *IEEE Trans. Inf. Theory*, vol. 66, no. 5, pp. 3000–3025, 2020. DOI: 10.1109/TIT.2020.2968316.

[28] C.-C. Wang, D. Koutsonikolas, Y. C. Hu, and N. Shroff, "FEC-based AP downlink transmission schemes for multiple flows: Combining the reliability and throughput enhancement of intra- and inter-flow coding," *Perform. Eval.*, vol. 68, no. 11, pp. 1118–1135, Nov. 2011, ISSN: 0166-5316. DOI: 10.1016/j.peva.2011.07.021.

[29] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003, ISSN: 0018-9448. DOI: 10.1109/TIT.2002.807285.

[30] J. Wang, "A survey of web caching schemes for the internet," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 5, pp. 36–46, Oct. 1999, ISSN: 0146-4833. DOI: 10.1145/505696.505701. [Online]. Available: http://doi.acm.org/10.1145/505696.505701.

[31] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014, ISSN: 0018-9448. DOI: 10.1109/TIT.2014.2306938.

[32] C. Tian, "Symmetry, demand types and outer bounds in caching systems," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jul. 2016, pp. 825–829. DOI: 10.1109/ISIT.2016.7541414.

[33] D. Cao, D. Zhang, P. Chen, N. Liu, W. Kang, and D. Gündüz, "Coded caching with asymmetric cache sizes and link qualities: The two-user case," *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6112–6126, Sep. 2019, ISSN: 1558-0857. DOI: 10.1109/TCOMM.2019.2921711.

[34] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015, ISSN: 1063-6692. DOI: 10.1109/TNET.2014.2317316.

[35] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821–5833, Sep. 2017, ISSN: 1557-9654. DOI: 10.1109/TIT.2017.2725272.

[36] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1146–1158, Feb. 2017, ISSN: 0018-9448. DOI: 10.1109/TIT.2016.2639522.

[37] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 349–366, Jan. 2018, ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2768517.

[38] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3923–3949, Jun. 2017, ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2695611.

[39] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan. 2019, ISSN: 0018-9448. DOI: 10.1109/TIT.2018.2870566.

[40] H. Ghasemi and A. Ramamoorthy, "Asynchronous coded caching," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jun. 2017, pp. 2438–2442. DOI: 10.1109/ISIT.2017.8006967.

[41] C. Li, "On rate region of caching problems with non-uniform file and cache sizes," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 238–241, Feb. 2017, ISSN: 1089-7798. DOI: 10.1109/LCOMM.2016.2594762.

[42] A. M. Daniel and W. Yu, "Optimization of heterogeneous coded caching," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1893–1919, 2020.

[43] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis., "Finite-length analysis of caching-aided coded multicasting," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct. 2016, ISSN: 1557-9654. DOI: 10.1109/TIT.2016.2599110.

[44] E. Ozfatura and D. Guenduez, "Uncoded caching and cross-level coded delivery for non-uniform file popularity," in *Proc. IEEE Int. Conf. on Communications (ICC)*, May 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422960.

[45] J. Hachem, N. Karamchandani, and S. N. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3108–3141, May 2017, ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2664817.

[46] P. Quinton, S. Sahraei, and M. Gastpar, "A novel centralized strategy for coded caching with non-uniform demands," *arXiv:1801.10563*, Jan. 2018. arXiv: 1801.10563 [cs.IT].

[47] S. Sahraei, P. Quinton, and M. Gastpar, "The optimal memory-rate trade-off for the non-uniform centralized caching problem with two files under uncoded placement," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 7756–7770, Dec. 2019, ISSN: 1557-9654. DOI: 10.1109/TIT.2019.2930692.

[48] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Coded caching for heterogeneous systems: An optimization perspective," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5321–5335, Aug. 2019, ISSN: 1558-0857. DOI: 10.1109/TCOMM.2019.2914393.

[49] J. Zhang, X. Lin, C. Wang, and X. Wang, "Coded caching for files with distinct file sizes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jun. 2015, pp. 1686–1690. DOI: 10.1109/ISIT.2015.7282743.

[50] L. Zheng, Q. Chen, Q. Yan, and X. Tang, "Decentralized coded caching scheme with heterogenous file sizes," *IEEE Trans. Veh. Technol.*, pp. 1–1, 2019, ISSN: 1939-9359. DOI: 10.1109/TVT.2019.2949979.

[51] J. Zhang, X. Lin, and C. Wang, "Closing the gap for coded caching with distinct file sizes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jul. 2019, pp. 687–691. DOI: 10.1109/ISIT.2019.8849540.

[52] C. Chang and C. Wang, "Coded caching with full heterogeneity: Exact capacity of the two-user/two-file case," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2019, pp. 6–10. DOI: 10.1109/ISIT.2019.8849597.

[53] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, 2005. DOI: 10.1109/TIT.2005.851744.

[54] H. Ghasemi and A. Ramamoorthy, "Improved lower bounds for coded caching," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4388–4413, Jul. 2017, ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2705166.

[55] C. Wang, S. Saeedi Bidokhti, and M. Wigger, "Improved converses and gap results for coded caching," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7051–7062, Nov. 2018, ISSN: 1557-9654. DOI: 10.1109/TIT.2018.2856885.

[56] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Optimization of heterogeneous caching systems with rate limited links," in *Proc. IEEE Int. Conf. on Communications (ICC)*, May 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7997394.

[57] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9. DOI: 10.1109/INFCOM.2010.5461964.

[58] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 568–582, Oct. 2000. DOI: 10.1109/90.879344.

[59] T. Luo, V. Aggarwal, and B. Peleato, "Coded caching with distributed storage," *IEEE Trans. Inf. Theory*, pp. 1–1, 2019. DOI: 10.1109/TIT.2019.2940979.

[60] T. Luo and B. Peleato, "The transfer load-i/o trade-off for coded caching," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1524–1527, Aug. 2018. DOI: 10.1109/LCOMM.2018.2840149.

[61] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3923–3949, Jun. 2017, ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2695611.

[62] S. Wang and B. Peleato, "Coded caching with heterogeneous user profiles," in *Proc. IEEE Int. Symp. Inform. Theory*, Jul. 2019, pp. 2619–2623. DOI: 10.1109/ISIT.2019.8849537.

[63] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005, ISSN: 0018-9448. DOI: 10.1109/TIT.2005.851744.

[64] A. Bondy and U. Murty, *Graph Theory*, ser. Graduate Texts in Mathematics. Springer London, 2011, ISBN: 9781846289699.

[65] M. J. Neely and R. Urgaonkar, "Opportunism, backpressure, and stochastic optimization with the wireless broadcast advantage," in *Proc. Asilomar Conference on Signals, Systems and Computers*, Oct. 2008, pp. 2152–2158. DOI: 10.1109/ACSSC.2008.5074815.

A. SUPPLEMENTARY MATERIALS FOR CHAPTER 2

A.1 Proof of Proposition 2.2.2

Assumptions and notations: Consider any arbitrarily given rate vector $\vec{R} = (R_1, \cdots, R_K)$. Without loss of generality, also assume $\min_{k\in[K]} R_k > 0$ and $\min_{k\in[K]} p_k > 0$. Otherwise, we can simply remove the degenerate user and treat the K-session transmission problem as an equivalent $(K-1)$-session transmission problem. For notational simplicity, we denote $R_{\max} \triangleq \max_{k\in[K]} R_k$ and $p_{\min} \triangleq \min_{k\in[K]} p_k$. Also note that since we assume the arrival process being Poisson with arrival rate R_k, the variance of the arrival process is also R_k. We use $V_k \triangleq R_k$ to denote the variance of the i.i.d. arrival process and $V_{\max} \triangleq R_{\max}$ to denote the maximum of the variances of the K arrival processes.[1]

High-level description of the proof: For any stable rate $\vec{R}$ and the corresponding sequential network coding scheme, see Definition 2.2.3, and for any $\epsilon > 0$ and $0 < \delta < R_{\min}$, we will use this sequential coding scheme to design a block coding scheme with block length n and rate vector

$$\vec{R}^{\delta} \triangleq \vec{R} - \delta \cdot \mathbf{1} = (R_1 - \delta, \cdots, R_K - \delta)$$

such that the error rate of the block coding scheme is less than or equal to ϵ. This shows that for any stable $\vec{R}$ of the sequential-coding setting, the corresponding $\vec{R}^{\delta}$ is achievable in the block-coding setting. Since the capacity region is the closure of all achievable rates, this construction completes the proof of Proposition 2.2.2.

Detailed proof: We first prove the following lemma.

Lemma 4. *For any stable rate $\vec{R} = (R_1, \ldots, R_K)$ and the corresponding sequential network coding scheme, see Definition 2.2.3, there exist a constant $D < \infty$ and an infinite deterministic integer sequence $t_1 < t_2 < t_3 < \cdots$ such that*

$$\mathsf{E}\{|Q(t_{\mathrm{i}})|\} \leq D, \quad \forall \mathrm{i} \in [1, \infty), \tag{A.1}$$

[1]↑By using the variance notation $V_{\max}$, our proof also holds for any i.i.d. arrival processes with finite second moment $V_{\max} < \infty$.

where $|Q(t)|$ is the amount of memory usage at time t.

Proof. Since $\vec{R}$ is stable, we can define a finite number U_Q by

$$U_Q \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \mathsf{E}\left\{|Q(\tau)|\right\} < \infty. \tag{A.2}$$

For any finite but fixed $D > U_Q$, we will prove by contradiction that we can find a subsequence $t_1 < t_2 < \cdots$ satisfying Lemma 4. Suppose not. Then there exists $T \in \mathbb{N}$ such that $\mathsf{E}\{|Q(t)|\} > D$ for all $t \geq T$. Together with (A.2) we then have $U_Q \geq D$, which contradicts the initial choice of $D > U_Q$. The proof is thus complete. □

Using the D value and the sequence $\{t_{\mathrm{i}}\}$ that satisfy Lemma 4, we then define three positive integers n_2, n_1, and n by

$$n_2 \triangleq \left\lceil \max\left\{ \frac{8K}{p_{\min}} \ln\left(\frac{3K}{\epsilon}\right), \frac{6KD}{p_{\min}\cdot\epsilon} \right\} \right\rceil \tag{A.3}$$

$$n_1 \triangleq t_{\mathrm{i}_0} > \max\left\{ 2n_2\left(\frac{R_{\max}}{\delta} - 1\right), \frac{12KV_{\max}}{\epsilon\delta^2} \right\} \tag{A.4}$$

$$n \triangleq n_1 + n_2. \tag{A.5}$$

That is, we first find the n_2 value by (A.3). Then we choose n_1 value to be one of the entries of $\{t_{\mathrm{i}}\}$ that is strictly larger than the right-hand side of (A.4). Finally, we define $n = n_1 + n_2$, which will be used as the block coding length in our construction. In the following we describe how to construct a new block coding scheme of length n from the given sequential coding scheme.

Given any $(nR_1^\delta, \ldots, nR_K^\delta)$ packets that need to be transmitted via a block coding scheme, we first use these packets to "simulate" K i.i.d. Poisson packet arrival processes with arrival rates $(R_1, \cdots, R_K)$ for n_1 time slots. More specifically, in the beginning of transmission all nR_k^δ packets are marked as "not-yet-arrived". Then for each time slot t, we randomly generate the $A_k(t)$ value according to a Poisson distribution with parameter R_k, and we change the labels of $A_k(t)$ packets in the original nR_k^δ packets from "not-yet-arrived" to "arrived".

Recall that $M_k(n_1)$ denote the cumulative number of packet arrivals during the first n_1 time slots, which is a random variable with average $\mathsf{E}\{M_k(n_1)\} = n_1 R_k$. If the cumulative number of packet arrivals $M_k(n_1) < nR_k^\delta$, then it means that in the end of time n_1 some packets are still labeled as "not-yet-arrived". In this case we declare encoding/decoding failure and the block coding scheme terminates. If $M_k(n_1) \geq nR_k^\delta$, then the labels of all packets have eventually been changed to "arrived". For the last $M_k(n_1) - nR_k^\delta$ packets, i.e., after all nR_k^δ packets have been labeled "arrived", we can simply insert new all-zero packets (the dummy packets) into the system.

Since the original nR_k^δ packets now "arrive" at source s as a Poisson random process with rate R_k, we can apply the sequential coding scheme during the first n_1 time slots. At the end of time n_1, the total number of packets in the memory of the source is $|Q(n_1)|$. Then for the remaining n_2 time slots we "broadcast" the remaining packets in the queue $Q(n_1)$ to all K destinations. Namely, each destination d_k, $k \in [K]$, would like to know all the content of $Q(n_1)$ by the end of time $n = n_1 + n_2$ (during time $[n_1 + 1, n_1 + n_2]$). To achieve this task, we transmit the $|Q(n_1)|$ packets one-by-one to d_1 first, and then transmit all $|Q(n_1)|$ packets one-by-one to d_2, so on and so forth until all d_k receive all the $|Q(n_1)|$ packets.[2]

If source s fails to deliver all $|Q(n_1)|$ packets to all K destinations during time $[n_1 + 1, n_1 + n_2]$, we declare encoding/decoding failure and the block coding scheme terminates. Note that since the sequential coding scheme is decodable, see (2.9) and (2.10) in Section 2.2, after delivering all packets in $Q(n_1)$ to all K destinations, each d_k can then decode its desired nR_k^δ packets. The description of the block coding scheme is thus complete.

Analyzing the above block code, the decoding error can only follow from the following three events: **Error Type 1:** When simulating the Poisson arrival processes in the first n_1 time slots, the actual number of random arrivals of user-k is smaller than nR_k^δ. That is, some of the to-be-transmitted packets never "arrive" at the source and thus cannot be delivered by our new scheme. **Error Type 2:** $|Q(n_1)| > \frac{3D}{\epsilon}$. Namely, at the end of time n_1, there are too many packets in the queue at the source. **Error Type 3:** $|Q(n_1)| \leq \frac{3D}{\epsilon}$ but there exists

[2]↑Herein we use *uncoded broadcast* during time $[n_1 + 1, n_1 + n_2]$. We can also use *network-coded-broadcast*. Both will serve the same purpose in our proof and we choose the former for its simplicity.

a destination d_k such that not all $|Q(n_1)|$ packets are successfully delivered to d_k during the n_2 time slots. Applying the union bound to the error probability, we have

$$\begin{aligned}
&\Pr\left(\bigcup_{k\in[K]}\{\hat{\mathbf{X}}_k \neq \mathbf{X}_k\}\right) \\
&\leq \sum_{k\in[K]} \Pr\left(M_k(n_1) < nR_k^\delta\right) + \Pr\left(|Q(n_1)| > \frac{3D}{\epsilon}\right) + \\
&\sum_{k\in[K]} \Pr\left(|Q(n_1)| \leq \frac{3D}{\epsilon}, \text{not all } |Q(n_1)| \text{ delivered to } d_k\right). \qquad \text{(A.6)}
\end{aligned}$$

We now upper bound each of the three probability terms individually. We upper bound the first term by the Chebyshev inequality. Specifically, $M_k(n_1)$ is a Poisson random variable with mean n_1R_k and variance n_1V_k. Comparing the mean of $M_k(n_1)$ versus the original number of packets nR_k^δ, we have

$$\begin{aligned}
\mathsf{E}\{M_k(n_1)\} - nR_k^\delta &= n_1R_k - nR_k^\delta \\
&= n_1R_k - (n_1+n_2)(R_k-\delta) \\
&= n_1\delta + n_2\delta - n_2R_k \\
&\geq n_1\delta + n_2\delta - n_2R_{\max} \\
&> n_1\delta + n_2\delta - \left(\frac{n_1}{2} + n_2\right)\delta = \frac{n_1\delta}{2} > 0. \qquad \text{(A.7)}
\end{aligned}$$

where the inequality in (A.7) follows from the following argument. By (A.4) we have $n_1 > 2n_2(\frac{R_{\max}}{\delta} - 1)$, which implies $(\frac{n_1}{2} + n_2)\delta > n_2R_{\max}$ and thus (A.7). Then each summand of the first summation term in (A.6) is upper bounded by

$$\begin{aligned}
&\Pr\left(M_k(n_1) < nR_k^\delta\right) \\
&\leq \Pr\left(\left|\frac{M_k(n_1)}{n_1} - R_k\right| > \frac{\delta}{2}\right) \qquad \text{(A.8)} \\
&\leq \frac{V_k/n_1}{\delta^2/4} \qquad \text{(A.9)} \\
&\leq \frac{4V_{\max}}{n_1\delta^2} \leq \frac{\epsilon}{3K}. \qquad \text{(A.10)}
\end{aligned}$$

where (A.8) follows from (A.7); (A.9) follows from the Chebyshev inequality; and (A.10) follows from the definition of n_1 in (A.4). The summation in the first term is thus upper bounded by $\frac{\epsilon}{3}$.

The second term in (A.6) is bounded by Markov's inequality. That is,

$$\begin{aligned}
&\Pr\left(|Q(n_1)| > \frac{3D}{\epsilon}\right) \\
&\leq \frac{\epsilon}{3D}\mathsf{E}\left\{|Q(n_1)|\right\} \\
&\leq \frac{\epsilon D}{3D} = \frac{\epsilon}{3}. \qquad \text{(A.11)}
\end{aligned}$$

where (A.11) follows from the definition of choosing $n_1 = t_{\mathrm{i}_0}$ in (A.4), which ensures $\mathsf{E}\{|Q(n_1)|\} \leq D$.

The third term in (A.6) is bounded by the Chernoff bound. By (A.3) we have

$$\frac{3D}{\epsilon} \leq \frac{n_2 p_{\min}}{2K} \leq \frac{n_2 p_k}{2K}. \qquad \text{(A.12)}$$

For simplicity, we divide n_2 time slots into K sub-intervals. Each sub-interval has length n_2/K and is responsible for delivering $|Q(n_1)|$ packets to user d_k for some $k \in [K]$. If we denote Γ_k as the number of d_k's received packets during the $\frac{n_2}{K}$ time slots, then Γ_k is a Binomial random variable with parameter $(\frac{n_2}{K}, p_k)$ and the event "not all the $|Q(n_1)|$ packets are delivered successfully to d_k by the end of time n" is equivalent to $\Gamma_k < |Q(n_1)|$, i.e.,

$$\begin{aligned}
&\Pr\Bigg(|Q(n_1)| \leq \frac{3D}{\epsilon}, \\
&\qquad \text{not all } |Q(n_1)| \text{ packets are delivered to } d_k\Bigg) \\
\leq &\Pr\left(\Gamma_k < |Q(n_1)|,\ |Q(n_1)| \leq \frac{3D}{\epsilon}\right) \\
\leq &\Pr\left(\Gamma_k \leq \left(1 - \frac{1}{2}\right)\frac{n_2 p_k}{K}\right) \qquad \text{(A.13)} \\
\leq &\exp\left(-\frac{1}{4}\cdot\frac{n_2 p_k}{2K}\right) \qquad \text{(A.14)} \\
\leq &\exp\left(-\frac{n_2 p_{\min}}{8K}\right) \leq \frac{\epsilon}{3K}. \qquad \text{(A.15)}
\end{aligned}$$

where (A.13) follows from (A.12), (A.14) follows from the Chernoff bound, and (A.15) follows from the definition of n_2 in (A.3).

Hence (A.6), (A.10), (A.11), and (A.15) yield

$$\Pr\left(\bigcup_{k\in[K]}\{\hat{\mathbf{X}}_k \neq \mathbf{X}_k\}\right) \leq \epsilon. \tag{A.16}$$

This shows that the above block coding scheme achieves rate $\vec{R}^{\delta}$. The proof is thus complete.

A.2 On The Optimality of Proposition 2.3.1

In this appendix, we prove that Proposition 2.3.1 is optimal in many scenarios while being suboptimal in some other scenarios.

A.2.1 Proof of Corollary 1

Comparing Propositions 2.2.1 and 2.3.1, "the stability region matching the capacity region" is equivalent to the following statement:

> Given any rate $\vec{R}$ satisfying the outer bound (2.5) in Proposition 2.2.1, there exists non-negative variables $\{x_T : T \in 2^{[K]}\backslash\{\emptyset\}\}$ satisfying Conditions 1 and 2 in Proposition 2.3.1, provided we change the strict inequality in (2.17) to $\leq$.

In the sequel, we prove that the above statement holds for the following cases.

Case 1: $K = 3$. Consider any fixed rate (R_1, R_2, R_3) that is within the outer bound described in Proposition 2.2.1. For any three distinct indices $\mathrm{i}, \mathrm{j}, k \in \{1, 2, 3\}$ (no two being equal), define a function

$$\tau(\mathrm{i}, \mathrm{j}, k) \triangleq \frac{R_{\mathrm{i}}}{p_{\cup\{\mathrm{i},\mathrm{j},k\}}} + \frac{R_{\mathrm{j}}}{p_{\cup\{\mathrm{j},k\}}} + \frac{R_k}{p_k}, \tag{A.17}$$

Since (R_1, R_2, R_3) satisfies Proposition 2.2.1, we have $\max_{\mathrm{i},\mathrm{j},k} \tau(\mathrm{i}, \mathrm{j}, k) \leq 1$.

We now construct the variables $\{x_T : T \in 2^{[3]} \backslash \{\emptyset\}\}$ by

$$x_{\{2\}} = \frac{R_2}{p_{\cup\{1,2,3\}}}, \tag{A.18}$$

$$x_{\{1,2\}} = -\frac{R_1 + R_2}{p_{\cup\{1,2,3\}}} - \frac{R_3}{p_3} + \max\{\tau(1,2,3), \tau(2,1,3)\}, \tag{A.19}$$

$$\begin{aligned} x_{\{1,2,3\}} =& \frac{R_1 + R_2 + R_3}{p_{\cup\{1,2,3\}}} + \frac{R_1}{p_1} + \frac{R_2}{p_2} + \frac{R_3}{p_3} + \max_{\forall \mathrm{i,j},k} \tau(\mathrm{i},\mathrm{j},k) \\ & - \max\{\tau(1,2,3), \tau(2,1,3)\} \\ & - \max\{\tau(1,3,2), \tau(3,1,2)\} \\ & - \max\{\tau(3,2,1), \tau(2,3,1)\}. \end{aligned} \tag{A.20}$$

Variables $x_{\{1\}}$ (resp. $x_{\{3\}}$) is defined in a symmetric way as $x_{\{2\}}$ by swapping the user indices 2 and 1 (resp. 2 and 3) in (A.18). $x_{\{1,3\}}$ (resp. $x_{\{2,3\}}$) is defined in a symmetric way as $x_{\{1,2\}}$ by swapping the user indices 2 and 3 (resp. 1 and 3) in (A.19).

We now prove that the x_T variables in the above construction are non-negative. It is clear from (A.18) that $x_{\{2\}} \geq 0$. By symmetry we also have $x_{\{1\}} \geq 0$ and $x_{\{3\}} \geq 0$.

We also have

$$x_{\{1,2\}} \geq -\frac{R_1 + R_2}{p_{\cup\{1,2,3\}}} - \frac{R_3}{p_3} + \tau(2,1,3) \tag{A.21}$$

$$= R_1 \left(\frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) \geq 0 \tag{A.22}$$

where (A.21) follows from (A.19) and the equality of (A.22) follows from (A.17). Eq. (A.22) is non-negative since by definition $p_{\cup\{1,3\}} \leq p_{\cup\{1,2,3\}}$. By symmetry, $x_{\{1,3\}}$ and $x_{\{2,3\}}$ are also non-negative.

To show that $x_{\{1,2,3\}} \geq 0$, without loss of generality, we assume $\max_{\mathrm{i,j},k} \tau(\mathrm{i},\mathrm{j},k) = \tau(1,2,3)$ and discuss the following cases.

(i) Consider $\tau(1,3,2) \geq \tau(3,1,2)$ and $\tau(3,2,1) \geq \tau(2,3,1)$. Since $\tau(1,2,3)$ is assumed to be the largest, the inequality $\tau(1,2,3) \geq \tau(1,3,2)$ also yields

$$R_3 \left(\frac{1}{p_3} - \frac{1}{p_{\cup\{2,3\}}} \right) \geq R_2 \left(\frac{1}{p_2} - \frac{1}{p_{\cup\{2,3\}}} \right).$$

Therefore (A.20) becomes

$$\begin{aligned} x_{\{1,2,3\}} &= \frac{R_3}{p_3} - \frac{R_3}{p_{\cup\{2,3\}}} - \frac{R_2}{p_{\cup\{1,2\}}} + \frac{R_2}{p_{\cup\{1,2,3\}}} \\ &\geq R_2 \left(\frac{1}{p_2} - \frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,2\}}} + \frac{1}{p_{\cup\{1,2,3\}}} \right) \geq 0, \end{aligned}$$

where the last nonnegative inequality follows from [5, Lemma 5].

(ii) Consider $\tau(1,3,2) \geq \tau(3,1,2)$ and $\tau(3,2,1) < \tau(2,3,1)$. Eq. (A.20) becomes

$$x_{\{1,2,3\}} = R_3 \left(\frac{1}{p_3} - \frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,3\}}} + \frac{1}{p_{\cup\{1,2,3\}}} \right) \geq 0.$$

(iii) Consider $\tau(1,3,2) < \tau(3,1,2)$ and $\tau(3,2,1) \geq \tau(2,3,1)$. Since $\tau(1,2,3)$ is the largest, $\tau(1,2,3) \geq \tau(3,1,2)$ and it implies

$$\frac{R_1}{p_{\cup\{1,2,3\}}} + \frac{R_2}{p_{\cup\{2,3\}}} + \frac{R_3}{p_3} \geq \frac{R_3}{p_{\cup\{1,2,3\}}} + \frac{R_1}{p_{\cup\{1,2\}}} + \frac{R_2}{p_2}.$$

Therefore (A.20) becomes

$$\begin{aligned} x_{\{1,2,3\}} =& \frac{R_1 + R_2 - R_3}{p_{\cup\{1,2,3\}}} + \frac{R_3}{p_3} - \frac{R_1}{p_{\cup\{1,2\}}} - \frac{R_2}{p_{\cup\{1,2\}}} \\ \geq& R_2 \left(\frac{1}{p_2} - \frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,2\}}} + \frac{1}{p_{\cup\{1,2,3\}}} \right) \geq 0. \end{aligned}$$

(iv) Consider $\tau(1,3,2) < \tau(3,1,2)$ and $\tau(3,2,1) < \tau(2,3,1)$, which implies

$$\begin{aligned} R_1 \left(\frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) &> R_3 \left(\frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right), \\ R_3 \left(\frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) &> R_2 \left(\frac{1}{p_{\cup\{1,2\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right). \end{aligned}$$

The product of the above two inequalities implies

$$R_1 \left(\frac{1}{p_{\cup\{1,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right) > R_2 \left(\frac{1}{p_{\cup\{2,3\}}} - \frac{1}{p_{\cup\{1,2,3\}}} \right).$$

However since $\tau(1,2,3) \geq \tau(2,1,3)$, we also have

$$R_2\left(\frac{1}{p_{\cup\{2,3\}}}-\frac{1}{p_{\cup\{1,2,3\}}}\right) \geq R_1\left(\frac{1}{p_{\cup\{1,3\}}}-\frac{1}{p_{\cup\{1,2,3\}}}\right).$$

This contradiction shows that this case is impossible.

Thus far we have proven the non-negativity of the $\{x_T\}$ variables. We now prove that those $\{x_T\}$ satisfy (2.17) and (2.18). By the definitions in (A.18) to (A.20) one can directly verify that

$$\sum_{\forall T\in 2^{[3]}\setminus\{\emptyset\}} x_T = \max_{\mathrm{i,j},k} \tau(\mathrm{i},\mathrm{j},k) \tag{A.23}$$

which is no larger than one, see the discussion right after (A.17). The time-sharing condition (2.17) thus holds.

To verify (2.18), we consider only the case of $k=2$. The cases of $k=1,3$ follow by symmetry. There are five possible proper cuts in Condition 2, namely $\mathcal{C}_{\mathrm{i}}^{(2)}$, $\mathrm{i}\in\{1,2,3,4,5\}$ in (2.12) to (2.16). In the following, we examine (2.18) for each of these five cuts separately, also see (2.20) to (2.24).

For $\mathcal{C}_1^{(2)}=\{\mathrm{vn}_{\mathrm{grnd}}^{(2)}\}$, inequality (2.20) becomes

$$\begin{aligned}
&\left(x_{\{2\}}+x_{\{1,2\}}+x_{\{2,3\}}+x_{\{1,2,3\}}\right)\cdot p_2\\
=&\left(\frac{R_2}{p_2}+\max_{\forall \mathrm{i,j},k}\tau(\mathrm{i},\mathrm{j},k)-\max\{\tau(1,3,2),\tau(3,1,2)\}\right)\cdot p_2\\
\geq& R_2.
\end{aligned} \tag{A.24}$$

For $\mathcal{C}_2^{(2)}=\{\mathrm{vn}_{\mathrm{grnd}}^{(2)},\mathrm{vn}_{\{1,3\}}^{(2)}\}$, inequality (2.21) becomes

$$\begin{aligned}
&x_{\{2\}}\left(p_2+p_{\{1,3\}\bar{2}}\right)+x_{\{1,2\}}\left(p_2+p_{3\bar{2}}\right)+x_{\{2,3\}}\left(p_2+p_{1\bar{2}}\right)\\
&=x_{\{2\}}\cdot\left(p_{\cup\{1,2\}}+p_{\cup\{2,3\}}-p_{\cup\{1,2,3\}}\right)\\
&\quad+x_{\{1,2\}}\cdot p_{\cup\{2,3\}}+x_{\{2,3\}}\cdot p_{\cup\{1,2\}}\\
&=\left(x_{\{2\}}+x_{\{1,2\}}\right)\cdot p_{\cup\{2,3\}}+\left(x_{\{2\}}+x_{\{2,3\}}\right)\cdot p_{\cup\{1,2\}}-R_2\\
&\geq 2R_2-R_2=R_2.
\end{aligned} \tag{A.25}$$

For $\mathcal{C}_3^{(2)} = \{\mathrm{vn}_{\mathrm{grnd}}^{(2)}, \mathrm{vn}_{\{1\}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\}$, inequality (2.22) becomes

$$\begin{aligned}
&x_{\{2\}}\left(p_2 + p_{1\overline{\{2,3\}}} + p_{\{1,3\}\overline{2}}\right) + x_{\{2,3\}}\left(p_2 + p_{1\overline{2}}\right) \\
&= \left(x_{\{2\}} + x_{\{2,3\}}\right) \cdot p_{\cup\{1,2\}} \\
&\geq \left(\tau(3,2,1) - \frac{R_3}{p_{\cup\{1,2,3\}}} - \frac{R_1}{p_1}\right) \cdot p_{\cup\{1,2\}} = R_2. \qquad \text{(A.26)}
\end{aligned}$$

The case of $\mathcal{C}_4^{(2)} = \{\mathrm{vn}_{\mathrm{grnd}}^{(2)}, \mathrm{vn}_{\{3\}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\}$ is symmetric to the case of $\mathcal{C}_3^{(2)} = \{\mathrm{vn}_{\mathrm{grnd}}^{(2)}, \mathrm{vn}_{\{1\}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\}$. For $\mathcal{C}_5^{(2)} = \{\mathrm{vn}_{\mathrm{grnd}}^{(2)}, \mathrm{vn}_{\{1\}}^{(2)}, \mathrm{vn}_{\{3\}}^{(2)}, \mathrm{vn}_{\{1,3\}}^{(2)}\}$, inequality (2.24) becomes

$$x_{\{2\}} \cdot \left(p_2 + p_{1\overline{\{2,3\}}} + p_{3\overline{\{1,2\}}} + p_{\{1,3\}\overline{2}}\right) = R_2, \qquad \text{(A.27)}$$

where (A.27) follows from (A.18). Our construction of $\{x_T\}$ thus satisfies (2.18). The proof of Case 1 is complete.

Case 2: The channel is spatially independent and the rate-vector is one-sided fair. Without loss of generality, we assume the marginal success probability p_k are listed from the smallest to the largest. That is, $0 < p_1 \leq p_2 \leq \cdots \leq p_K$. A rate vector $\vec{R}$ is one-sided fair if $R_{\mathrm{i}} \cdot (1 - p_{\mathrm{i}}) \geq R_{\mathrm{j}} \cdot (1 - p_{\mathrm{j}})$ for all $\mathrm{i} < \mathrm{j}$. Detailed discussion on one-sided fairness can be found in [5].

Consider any spatially independent channel and any one-sided fair rate vector $(R_1, \cdots, R_K)$ satisfying the outer bound (2.5). For any k and $S \in 2^{[K]\setminus\{k\}}$, define

$$u_S^{(k)} \triangleq R_k \cdot \left(\sum_{\forall S_1 : ([K]\setminus S) \subseteq S_1 \subseteq [K]} \frac{(-1)^{|S_1| - (K - |S|)}}{p_{\cup S_1}}\right). \qquad \text{(A.28)}$$

We now construct the desired variables $\{x_T\}$ by choosing $x_T = u_{T\setminus k^*(T)}^{(k^*(T))}$, where $k^*(T) = \min\{\mathrm{i} : \mathrm{i} \in T\}$.

In the following we prove that the above $\{x_T\}$ are non-negative and satisfy Condition 1 in (2.17) and Condition 2 (2.18) simultaneously. Specifically, [5] proves that $u_S^{(k)}$ is non-negative for all $k \in [K]$ and $S \in 2^{[K]\setminus\{k\}}$. Since $u_{T\setminus k^*(T)}^{(k^*(T))} \geq 0$, we thus have $x_T \geq 0$ for all T.

To prove Condition 1 in (2.17), we have

$$\sum_{T\in 2^{[K]}\setminus\{\emptyset\}} u^{(k^*(T))}_{T\setminus k^*(T)} = \sum_{k\in[K]}\left(\sum_{S\subseteq\{k+1,\cdots,K\}} u^{(k)}_S\right) \tag{A.29}$$

$$= \sum_{k\in[K]} \frac{R_k}{p_{\cup[k]}} \le 1, \tag{A.30}$$

where the (A.30) follows the Möbius inversion [64] of partially ordered sets with two functions $u^{(k)}_S$ in (A.28) and $v^{(k)}_T \triangleq \frac{R_k}{p_{\cup T}}$.

To prove Condition 2 in (2.18), we use the result of Lemma 9 in Appendix A.6 and show that there exist nonnegative variables $\vec{y}$ defined in (A.54) such that (A.55), (A.56), and (A.57) are satisfied. Given the variables $\{x_T\}$ above, let the $\vec{y}$ variables for any fixed $k \in [K]$ be

$$y_{k;S_I\to d_k} = u^{(k)}_{S_I}\cdot p_k, \quad \forall S_I \in 2^{[K]\setminus\{k\}}, \tag{A.31}$$

$$y^{[S_X]}_{k;S_I\to S_O} = \begin{cases} u^{(k)}_{S_I}\cdot p_{S_X\overline{[K]\setminus(S_I\cup S_X)}}, & S_O = S_I\cup S_X, \\ 0, & \text{otherwise}, \end{cases}$$

$$\forall S_I \in 2^{[K]\setminus\{k\}}, S_X \in 2^{[K]\setminus(S_I\cup\{k\})}. \tag{A.32}$$

Since $u^{(k)}_S$ are nonnegative, we have $\{y_{k;S_I\to d_k}\}$ and $\{y^{[S_X]}_{k;S_I\to S_O}\}$ are also nonnegative. Recall that we define $x_T = u^{(k^*(T))}_{T\setminus k^*(T)}$. Since $\vec{R}$ is one-sided fair, by [5, Lemma 5] we have $u^{(k)}_S \le x_{S\cup\{k\}}$ for all k and S. This directly implies that $\vec{y}$ satisfies (A.56) and (A.57). To show (A.55) holds, for a given $k \in [K]$ and $S \in 2^{[K]\setminus\{k\}}$ the right-hand side of (A.55) is

$$u^{(k)}_S\cdot\left(p_k + \sum_{S_X\in 2^{[K]\setminus(S\cup\{k\})}} p_{S_X\overline{[K]\setminus(S\cup S_X)}}\right) = u^{(k)}_S\cdot p_{\overline{S}}. \tag{A.33}$$

We then consider the left-hand side of (A.55). If $S = \emptyset$, we have $u^{(k)}_\emptyset = \frac{R_k}{p_{\cup[K]}} = \frac{R_k}{1-p_{\overline{[K]}}}$ and $p_{\overline{\emptyset}} = 1$. The left-hand side of (A.55) then becomes

$$R_k + u^{(k)}_\emptyset\cdot p_{\overline{[K]}} = \frac{R_k}{1-p_{\overline{[K]}}} = u^{(k)}_\emptyset\cdot p_{\overline{\emptyset}}. \tag{A.34}$$

On the other hand, if $S \in 2^{[K]\setminus\{k\}}\setminus\{\emptyset\}$, the left-hand side of (A.55) is given by

$$\sum_{S_I \in 2^{[S]}} u_{S_I}^{(k)} \cdot p_{(S\setminus S_I)\overline{[K]\setminus S}} = u_S^{(k)} \cdot p_{\overline{S}}. \tag{A.35}$$

Therefore, (A.55) always holds with equality. The proof of Case 2 is complete.

Case 3: Symmetric channels. A 1-to-K broadcast PEC is said to be symmetric if the success probability $p_{S_1\overline{[K]\setminus S_1}} = p_{S_2\overline{[K]\setminus S_2}}$ for any $S_1, S_2 \in 2^{[K]}$ satisfying $|S_1| = |S_2|$. Consider any symmetric channel and any rate vector $\vec{R} = (R_1, \ldots, R_K)$ satisfying the outer bound (2.5). Without loss of generality, we also assume $R_1 \geq R_2 \geq \cdots \geq R_K$, which is always possible after relabeling the destinations. We then choose non-negative $x_T = u_{T\setminus k^*(T)}^{(k^*(T))}$, where $k^*(T) = \min\{\mathrm{i} : \mathrm{i} \in T\}$. Since the choice of $\{x_T\}$ is the same as that in Case 2, $\{x_T\}$ satisfying Condition 1 as in (A.30).

To show that such $\{x_T\}$ also satisfies Condition 2, we choose the same non-negative $\vec{y}$ as in (A.31) and (A.32) and we will show that $\vec{y}$ also satisfies (A.55), (A.56), and (A.57) in Lemma 9. The reason is as follows. Due to the symmetric channel and the sorted R_k in descending order, it is obvious that $x_T = u_{T\setminus k^*(T)}^{(k^*(T))} = \max_{k\in T}\{u_{T\setminus\{k\}}^{(k)}\}$. Then by the same reasons as discussed in Case 2, inequalities (A.56) and (A.57) is satisfied. Then following the same reason of (A.33), (A.34), and (A.35), $\vec{y}$ also satisfies (A.55). The proof of Case 3 is complete.

A.2.2 Proof of Example 1

Suppose the channel is spatially independent with marginal success probability being $\vec{p} = (p_1, p_2, p_3, p_4) = (\frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{4}{7})$ and rate vector being $\vec{R} = (R_1, R_2, R_3, R_4) = (\frac{96}{1193}, \frac{672}{5\cdot 1193}, \frac{288}{1193}, \frac{1952}{7\cdot 1193})$.

One can verify that all $4! = 24$ inequalities of the outer bound in Proposition 2.2.1 are satisfied. Specifically, the following four inequalities are satisfied with equality

$$\begin{cases} \frac{R_1}{p_{\cup\{1,2,3,4\}}} + \frac{R_4}{p_{\cup\{2,3,4\}}} + \frac{R_2}{p_{\cup\{2,3\}}} + \frac{R_3}{p_3} = 1, \\ \frac{R_4}{p_{\cup\{1,2,3,4\}}} + \frac{R_1}{p_{\cup\{1,2,3\}}} + \frac{R_2}{p_{\cup\{2,3\}}} + \frac{R_3}{p_3} = 1, \\ \frac{R_4}{p_{\cup\{1,2,3,4\}}} + \frac{R_2}{p_{\cup\{1,2,3\}}} + \frac{R_1}{p_{\cup\{1,3\}}} + \frac{R_3}{p_3} = 1, \\ \frac{R_4}{p_{\cup\{1,2,3,4\}}} + \frac{R_2}{p_{\cup\{1,2,3\}}} + \frac{R_3}{p_{\cup\{1,3\}}} + \frac{R_1}{p_1} = 1, \end{cases} \tag{A.36}$$

and the remaining 20 inequalities are satisfied with strict inequality.

We now turn our attention to Proposition 2.3.1, the stability region. Obviously, this channel model is asymmetric. The rate vector is also not one-sided fair since $R_2 \cdot (1-p_2) = \frac{2016}{25 \cdot 1193} < \frac{144}{1193} = R_3 \cdot (1-p_3)$, also see the discussion in the end of Appendix A.2.1. We choose the x_T variables as follows.

$$\begin{aligned} &x_{\{1\}} = \frac{105}{1193},\ x_{\{2\}} = \frac{147}{1193},\ x_{\{3\}} = \frac{315}{1193},\ x_{\{4\}} = \frac{305}{1193}, \\ &x_{\{1,2\}} = \frac{441}{61 \cdot 1193},\ x_{\{1,3\}} = \frac{945}{61 \cdot 1193},\ x_{\{1,4\}} = \frac{15}{1193}, \\ &x_{\{2,3\}} = \frac{21}{1193},\ x_{\{2,4\}} = \frac{21}{1193},\ x_{\{3,4\}} = \frac{45}{61 \cdot 1193}, \\ &x_{\{1,2,3\}} = \frac{10101}{11 \cdot 61 \cdot 1193}, x_{\{1,2,4\}} = \frac{1023}{61 \cdot 1193}, x_{\{2,3,4\}} = \frac{51}{1193}, \\ &x_{\{1,3,4\}} = \frac{15345}{7 \cdot 61 \cdot 1193},\ x_{\{1,2,3,4\}} = \frac{364101}{7 \cdot 11 \cdot 61 \cdot 1193}. \end{aligned}$$

Clearly these x_T are non-negative and their total sum is $\frac{5603521}{5603521} = 1$. The cut condition in (2.18) can be verified by simple computer programs. The proof of Example 1 is thus complete.

A.2.3 An Example of Unachievable Rates for $K = 4$

Consider the case of $K = 4$, rate vector $\vec{R} = (0.173284, 0.146994, 0.174068, 0.127000)$. We introduce the following equivalent notation for the PEC parameters $\{p_{S\overline{[K]\setminus S}}\}$. That is,

$$p_{b_1b_2b_3b_4} = p_{\{\mathrm{i}:b_\mathrm{i}=1\}\overline{\{\mathrm{j}:b_\mathrm{j}=0\}}}. \tag{A.37}$$

That is, each b_i indicates whether the i-th user has received the packet. For example, $p_{1011} = p_{\{1,3,4\}\overline{2}}$ is the probability that d_1, d_3, and d_4 receive the packet but not d_2. Using this new notation, we set the channel parameters to be

$$\begin{aligned}
&p_{0000} = .1085,\ p_{0001} = .0132,\ p_{0010} = .1072,\ p_{0011} = .1040,\\
&p_{0100} = .0846,\ p_{0101} = .0950,\ p_{0110} = .0563,\ p_{0111} = .0059,\\
&p_{1000} = .1043,\ p_{1001} = .1099,\ p_{1010} = .0065,\ p_{1011} = .0344,\\
&p_{1100} = .0023,\ p_{1101} = .0693,\ p_{1110} = .0033,\ p_{1111} = .0953,
\end{aligned}$$

such that $\sum_{b_1,b_2,b_3,b_4\in\{0,1\}} p_{b_1b_2b_3b_4} = 1$.

A simple computer program can be used to verify that these choices of $\vec{R}$ and $p_{b_1b_2b_3b_4}$ satisfy all 4! inequalities of the outer bound in Proposition 2.2.1. If we define $\vec{R}_\alpha = \alpha \cdot \vec{R}$, we can use an LP solver to find the largest possible α such that $\vec{R}_\alpha$ satisfies the inner bound in Proposition 2.3.1. In this example, the largest $\alpha = 99.86\%$. This shows that the inner bound in Proposition 2.3.1 does not always meet the outer bound in Proposition 2.2.1.

A.3 Proof of Lemmas 1 and 2

We first prove Lemma 1. Assume any arbitrary but fixed virtual node scheduling frequency $\{x_T : T \in 2^{[K]}\setminus\{\emptyset\}\}$. The cut value of a node-cut $\mathcal{C}^{(k)}$ (not necessarily a proper cut) is the summation of the capacity of edges of the following two types. Type 1: Edges from a virtual node $\mathrm{vn}_S^{(k)} \notin \mathcal{C}^{(k)}$ to an auxiliary node $S_X \in 2^{[K]\setminus(S\cup\{k\})}$ that has an outgoing infinite-capacity edge ending in a $\mathrm{vn}_{\tilde{S}}^{(k)} \in \mathcal{C}^{(k)}$; Type 2: Edges from a virtual node $\mathrm{vn}_S^{(k)} \notin \mathcal{C}^{(k)}$ to the sink node $\mathrm{vn}_{\mathrm{grnd}}^{(k)}$.

For any given $\text{vn}_S^{(k)} \notin \mathcal{C}^{(k)}$, the total edge capacity of Type-1 edges are:

$$x_{S\cup\{k\}} \left(\sum_{\substack{\forall S_X: \\ S_X \in \mathcal{F}(k,S,\mathcal{C}^{(k)})}} p_{S_X \overline{[K]\setminus(S_X \cup S)}} \right) \tag{A.38}$$

where $p_{S_X \overline{[K]\setminus(S_X\cup S)}}$ is the edge capacity entering auxiliary node S_X; and $\mathcal{F}(k, S, \mathcal{C}^{(k)})$ defined in (2.19) consists of all auxiliary nodes $\{S_X \in 2^{[K]\setminus(S\cup\{k\})}\}$ that have an infinite-capacity edge ending in some $\text{vn}_{\tilde{S}}^{(k)} \in \mathcal{C}^{(k)}$.

For any given $\text{vn}_S^{(k)} \notin \mathcal{C}^{(k)}$, there is only one Type-2 edge and its capacity is

$$x_{S\cup\{k\}} p_k. \tag{A.39}$$

By adding (A.38) and (A.39) together and then by summing over all $\text{vn}_S^{(k)} \notin \mathcal{C}^{(k)}$, the left-hand side of (2.18) computes the cut value of $\mathcal{C}^{(k)}$. The proof of Lemma 1 is complete.

We now prove Lemma 2. Since any proper-cut is also a cut, [Statement 1] implies [Statement 2]. We now prove that [Statement 2] implies [Statement 1].

Suppose [Statement 2] holds. For any virtual node $\text{vn}_S^{(k)}$, define $\mathcal{V}(\text{vn}_S^{(k)}) \triangleq \{\text{vn}_S^{(k)} : S \supseteq S\}$. By definition, we always have $\text{vn}_S^{(k)} \in \mathcal{V}(\text{vn}_S^{(k)})$. Then for any arbitrary cut $\partial^{(k)}$, define $\mathcal{C}^{(k)} \triangleq \left(\bigcup_{\text{vn}_S^{(k)} \in \partial^{(k)}} \mathcal{V}(\text{vn}_S^{(k)})\right) \cup \text{vn}_{\text{grnd}}^{(k)}$. It is clear that $\partial^{(k)} \subseteq \mathcal{C}^{(k)}$ and it is also true by Definition 2.3.3 that the above $\mathcal{C}^{(k)}$ is a proper cut.

From the definition of $\mathcal{F}$ in (2.19), it is clear that $\mathcal{F}(k, S, \partial^{(k)}) \subseteq \mathcal{F}(k, S, \mathcal{C}^{(k)})$ since $\partial^{(k)} \subseteq \mathcal{C}^{(k)}$. Furthermore, by the construction of $\mathcal{C}^{(k)}$ any $\text{vn}_{\tilde{S}}^{(k)} \in \mathcal{C}^{(k)}$ corresponds to a $\text{vn}_S^{(k)} \in \partial^{(k)}$ satisfying $S \subseteq \tilde{S}$. Therefore (2.19) implies that we actually have $\mathcal{F}(k, S, \partial^{(k)}) = \mathcal{F}(k, S, \mathcal{C}^{(k)})$.

If we use $\mathrm{cv}(\partial^{(k)})$ and $\mathrm{cv}(\mathcal{C}^{(k)})$ to denote the cut values of $\partial^{(k)}$ and $\mathcal{C}^{(k)}$, respectively, we then have

$$\mathrm{cv}\left(\partial^{(k)}\right) = \sum_{\substack{S:S\in 2^{[K]\setminus\{k\}},\\ \mathrm{vn}_S^{(k)}\notin\partial^{(k)}}} x_{S\cup\{k\}}\cdot\left(p_k + \sum_{\substack{\forall S_X:\\ S_X\in\mathcal{F}(k,S,\partial^{(k)})}} p_{S_X\overline{[K]\setminus(S_X\cup S)}}\right)$$

$$\geq \sum_{\substack{S:S\in 2^{[K]\setminus\{k\}},\\ \mathrm{vn}_S^{(k)}\notin\mathcal{C}^{(k)}}} x_{S\cup\{k\}}\cdot\left(p_k + \sum_{\substack{\forall S_X:\\ S_X\in\mathcal{F}(k,S,\partial^{(k)})}} p_{S_X\overline{[K]\setminus(S_X\cup S)}}\right) \tag{A.40}$$

$$= \sum_{\substack{S:S\in 2^{[K]\setminus\{k\}},\\ \mathrm{vn}_S^{(k)}\notin\mathcal{C}^{(k)}}} x_{S\cup\{k\}}\cdot\left(p_k + \sum_{\substack{\forall S_X:\\ S_X\in\mathcal{F}(k,S,\mathcal{C}^{(k)})}} p_{S_X\overline{[K]\setminus(S_X\cup S)}}\right) = \mathrm{cv}\left(\mathcal{C}^{(k)}\right) \tag{A.41}$$

$$\geq R_k \tag{A.42}$$

where (A.40) follows from changing the summation over $\mathrm{vn}_S^{(k)}\notin\partial^{(k)}$ to a smaller set $\mathrm{vn}_S^{(k)}\notin\mathcal{C}^{(k)}$; (A.41) follows from the fact that $\mathcal{F}(k,S,\partial^{(k)}) = \mathcal{F}(k,S,\mathcal{C}^{(k)})$; and (A.42) follows from [Statement 2]. Since $\mathrm{cv}(\partial^{(k)})\geq R_k$ is now proven for any arbitrary $\partial^{(k)}$, [Statement 1] holds. The proof is thus complete.

A.4 A Simple Schwartz-Zippel Lemma

Lemma 5. *Consider any integer M and any finite field $\mathsf{GF}(q)$ satisfying $q\geq M\geq 1$. For any M vectors $\mathbf{b}_1,\ldots,\mathbf{b}_M$ and any M matrices $\mathbf{B}^{(1)},\ldots,\mathbf{B}^{(M)}$, if $\mathbf{b}_m$ is linearly independent of the rows of $\mathbf{B}^{(m)}$ for all $1\leq m\leq M$, then there exists $\{\beta_m\in\mathsf{GF}(q):m\in[M]\}$ such that the sum $\sum_{m=1}^{M}\beta_m\cdot\mathbf{b}_m$ is linearly independent of the rows of $\mathbf{B}^{(m)}$ for all $m=1$ to M.*

Proof. Lemma 5 holds trivially if $M=1$. In the following proof we assume $M\geq 2$. For any matrix $\mathbf{B}$ and any row vector $\mathbf{v}$, we use $\mathbf{v}\in\langle\mathbf{B}\rangle$ to denote that $\mathbf{v}$ belongs to the row space of $\mathbf{B}$. For any $m\in[M]$, let

$$\mathcal{D}_m \triangleq \left\{(\beta_1,\cdots,\beta_M)\in\left(\mathsf{GF}(q)\right)^M : \left(\sum_{\mathrm{i}=1}^{M}\beta_{\mathrm{i}}\mathbf{b}_{\mathrm{i}}\right)\in\langle\mathbf{B}^{(m)}\rangle\right\}.$$

Namely, $\mathcal{D}_m$ contains the $(\beta_1, \cdots, \beta_M)$ coefficients such that the resulting vector $\left(\sum_{\mathrm{i}=1}^{M} \beta_{\mathrm{i}} \mathbf{b}_{\mathrm{i}}\right)$ belongs to the row space of $\mathbf{B}^{(m)}$. Clearly, any coefficient vector $(\beta_1, \cdots, \beta_M)$ inside $(\mathsf{GF}(q))^M \setminus \bigcup_{m=1}^{M} \mathcal{D}_m$ satisfies Lemma 5. As a result, we only need to prove that the set $(\mathsf{GF}(q))^M \setminus \bigcup_{m=1}^{M} \mathcal{D}_m$ is non-empty, or, equivalently, $\left|\bigcup_{m=1}^{M} \mathcal{D}_m\right| < q^M$.

Fix any m value and fix any $(\beta_1, \ldots, \beta_{m-1}, \beta_{m+1}, \ldots, \beta_M)$ values, we claim that there exists at most one $\beta_m \in \mathsf{GF}(q)$ such that the combined vector $(\beta_1, \ldots, \beta_M)$ is in $\mathcal{D}_m$. We prove this claim by contradiction. Suppose there are two distinct coefficients β_m and β_m such that

$$\mathbf{w} \triangleq \left(\beta_m \cdot \mathbf{b}_m + \sum_{k \in [M], k \neq m} \beta_k \cdot \mathbf{b}_k\right) \in \langle \mathbf{B}^{(m)} \rangle$$
$$\text{and } \mathbf{w} \triangleq \left(\beta_m \cdot \mathbf{b}_m + \sum_{k \in [M], k \neq m} \beta_k \cdot \mathbf{b}_k\right) \in \langle \mathbf{B}^{(m)} \rangle.$$

Then the difference $\mathbf{w} - \mathbf{w} = (\beta_m - \beta_m) \cdot \mathbf{b}_m$ must also be in $\langle \mathbf{B}^{(m)} \rangle$. However recall that Lemma 5 assumes $\mathbf{b}_m$ is linearly independent of the rows of $\mathbf{B}^{(m)}$. We thus have $\beta_m - \beta_m = 0$, which contradicts to the assumption that β_m and β_m are distinct.

Since for any $M-1$ coefficients $(\beta_1, \ldots, \beta_{m-1}, \beta_{m+1}, \ldots, \beta_M)$, there is at most one β_m satisfying $(\beta_1, \ldots, \beta_M) \in \mathcal{D}_m$, we must have $|\mathcal{D}_m| \leq q^{M-1}$. Also note that the zero vector $(\beta_1, \ldots, \beta_M) = (0, 0, \ldots, 0) \in \mathcal{D}_m$ for all m. By a simple union bound argument, $\left|\bigcup_{m=1}^{M} \mathcal{D}_m\right| \leq M \cdot q^{M-1} - (M-1) < q^M$, where the last inequality holds strictly as long as $q \geq M \geq 2$. The proof is complete.

□

A.5 Proof of Proposition 2.4.1

We now analyze the 5-component sequential coding scheme described in Section 2.4. Since some quantities like the $2K$ matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ evolve over time, we append (t) to indicate the quantities as a function of time t when necessary.

Recall that $\mathbf{X}_k(t)$ denotes the set of the flow-k packets that have already arrived at source s by the end of time t and the total number is denoted by $M_k(t) = |\mathbf{X}_k(t)|$. At

each time t, the linearly coded packets $Y(t)$ and queuing entries $W_{(k,\mathrm{j}_k)}(t) \in Q_S^{(k)}(t)$ for all $S \in 2^{[K]\setminus\{k\}}$ are linear combination of all the arrived packets $(\mathbf{X}_1(t), \cdots \mathbf{X}_K(t))$. Therefore the corresponding global encoding kernel will be an $M(t)$-dimensional row vector in $(\mathsf{GF}(q))^{M(t)}$, where $M(t) \triangleq \sum_{k=1}^{K} M_k(t)$ is the total number of arrived packets.

For simplicity, we use $\mathbf{u}_{(k,\mathrm{j})}(t) = \mathsf{row}_{(k,\mathrm{j})}(\mathbf{U}^{(k)}(t))$ to denote the row corresponding to queued entry $W_{(k,\mathrm{j})}(t)$. Therefore, we have $W_{(k,\mathrm{j})}(t) = \mathbf{u}_{(k,\mathrm{j})}(t) \cdot \vec{\mathbf{X}}_{\mathsf{arr}}$ where $\vec{\mathbf{X}}_{\mathsf{arr}}$ is a column vector that consists of all the packets that have arrived according to its arrival order. For ease of notation, we sometimes abuse the notation slightly by writing $W_{(k,\mathrm{j})}(t) = \mathbf{u}_{(k,\mathrm{j})}(t) \cdot \mathbf{X}(t)$ where $\mathbf{X}(t) \triangleq \{\mathbf{X}_1(t), \ldots, \mathbf{X}_K(t)\}$ is the collection of all the arrived packets. In this notation, we implicitly assume that the set $\mathbf{X}(t)$ is arranged accordingly to the arrival order, the same way as the vector $\mathbf{u}_{(k,\mathrm{j})}(t)$ is arranged.

For any k and $1 \leq \mathrm{i} \leq M_k(t)$, we use $\mathbf{e}_{(k,\mathrm{i})}$ to denote an $M(t)$-dimensional row vector with the coordinate/column corresponding to the packet $X_{k,\mathrm{i}}$ being one and all the other coordinates being zero. Define the matrix $\mathbf{I}^{(k)}(t)$ as the vertical concatenation of rows $\mathbf{e}_{(k,1)}, \ldots, \mathbf{e}_{(k,M_k(t))}$.

For example, suppose five packets have arrived by time t. We sort them from the oldest to the youngest (according to the order of arrivals) and they are $X_{1,1}$, $X_{2,1}$, $X_{1,2}$, $X_{3,1}$, $X_{2,2}$. Namely, the first packet of the user-2 stream is the second oldest among the five arrived packets and the second packet of the user-2 stream is the youngest of all. Then

$$\mathbf{I}^{(2)}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

One can see that $\mathbf{I}^{(2)}(t)$ is *not an identity matrix* even though it describes the coding vectors of all original uncoded user-2 packets. The need for defining such $\mathbf{I}^{(k)}(t)$ matrix is due to the fact that the arrivals of packets of different users are multiplexed, therefore consecutive columns generally does not represent consecutive packets of the same user. Instead, the columns are based on the arrival order.

Also we define $\mathbf{W}_k(t)$ as a column vector that vertically concatenates all the queue entries $W_{(k,\mathrm{i})}$ in flow-k queues; and define $\mathbf{Z}_k(t)$ as a column vector that vertically concatenates all the packets that have been received by destination d_k. Then we have the following lemmas.

Lemma 6. *In the end of time t, for all $k \in [K]$ we have $\mathbf{W}_k(t) = \mathbf{U}^{(k)}(t)\mathbf{X}(t)$, $\mathbf{Z}_k(t) = \mathbf{V}^{(k)}(t)\mathbf{X}(t)$, and $\mathbf{X}_k(t) = \mathbf{I}^{(k)}(t)\mathbf{X}(t)$.*

Proof. The result is straightforward following the construction of $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ in Section 2.4.2 and the definitions in this appendix. □

Recall that $\langle \mathbf{A} \rangle$ denotes the row space of matrix $\mathbf{A}$ and we use $\langle \mathbf{A} \rangle \oplus \langle \mathbf{B} \rangle$ to denote the *sum space* of $\langle \mathbf{A} \rangle$ and $\langle \mathbf{B} \rangle$. That is,

$$\langle \mathbf{A} \rangle \oplus \langle \mathbf{B} \rangle \triangleq \langle (\mathbf{A}^{\mathsf{T}}, \mathbf{B}^{\mathsf{T}})^{\mathsf{T}} \rangle. \tag{A.43}$$

Then we have

Definition A.5.1 (Non-interfering vector)**.** *In the end of time t, any arbitrarily given row vector $\mathbf{u}$ is "non-interfering" from the perspective of destination d_{i} if $\mathbf{u} \in \langle \mathbf{V}^{(\mathrm{i})}(t) \rangle \oplus \langle \mathbf{I}^{(\mathrm{i})}(t) \rangle$.*

Lemma 7. *In the end of time t, for any fixed $k \in [K]$ and fixed $S \in 2^{[K]\setminus\{k\}}$, consider any arbitrarily given entry $W_{(k,\mathrm{j})}(t)$ in queue $Q_S^{(k)}$. The corresponding vector $\mathbf{u}_{(k,\mathrm{j})}(t)$ is* non-interfering *from the perspective of d_{i} for all $\mathrm{i} \in S \cup \{k\}$.*

Proof. We prove this lemma by mathematical induction on time index t. First consider in the end of the time 0 (before any transmission), since all the queues are empty, Lemma 7 holds in the end of time 0.

Suppose Lemma 7 is satisfied in the end of time $t-1$. We first argue that any vector $\mathbf{u}$ that is non-interfering for user d_k at time $t-1$ is also non-interfering for user d_k at time t, once we pad the end of $\mathbf{u}$ by zeros that take into account the newly added columns during time t. The reason is that $\langle \mathbf{V}^{(k)}(t) \rangle$ represents the row space spanned by the global encoding kernels of all the packets received by d_k at time t. As t becomes larger, the row space becomes larger as well. Again, if we abuse the notation slightly by implicitly assuming zeros are padded to take into account the newly added columns, then this simple observation implies

$\langle \mathbf{V}^{(k)}(t-1)\rangle \subseteq \langle \mathbf{V}^{(k)}(t)\rangle$. Similarly, $\langle \mathbf{I}^{(k)}(t-1)\rangle \subseteq \langle \mathbf{I}^{(k)}(t)\rangle$ since $\mathbf{I}^{(k)}(t)$ represents the row space spanned by the user-k packets that have already arrived at the source s. Since both $\mathbf{V}^{(k)}(t)$ and $\mathbf{I}^{(k)}(t)$ monotonically increase over time, any $\mathbf{u}$ that is non-interfering at time $t-1$ is also non-interfering at time t. As a result, we only need to consider those queued entries that are either newly injected to the queues during time t, or those that have been modified during time t.

According to Section 2.4.2, Component 2 will insert new entries in the queues and Component 5 will modify the queued entries according to the packet reception set $S_{\rm rx}(t)$. All other components do not change the queued entry. In the sequel, we discuss Components 2 and 5, respectively.

In Component 2, if there is a flow-k input packet $X_{k,\rm i}$ arriving at source, then we create a new entry $W_{(k,\rm i)}(t) = X_{k,\rm i}$ in $Q^{(k)}_{\emptyset}$. In the end of time t we add a new row $\mathbf{e}_{(k,\rm i)}$ to matrix $\mathbf{U}^{(k)}$. Since $\mathbf{e}_{(k,\rm i)} \in \langle \mathbf{I}^{(k)}(t)\rangle$, the newly added entry $W_{(k,\rm i)}(t)$ in $Q^{(k)}_{\emptyset}$ is non-interfering from the perspective of d_k. Lemma 7 holds for the newly inserted entry.

After transmission $Y(t)$, Component 5 updates the queues according to the reception set $S_{\rm rx}(t)$. We discuss the following cases in the end of time t.

Case 1: $k \in S_{\rm rx}(t)$, that is, d_k receives $Y(t)$. According to Component 5, the entry $W_{(k,\rm j_k)}(t-1)$ is then removed from the queue without creating any new entry. Since Lemma 7 focuses only on the entries still in the queues, Lemma 7 holds naturally.

Case 2: $k \notin S_{\rm rx}(t)$. According to *Component 5*, the entry $W_{(k,\rm j_k)}(t-1)$ is removed from $Q^{(k)}_{T^*\backslash\{k\}}(t)$ and a new entry $W_{(k,\rm j_k)}(t) = Y(t)$ is injected to a queue $Q^{(k)}_{\tilde{S}^*}(t)$ satisfying $\tilde{S}^* \subseteq (T^*\backslash\{k\}) \cup S_{\rm rx}(t)$, with the corresponding new global encoding kernel $\mathbf{u}_{(k,\rm j_k)}(t) = \mathbf{y}(t)$.

By induction, for any fixed $k \in T^*$, $\mathbf{u}_{(k,\rm j_k)}(t) \in \langle \mathbf{V}^{(\rm i)}(t-1)\rangle \oplus \langle \mathbf{I}^{(\rm i)}(t-1)\rangle$ for all $\rm i \in (T^*\backslash\{k\}) \cup \{k\} = T^*$. Therefore, the corresponding global encoding kernel $\mathbf{y}(t) = \sum_{k\in T^*} \beta_k \mathbf{u}_{(k,\rm j_k)}(t) \in \langle \mathbf{V}^{(\rm i)}(t-1)\rangle \oplus \langle \mathbf{I}^{(\rm i)}(t-1)\rangle$ is non-interfering for all $\rm i \in T^*$. Since the new packet $W_{(k,\rm j_k)}(t) = Y(t)$ will be injected to $Q^{(k)}_{\tilde{S}^*}$ with the global encoding kernel $\mathbf{u}_{(k,\rm j_k)}(t) = \mathbf{y}(t)$, what remains to prove is that $\mathbf{y}(t)$ is non-interfering for all $\rm i \in (\tilde{S}^* \cup \{k\})\backslash T^*$.

To that end, we observe that since $\tilde{S}^* \subseteq (T^*\backslash\{k\}) \cup S_{\rm rx}(t)$, we have $((\tilde{S}^* \cup \{k\})\backslash T^*) \subseteq S_{\rm rx}(t)$. On the other hand, for all $\rm i \in S_{\rm rx}(t)$, user $d_{\rm i}$ receives $Y(t)$. Therefore, $\mathbf{y}(t) \in \langle \mathbf{V}^{(\rm i)}(t)\rangle$ for all $\rm i \in S_{\rm rx}(t)$. The newly added vector $\mathbf{y}(t)$ is non-interfering from the perspective of all

d_{i}, $\mathrm{i} \in (\tilde{S}^* \cup \{k\} \backslash T^*) \subseteq S_{\mathrm{rx}}(t)$. Discussion of Cases 1 and 2 show that Lemma 7 holds after Component 5 as well. The proof is thus complete by induction on the time index t. □

We now prove the following lemma.

Lemma 8. *In the end of time t we have $\langle \mathbf{V}^{(k)}(t) \rangle \oplus \langle \mathbf{U}^{(k)}(t) \rangle = \langle \mathbf{V}^{(k)}(t) \rangle \oplus \langle \mathbf{I}^{(k)}(t) \rangle$ for all $k \in [K]$.*

Proof. We prove this lemma by mathematical induction on time index t. First consider in the end of time 0 (before any transmission). Since there has not been any packet arrival at source s, we have $\langle \mathbf{I}^{(k)}(0) \rangle = \{\mathbf{0}\}$, i.e., the row space of the arrived messages contains only the all-zero vector. Similarly, since all queues are empty at time 0, we have $\langle \mathbf{U}^{(k)}(0) \rangle = \{\mathbf{0}\}$. Finally, since no packet has ever been delivered to the destinations, we have $\langle \mathbf{V}^{(k)}(0) \rangle = \{\mathbf{0}\}$. As a result, Lemma 8 holds at time 0.

Suppose, Lemma 8 is satisfied in the end of time $t-1$. We will prove that Lemma 8 still holds when we execute Components 2 to 5 sequentially in time t.

In the beginning of time t, in Component 2, we add new row vector $\mathbf{u}_{(k,\mathrm{i})} = \mathbf{e}_{(k,\mathrm{i})}$ for each arrival input packet $X_{k,\mathrm{i}}$, $\mathrm{i} \in \{M_k(t-1)+1, \ldots, M_k(t)\}$. For simplicity, we use $\mathbf{E}^{(k)}(t)$ to denote the matrix with the rows $\{\mathbf{e}_{(k,\mathrm{i})} : \mathrm{i} = M_k(t-1)+1, \ldots, M_k(t)\}$. It is then clear that $\langle \mathbf{U}^{(k)}(t) \rangle = \langle \mathbf{U}^{(k)}(t-1) \rangle \oplus \langle \mathbf{E}^{(k)}(t) \rangle$ and $\langle \mathbf{I}^{(k)}(t) \rangle = \langle \mathbf{I}^{(k)}(t-1) \rangle \oplus \langle \mathbf{E}^{(k)}(t) \rangle$. Therefore, (8) holds after executing Component 2.

Component 3 selects the coding set T^* and does not change the matrices $\mathbf{U}^{(k)}(t)$, $\mathbf{V}^{(k)}(t)$, and $\mathbf{I}^{(k)}(t)$. Similarly, Component 4 chooses the mixing coefficients β_k, $k \in T^*$, and does not change the matrices $\mathbf{U}^{(k)}(t)$, $\mathbf{V}^{(k)}(t)$, and $\mathbf{I}^{(k)}(t)$. As a result, (8) holds after executing Components 3 and 4 as well.

We now consider Component 5. For ease of notation, we use the subscript "old" to denote the $\mathbf{U}^{(k)}(t)$ and $\mathbf{V}^{(k)}(t)$ matrices right before we execute Component 5 and use the subscript "new" to denote the matrices right after we execute Component 5. Recall that for all $k_0 \in T^*$, we choose the head-of-line packet $W_{(k_0,\mathrm{j}_{k_0})}(t)$ and construct $\mathbf{U}^{(k_0)}(t)$ by removing the row $\mathbf{u}_{(k_0,\mathrm{j}_{k_0})}(t)$ corresponding to $W_{(k_0,\mathrm{j}_{k_0})}(t)$ from $\mathbf{U}^{(k_0)}(t)$. We now consider the following cases.

Case 1: Consider those destinations d_{k_0} such that $k_0 \notin T^*$. According to Component 5, there is no change to any of the flow-k_0 queues. Therefore, Lemma 8 holds for such k_0 after executing Component 5.

Case 2: Consider those destinations d_{k_0} such that $k_0 \in T^*$ and $k_0 \in S_{\text{rx}}(t)$. If $Q^{(k_0)}_{T^*\backslash\{k_0\}}$ is empty, then we use a null packet $W_{(k_0,0)}$ in Component 4 such that $\mathbf{U}^{(k_0)}_{\text{new}} = \mathbf{U}^{(k_0)}_{\text{old}}$ and $\mathbf{V}^{(k_0)}_{\text{new}} = \mathbf{V}^{(k_0)}_{\text{old}}$ and Lemma 8 thus holds. If $Q^{(k_0)}_{T^*\backslash\{k_0\}}$ is non-empty, according to Component 5, the entry $W_{(k_0,\mathrm{j}_{k_0})}(t)$ is removed from $Q^{(k_0)}_{T^*\backslash\{k_0\}}(t)$ without creating new entry. Therefore the new row space $\langle \mathbf{V}^{(k_0)}_{\text{new}} \rangle$ becomes $\langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{y}(t) \rangle$ and the new row space $\langle \mathbf{U}^{(k_0)}_{\text{new}} \rangle = \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle$. In the following we will prove

$$\begin{aligned}
&\langle \mathbf{V}^{(k_0)}_{\text{new}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{new}} \rangle \\
=&\langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle \\
=&\langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle && \text{(A.44)} \\
=&\langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle. && \text{(A.45)}
\end{aligned}$$

Eq. (A.45) holds due to the following reason. Lemma 7 shows that for all $k \in T^*$ $\mathbf{u}_{(k,\mathrm{j}_k)}(t)$ is non-interfering from the perspective d_{i}, $\mathrm{i} \in (T^*\backslash\{k\}) \cup \{k\} = T^*$. Therefore by the induction condition, we have for all $k \in T^*$ $\mathbf{u}_{(k,\mathrm{j}_k)}(t) \in \langle \mathbf{V}^{(k_0)}(t) \rangle \oplus \langle \mathbf{I}^{(k_0)}(t) \rangle = \langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle$ and hence $\mathbf{y}(t) = \sum_{k \in T^*} \beta_k \mathbf{u}_{(k,\mathrm{j}_k)}(t) \in \langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle$.

We then show that equality (A.44) always holds. Since $\mathbf{U}^{(k_0)}_{\text{old}}$ is obtain by removing $\mathbf{u}_{(k_0,\mathrm{j}_{k_0})}(t)$, if $\mathbf{u}_{(k_0,\mathrm{j}_{k_0})}(t) \in \langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle$, then $\langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle = \langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle$ and (A.44) holds naturally. If $\mathbf{u}_{(k_0,\mathrm{j}_{k_0})}(t) \notin \langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle$, we then notice that

$$\langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle \supseteq \langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle \tag{A.46}$$

and

$$\mathsf{rank}(\langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle) \geq \mathsf{rank}(\langle \mathbf{V}^{(k_0)}_{\text{old}} \rangle \oplus \langle \mathbf{U}^{(k_0)}_{\text{old}} \rangle) - 1 \tag{A.47}$$

where both inequalities are due to that $\mathbf{U}_{\text{old}}^{(k_0)}$ is obtained from $\mathbf{U}_{\text{old}}^{(k_0)}$ by removing only 1 row. Nonetheless, by the way we select $\{\beta_k\}$ in Component 4 such that $\mathbf{y}(t) \notin \langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle$. Therefore, we also have

$$\begin{aligned} &\mathsf{rank}(\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle) \\ &= \mathsf{rank}(\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle) + 1. \end{aligned} \tag{A.48}$$

Together, we have

$$\mathsf{rank}(\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle) \tag{A.49}$$

$$= \mathsf{rank}(\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle) \tag{A.50}$$

$$\geq \mathsf{rank}(\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle) \tag{A.51}$$

$$\geq \left(\mathsf{rank}(\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle) - 1\right) + 1 \tag{A.52}$$

where (A.50) is a repeat of (A.45); (A.51) follows from (A.46); and (A.52) follows from (A.47) and (A.48). This implies that the two subspaces $\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle$ and $\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle$ have the same rank. Since (A.46) also implies

$$\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle \supseteq \langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{y}(t) \rangle \oplus \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle$$

the two spaces must be equal. We have thus proven (A.44). Since (A.44) holds after executing Component 5, by the induction hypothesis, we have proven Lemma 8 for such k_0 after executing Component 5.

Case 3: Consider the destinations d_{k_0} such that $k_0 \in T^*$ and $k_0 \notin S_{\text{rx}}(t)$. That is d_{k_0} does not receive the desired packet $Y(t)$. According to Component 5, the entry $W_{(k_0,\mathrm{j}_{k_0})}(t)$ is then removed from $Q_{T^* \setminus \{k_0\}}^{(k_0)}(t)$ and an entry $W_{(k_0,\mathrm{j}_{k_0})}(t) = Y(t)$ is created in $Q_{\tilde{S}^*}^{(k_0)}(t)$ corresponding to new row $\mathbf{u}_{(k_0,\mathrm{j}_{k_0})}(t) = \mathbf{y}(t)$. Therefore the new row space $\langle \mathbf{V}_{\text{new}}^{(k_0)} \rangle$ equals to $\langle \mathbf{V}_{\text{old}}^{(k_0)} \rangle$ and the new row space $\langle \mathbf{U}_{\text{new}}^{(k_0)} \rangle = \langle \mathbf{U}_{\text{old}}^{(k_0)} \rangle \oplus \langle \mathbf{y}(t) \rangle$. Comparing Cases 2 and 3, we notice that the only difference is that in Case 2, the new vector $\mathbf{y}(t)$ is later classified as part of the $\mathbf{V}_{\text{new}}^{(k_0)}$ matrix but in Case 3, the vector $\mathbf{y}(t)$ is later classified as part of the $\mathbf{U}_{\text{new}}^{(k_0)}$ matrix. Since the

statement in Lemma 8 does not distinguish whether the changes actually happen to $\mathbf{V}_{\text{new}}^{(k_0)}$ or $\mathbf{U}_{\text{new}}^{(k_0)}$, by the same arguments as in Case 2, Lemma 8 holds for Case 3 as well.

Since Lemma 8 holds after sequentially executing Components 2 to 5, the proof is completed by induction on the time index t. □

Note that Lemma 8 directly implies the decodability defined in (2.9) and (2.10). Specifically, in the end of time slot t for any $k \in [K]$, $\langle \mathbf{V}^{(k)}(t) \rangle \oplus \langle \mathbf{U}^{(k)}(t) \rangle = \langle \mathbf{V}^{(k)}(t) \rangle \oplus \langle \mathbf{I}^{(k)}(t) \rangle$ implies that there exists a matrix $\mathbf{G}^{(k)}(t)$ such that

$$\begin{bmatrix} \mathbf{V}^{(k)}(t) \\ \mathbf{I}^{(k)}(t) \end{bmatrix} = \mathbf{G}^{(k)}(t) \begin{bmatrix} \mathbf{V}^{(k)}(t) \\ \mathbf{U}^{(k)}(t) \end{bmatrix}.$$

Therefore by Lemma 6, we have

$$\begin{bmatrix} \mathbf{Z}_k(t) \\ \mathbf{X}_k(t) \end{bmatrix} = \mathbf{G}^{(k)}(t) \begin{bmatrix} \mathbf{Z}_k(t) \\ \mathbf{W}_k(t) \end{bmatrix}. \tag{A.53}$$

Note that per our definition the received linearly coded packets are $[Z_k]_1^t = \mathbf{Z}_k(t)$ and the packets in the queues are $Q(t) = \{\mathbf{W}_1(t), \mathbf{W}_2(t), \cdots, \mathbf{W}_K(t)\}$. Therefore, (A.53) implies that there exists a decoder $g_{k,t}^{(\text{sq})}$ such that $\mathbf{X}_k(t) = g_{k,t}^{(\text{sq})}(\mathbf{Z}_k(t), \mathbf{W}_k(t), [S_{\text{rx}}]_1^t, [\vec{A}]_1^t)$, see (2.9) and (2.10). The proof of Proposition 2.4.1 is complete.

A.6 Proof of Propositions 2.4.2 and 2.4.3

Lemma 9. *Given any arbitrary integer $k \in [K]$ and rate value R_k, the statement "$\vec{x} \triangleq \{x_T \geq 0 : \forall T \in 2^{[K]} \backslash \{\emptyset\}\}$ satisfies Condition 2 in Proposition 2.3.1 for the given k" is equivalent to the existence of non-negative variables*

$$\vec{y} \triangleq \Big\{ y_{k;S_I \to d_k} \geq 0, y_{k;S_I \to S_O}^{[S_X]} \geq 0 : S_I \in 2^{[K] \backslash \{k\}}, \\ S_X \in 2^{[K] \backslash (S_I \cup \{k\})}, S_O \in 2^{S_I \cup S_X} \Big\} \tag{A.54}$$

satisfying the following three groups of equations: Group 1:

$$R_k \cdot 1_{\{S=\emptyset\}} + \sum_{\forall S_X, S_I} y_{k;S_I \to S}^{[S_X]} \leq y_{k;S \to d_k} + \sum_{\forall S_X, S_O} y_{k;S \to S_O}^{[S_X]},$$
$$\forall S \in 2^{[K]\setminus\{k\}} \tag{A.55}$$

Group 2:

$$y_{k;S_I \to d_k} \leq x_{S_I \cup \{k\}} \cdot p_k, \quad \forall S_I \in 2^{[K]\setminus\{k\}} \tag{A.56}$$

and Group 3:

$$\sum_{\forall S_O \in 2^{S_I \cup S_X}} y_{k;S_I \to S_O}^{[S_X]} \leq x_{S_I \cup \{k\}} \cdot p_{S_X \overline{[K]\setminus(S_I \cup S_X)}},$$
$$\forall S_I \in 2^{[K]\setminus\{k\}}, S_X \in 2^{[K]\setminus(S_I \cup \{k\})} \tag{A.57}$$

Proof. From Lemmas 1 and 2, it is clear that Condition 2 in Proposition 2.3.1 is equivalent to "the minimum-cut-value of the k-th virtual sub-network in Section 2.4 is no less than R_k". In the k-th virtual sub-network, if we use variables $y_{k;S_I \to d_k}$ to denote the flow value over the edge from $\text{vn}_{S_I}^{(k)}$ to $\text{vn}_{\text{grnd}}^{(k)}$ and use $y_{k;S_I \to S_O}^{[S_X]}$ to denote the flow value over the edge from $\text{vn}_{S_I}^{(k)}$ though auxiliary node S_X and then to $\text{vn}_{S_O}^{(k)}$, then Group 1 equality (A.55) is simply the flow-conservation law if we replace the $\leq$ in (A.55) by $=$. Groups 2 and 3 inequalities in (A.56) and (A.57) impose that the flow values are no larger than the assigned edge capacities. Therefore this lemma becomes a directly result of applying the max-flow min-cut theorem to the k-th virtual sub-network, if we replace the $\leq$ in (A.55) by $=$. Finally, it is well known, see [64], that replacing $=$ in the flow condition (A.55) by $\leq$ does not alter the max-flow characterization. The proof of this lemma is thus complete. □

Throughout this section of appendix, we perform exclusively queue-length-based stability analysis [65], which focuses on the length of the queue $|Q_S^{(k)}|$ instead of the content of the queue $Q_S^{(k)}$. To simplify the notation, in this appendix we slightly abuse the notation and

directly use $Q_S^{(k)}$ to represent the queue length so that we do not need to add the length operator $|\cdot|$ for all the queues.

Definition A.6.1 (Offered packet movement)**.** *The offered packet movement (including real and dummy packets) at time t is a set of Bernoulli random variables*

$$\left\{Y_{k;S_I\to d_k}(t) : k\in[K], S_I\in 2^{[K]\backslash\{k\}}\right\} \tag{A.58}$$

$$\text{and } \left\{Y^{[S_X]}_{k;S_I\to S_O}(t) : k\in[K], S_I\in 2^{[K]\backslash\{k\}}, S_X\in 2^{[K]\backslash(S_I\cup\{k\})}, S_O\in 2^{S_I\cup S_X}\right\} \tag{A.59}$$

such that the random variable $Y_{k;S_I\to d_k}(t)=1$ if and only if there is a packet removed from the virtual queue $Q_{S_I}^{(k)}$ to destination d_k (see Component 5) and $Y^{[S_X]}_{k;S_I\to S_O}(t)=1$ if and only if there is a packet movement from $Q_{S_I}^{(k)}$ to $Q_{S_O}^{(k)}$ in the scenario $k\notin S_{\rm rx}(t)$ and $S_X=S_{\rm rx}(t)\backslash S_I$.

The offered packet movement random variable $\vec{Y}(t)$ is defined from the edge's perspective. We now define the queue-length displacement random variables $\Delta Q_S^{(k)}(t)$, which is defined based on the queue's perspective.

$$\begin{aligned}\Delta Q_S^{(k)}(t) \triangleq A_k(t)\cdot 1_{\{S=\emptyset\}} &+ \sum_{\forall S_X,S_I} Y^{[S_X]}_{k;S_I\to S}(t)\\ &- Y_{k;S\to d_k}(t) - \sum_{\forall S_X,S_O} Y^{[S_X]}_{k;S\to S_O}(t).\end{aligned} \tag{A.60}$$

It is clear from the above definition, we have

$$Q_S^{(k)}(t+1) = \left(Q_S^{(k)}(t)+\Delta Q_S^{(k)}(t)\right)^+ \tag{A.61}$$

since $\Delta Q_S^{(k)}(t)$ captures the packet arrival process $A_k(t)$ and the packet removal/movement random variables $\vec{Y}(t)$ at time t.

A.6.1 Proof of Proposition 2.4.2

We prove this proposition by the Lyapunov drift analysis. Define a Lyapunov function as follows.

$$L_q(\vec{Q},t) \triangleq \frac{1}{2}\sum_{k\in[K],S\in 2^{[K]\setminus\{k\}}} \left(Q_S^{(k)}(t)\right)^2. \tag{A.62}$$

We will prove that there exists a negative drift when the summation of all queue lengths is sufficiently large. That is,

$$\begin{aligned}&\mathsf{E}\left\{L_q(\vec{Q},t+1)-L_q(\vec{Q},t)\middle|\vec{Q}(t)\right\}\\ &\leq \mathsf{const}-\epsilon\cdot\left(\sum_{k\in[K],S\in 2^{[K]\setminus\{k\}}} Q_S^{(k)}(t)\right),\end{aligned} \tag{A.63}$$

for some fixed constant const. The negative drift then immediately implies that the queue lengths are stable.

To prove the negative drift, we first notice that by Lemma 9 if there exists variables $\vec{x}$ satisfying (2.18) and strict (2.17), then there exist $\vec{y}$, joint with $\vec{x}$, satisfying (A.55), (A.56), (A.57), and strict (2.17). We now argue that, without loss of generality, we can further assume that $\vec{x}$ and $\vec{y}$ satisfy (2.17), (A.56), and (A.57) with exact equality, and satisfy (A.55) with strict inequality for all $k\in[K]$ and $S\in 2^{[K]\setminus\{k\}}$.

To that end, we observe that since (2.17) holds with strict inequality, we can increase all x_T for all T by the same amount $\delta>0$ such that the new x_T satisfies (2.17) with equality. The increase of x_T for all T ensures that (A.56) is now satisfied with strict inequality since we assume $p_k>0$ for all k. Inequality (A.57) still holds since x_T only appears in the right-hand side of (A.57). Since (A.56) is now strict inequality, we can increase $y_{k;S_I\to d_k}$ value by a strictly positive value until (A.56) becomes equality for all $k\in[K]$, $S_I\in 2^{[K]\setminus\{k\}}$. The increase of $y_{k;S_I\to d_k}$ ensures that (A.55) becomes strict inequality for all $k\in[K]$, $S_I\in 2^{[K]\setminus\{k\}}$, since $y_{k;S_I\to d_k}$ only appears on the right-hand side of (A.55). Finally, if any of (A.57) is a strictly inequality, then we can increase the corresponding $y^{[S_X]}_{k;S_I\to S_I}$ value until (A.57) becomes equality. The increase of $y^{[S_X]}_{k;S_I\to S_I}$ does not alter (A.55) since $y^{[S_X]}_{k;S_I\to S_I}$ appears on both sides

of (A.55). The final $\vec{x}$ and $\vec{y}$ satisfy (2.17), (A.56), and (A.57) with equality and satisfy (A.55) with strict inequality for all k, S. This important observation implies that for some $\epsilon > 0$,

$$\begin{aligned} & R_k \cdot 1_{\{S=\emptyset\}} + \sum_{\forall S_X, S_I} y^{[S_X]}_{k;S_I \to S} - y_{k;S \to d_k} - \sum_{\forall S_X, S_O} y^{[S_X]}_{k;S \to S_O} \\ & \leq -\epsilon, \quad \forall k \in [K], S \in 2^{[K]\setminus\{k\}}. \end{aligned} \tag{A.64}$$

We are now ready to prove the negative drift. Squaring (A.61) and taking the conditional expectations yields

$$\begin{aligned} & \mathsf{E}\left\{ \left(Q^{(k)}_S(t+1)\right)^2 \middle| \vec{Q}(t) \right\} \\ \leq & \mathsf{E}\left\{ \left(Q^{(k)}_S(t) + \Delta Q^{(k)}_S(t)\right)^2 \middle| \vec{Q}(t) \right\} \\ = & \mathsf{E}\left\{ \left(Q^{(k)}_S(t)\right)^2 \middle| \vec{Q}(t) \right\} + 2 \cdot \mathsf{E}\left\{ Q^{(k)}_S(t) \cdot \Delta Q^{(k)}_S(t) \middle| \vec{Q}(t) \right\} \\ & + \mathsf{E}\left\{ \left(\Delta Q^{(k)}_S(t)\right)^2 \middle| \vec{Q}(t) \right\}. \end{aligned} \tag{A.65}$$

We then notice that for any given k, $Y_{k;S_I \to d_k}(t)$ and $Y^{[S_X]}_{k;S_I \to S_O}(t)$ denote the packet movement within the virtual queue at time t. Therefore, at most one of them can be of value 1 at a given time t. Then by the definition of $\Delta Q^{(k)}_S$ in (A.60), we have for all $k \in [K]$, $S \in 2^{[K]\setminus\{k\}}$

$$A_k(t) \cdot 1_{\{S=\emptyset\}} - 1 \leq \Delta Q^{(k)}_S(t) \leq A_k(t) \cdot 1_{\{S=\emptyset\}} + 1. \tag{A.66}$$

We thus have

$$\begin{aligned} & \mathsf{E}\left\{ \left(\Delta Q^{(k)}_S(t)\right)^2 \middle| \vec{Q}(t) \right\} \leq \mathsf{E}\left\{ \left(A_k(t)+1\right)^2 \middle| \vec{Q}(t) \right\} \\ = & \mathsf{E}\left\{ \left(A_k(t)\right)^2 \middle| \vec{Q}(t) \right\} + 2 \cdot \mathsf{E}\left\{ A_k(t) \middle| \vec{Q}(t) \right\} + 1 \\ = & (R_k + R_k^2) + 2R_k + 1, \end{aligned} \tag{A.67}$$

where we use the fact that a Poisson random variable $A_k(t)$ with arrival rate R_k has $\mathsf{E}\{A_k^2(t)\} = R_k + R_k^2$. Then by (A.65) and (A.67), the Lyapunov drift function at time t becomes

$$\begin{aligned}
&\mathsf{E}\left\{ L_q(\vec{Q}, t+1) - L_q(\vec{Q}, t) \middle| \vec{Q}(t) \right\} \\
\leq &\sum_{k\in[K], S\in 2^{[K]\setminus\{k\}}} \left(\frac{1}{2}\mathsf{E}\left\{ \left(\Delta Q_S^{(k)}(t)\right)^2 \middle| \vec{Q}(t) \right\} \right. \\
&\left. + \mathsf{E}\left\{ Q_S^{(k)}(t) \cdot \Delta Q_S^{(k)}(t) \middle| \vec{Q}(t) \right\} \right) \\
= &\mathsf{const} + \sum_{k\in[K], S\in 2^{[K]\setminus\{k\}}} Q_S^{(k)}(t) \cdot \mathsf{E}\left\{ \Delta Q_S^{(k)}(t) \middle| \vec{Q}(t) \right\}
\end{aligned}$$

where the constant being $\mathsf{const} \triangleq \sum_{k\in[K]} 2^{K-2}(R_k^2 + 3R_k + 1)$. The remaining proof of the negative drift becomes proving

$$\sum_{k\in[K], S\in 2^{[K]\setminus\{k\}}} Q_S^{(k)}(t) \cdot \mathsf{E}\left\{ \Delta Q_S^{(k)}(t) \middle| \vec{Q}(t) \right\} \tag{A.68}$$

$$\leq -\epsilon \left(\sum_{k\in[K], S\in 2^{[K]\setminus\{k\}}} Q_S^{(k)}(t) \right). \tag{A.69}$$

To prove (A.69), we first show that our scheme of choosing the coding set $T^*(t)$ in Component 3 and the destination queue $Q_{\tilde{S}^*}^{(k_0)}$ in Component 5 minimizes the summation in (A.68) among all the schemes which move packets among the nodes in the virtual network, conditioning on knowing the queue lengths $\vec{Q}(t)$.

Suppose that at time t, a competing scheme chooses the coding set $T(t)$ and we denote the resulting offered packet movement by $\vec{Y}(t)$ and the resulting queue-length displacement by $\Delta Q_S^{(k)}(t)$ for all k, S. We then have

$$\begin{aligned}
&\sum_{\forall k,S} Q_S^{(k)}(t) \cdot \mathsf{E}\left\{ \Delta Q_S^{(k)}(t) \middle| \vec{Q}(t) \right\} \\
= &\sum_{k\in[K]} R_k \cdot Q_\emptyset^{(k)}(t) - \sum_{\forall k,S} Q_S^{(k)}(t) \mathsf{E}\left\{ Y_{k;S\to d_k}(t) | \, \vec{Q}(t) \right\} \\
- &\left(\sum_{\forall k, S_I, S_X, S_O} \left(Q_{S_I}^{(k)}(t) - Q_{S_O}^{(k)}(t) \right) \mathsf{E}\left\{ Y_{k;S_I\to S_O}^{[S_X]}(t) \middle| \vec{Q}(t) \right\} \right)
\end{aligned} \tag{A.70}$$

where (A.70) follows from (A.60). Recall that $T(t)$ is a deterministic function of $\vec{Q}(t)$ and denote the packet movement destination by $Q_{\tilde{S}}^{(k)}(t)$, where $\tilde{S}$ is a function[3] of S_X. Due to the memorylessness of the underlying PECs, we have the conditional expectation being

$$\mathsf{E}\left\{Y_{k;S_I\to d_k}(t)\,\middle|\,\vec{Q}(t)\right\} = 1_{\{T(t)=S_I\cup\{k\}\}}\cdot p_k \tag{A.71}$$

$$\begin{aligned}\mathsf{E}\left\{Y_{k;S_I\to S_O}^{[S_X]}(t)\middle|\,\vec{Q}(t)\right\} &= 1_{\{T(t)=S_I\cup\{k\}\}}\cdot\\ &\Pr\left(k\notin S_{\rm rx}(t), S_X = S_{\rm rx}(t)\backslash S_I\right)\cdot 1_{\{S_O=\tilde{S}\}}\end{aligned} \tag{A.72}$$

where (A.71) holds since the offered movement $Y_{k;S_I\to d_k}(t)=1$ if and only if $T(t)=S_I\cup\{k\}$, $k\in S_{\rm rx}(t)$, and by the memorylessness of the channel; and (A.72) holds since $Y_{k;S_I\to S_O}^{[S_X]}(t)=1$ if and only if $T(t)=S_I\cup\{k\}$, $k\notin S_{\rm rx}(t)$, $S_X=S_{\rm rx}(t)\backslash S_I$, $S_O=\tilde{S}$, and by the memorylessness of the channel.

Observe that $\Pr(k\notin S_{\rm rx}(t), S_X=S_{\rm rx}(t)\backslash S_I) = p_{S_X\overline{[K]\backslash(S_I\cup S_X)}}$, eq. (A.70) then becomes

$$\begin{aligned}&\sum_{k\in[K]} R_k\cdot Q_\emptyset^{(k)}(t) - \sum_{\forall S_I\in 2^{[K]\backslash\{k\}}} 1_{\{T(t)=S_I\cup\{k\}\}}\cdot\\ &\left(Q_{S_I}^{(k)}(t)\cdot p_k + \sum_{\forall S_X} p_{S_X\overline{[K]\backslash(S_I\cup S_X)}}\left(Q_{S_I}^{(k)}(t) - Q_{\tilde{S}}^{(k)}(t)\right)\right)\\ &= \sum_{k\in[K]} R_k\cdot Q_\emptyset^{(k)}(t) - \sum_{k\in T(t)}\Big(Q_{T(t)\backslash\{k\}}^{(k)}(t)\\ &\quad - \sum_{\forall S_X\in 2^{[K]\backslash T(t)}} p_{S_X\overline{[K]\backslash(S_X\cup T(t)\backslash\{k\})}}\cdot Q_{\tilde{S}}^{(k)}(t)\Big),\end{aligned} \tag{A.73}$$

where the last equality uses the observation that $p_k + \sum_{\forall S_X\in 2^{[K]\backslash T(t)}} p_{S_X\overline{[K]\backslash(S_X\cup T(t)\backslash\{k\})}} = 1$.

[3]↑A more precise notation should be $\tilde{S}(S_X)$ instead. However for notational simplicity we simply use $\tilde{S}$.

Therefore from (A.73) we obtain

$$\sum_{\forall k,S} Q_S^{(k)}(t) \cdot \mathsf{E}\left\{\Delta Q_S^{(k)}(t) \middle| \vec{Q}(t)\right\}$$

$$\geq \sum_{k\in[K]} R_k \cdot Q_\emptyset^{(k)}(t) - \sum_{k\in T(t)} \mathrm{bp}\left(Q_{T(t)\backslash\{k\}}^{(k)}(t)\right) \tag{A.74}$$

$$\geq \sum_{k\in[K]} R_k \cdot Q_\emptyset^{(k)}(t) - \sum_{k\in T^*(t)} \mathrm{bp}\left(Q_{T^*(t)\backslash\{k\}}^{(k)}(t)\right) \tag{A.75}$$

$$= \sum_{\forall k,S} Q_S^{(k)}(t) \cdot \mathsf{E}\left\{\Delta Q_S^{(k)}(t) \middle| \vec{Q}(t)\right\} \tag{A.76}$$

where (A.74) follows from (A.70), (A.73), and the definition of the backpressure expression in (2.26). Specifically, for the same event $\{k \notin S_{\mathrm{rx}}(t), S_X = S_{\mathrm{rx}}(t)\backslash T(t)\}$, the backpressure expression $\mathrm{bp}(Q_{T(t)\backslash\{k\}}^{(k)}(t))$ involves the term $q(k, (T(t)\backslash\{k\}) \cup S_X)$ in (2.25), which chooses $\tilde{S}^*$ that minimizes $Q_{\tilde{S}}^{(k)}$. Comparing the backpressure expression to (A.73), replacing $\tilde{S}$ by $\tilde{S}^*$ decreases the overall value and we thus have (A.74). Eq. (A.75) follows from (2.27) since our coding set choice $T^*(t)$ maximizes the sum of the backpressure; (A.76) holds by the same reasons as (A.70), (A.73), and (2.26) with the following simple substitution $\Delta Q_S^{(k)}(t) = \Delta Q_S^{(k)}(t)$, $T(t) = T^*(t)$, and $\tilde{S} = \tilde{S}^*$. The above arguments show that at any time t our scheme attains the minimum $\sum_{\forall k,S} Q_S^{(k)}(t) \cdot \mathsf{E}\{\Delta Q_S^{(k)}(t)|\vec{Q}(t)\}$ among all possible scheme designs that move packets among the nodes in the virtual network.

Given $\vec{x}$ and $\vec{y}$ satisfying (A.56), (A.57) with equality and (A.64), we consider a competing scheme that chooses the coding set $T(t)$ randomly according to the probability distribution $\{x_T\}$ (recalling that $\sum_T x_T = 1$), and chooses the new destination queue $Q_{\tilde{S}}^{(k)}$, under event $\{k \notin S_{\mathrm{rx}}(t), S_X = S_{\mathrm{rx}}(t)\backslash T(t)\}$, randomly with the conditional distribution

$$\Pr\left(\tilde{S}\middle| S_X, T(t)\right) = \frac{y_{k;T(t)\backslash\{k\}\to\tilde{S}}^{[S_X]}}{x_{T(t)} \cdot p_{S_X\overline{[K]\backslash(S_X\cup T(t)\backslash\{k\})}}} \tag{A.77}$$

which satisfies $\sum_{\forall \tilde{S}} \Pr\left(\tilde{S} \middle| S_X, T(t)\right) = 1$. If we use $\Delta Q_S^{(k)}(t)$ to denote the drift under this competing scheme, we will have

$$\begin{aligned} &\sum_{k\in[K], S\in 2^{[K]\setminus\{k\}}} Q_S^{(k)}(t) \cdot \mathsf{E}\left\{\Delta Q_S^{(k)}(t) \middle| \vec{Q}(t)\right\} \\ \leq &-\epsilon \left(\sum_{k\in[K], S\in 2^{[K]\setminus\{k\}}} Q_S^{(k)}(t)\right). \end{aligned} \tag{A.78}$$

The reason is that this scheme randomly chooses the coding set $T(t)$ and destination queue $\tilde{S}$ independently such that

$$\begin{aligned} \mathsf{E}\left\{Y_{k;S\to d_k}(t) \middle| \vec{Q}(t)\right\} &= \Pr\left(T(t) = S\cup\{k\}\right)\cdot p_k \\ &= y_{k;S\to d_k} \end{aligned} \tag{A.79}$$

$$\begin{aligned} &\mathsf{E}\left\{Y_{k;S_I\to S_O}^{[S_X]}(t) \middle| \vec{Q}(t)\right\} = \Pr\left(T(t) = S_I\cup\{k\}\right)\cdot \\ &p_{S_X\overline{[K]\setminus(S_I\cup S_X)}} \cdot \Pr\left(S_O|S_X, T(t)\right) = y_{k;S_I\to S_O}^{[S_X]}. \end{aligned} \tag{A.80}$$

where (A.79) and (A.80) hold due to $\Pr(T(t) = S_I\cup\{k\}) = x_{S_I\cup\{k\}}$, the memorylessness of the underlying PECs, and (A.77). Consequently, we have

$$\begin{aligned} \mathsf{E}\left\{\Delta Q_S^{(k)}(t) \middle| \vec{Q}(t)\right\} &= R_k\cdot 1_{\{S=\emptyset\}} + \sum_{\forall S_X, S_I} y_{k;S_I\to S}^{[S_X]} \\ - y_{k;S\to d_k} &- \sum_{\forall S_X, S_O} y_{k;S\to S_O}^{[S_X]} \leq -\epsilon, \end{aligned} \tag{A.81}$$

where the equality holds by substituting (A.79) and (A.80) into (A.60) and by (A.64).

Finally, since we have already proven that our scheme minimizes (A.68), by (A.68) and (A.78) we thus have (A.69). The proof of the negative drift and the corresponding Lyapunov analysis is thus complete.

A.6.2 Proof of Proposition 2.4.3

Consider any rate $\vec{R} = (R_1, \ldots, R_K)$ that can be stabilized by the proposed 5-component sequential coding scheme. By Lemma 4, there exists a constant $D < \infty$ and a subsequence $\{t_m\}$ such that for all positive integers $m = 1, 2, \cdots$

$$\sum_{k\in[K], S\in 2^{[K]\setminus\{k\}}} \mathsf{E}\left\{Q_S^{(k)}(t_m)\right\} \leq D. \tag{A.82}$$

Since each queue length is always nonnegative, we immediately have: For all $k \in [K]$ and $S \in 2^{[K]\setminus\{k\}}$,

$$\mathsf{E}\left\{Q_S^{(k)}(t_m)\right\} \leq D. \tag{A.83}$$

Let $T^*(t)$ denote the coding set chosen in Component 3. We now construct[4] the following nonnegative variables $\vec{x}$ and $\vec{y}$ based on our 5-component sequential coding scheme:

$$x_T \triangleq \lim_{m\to\infty} \frac{1}{t_m} \sum_{\tau=0}^{t_m-1} 1_{\{T^*(\tau)=T\}}, \tag{A.84}$$

$$y_{k;S_I\to d_k} \triangleq \lim_{m\to\infty} \frac{1}{t_m} \sum_{\tau=0}^{t_m-1} \mathsf{E}\left\{Y_{k;S_I\to d_k}(\tau)\right\}, \tag{A.85}$$

$$y^{[S_X]}_{k;S_I\to S_O} \triangleq \lim_{m\to\infty} \frac{1}{t_m} \sum_{\tau=0}^{t_m-1} \mathsf{E}\left\{Y^{[S_X]}_{k;S_I\to S_O}(\tau)\right\}. \tag{A.86}$$

We then prove that above $\vec{x}$ variables satisfy (2.17) with equality and (2.18) in Proposition 2.3.1. By Lemma 9, we equivalently show that the above $\vec{x}$ and $\vec{y}$ satisfy (2.17) with equality and jointly satisfy (A.55) to (A.57). Firstly, we have $\sum_{\forall T\in 2^{[K]}\setminus\{\emptyset\}} x_T = \lim_{m\to\infty} \frac{t_m}{t_m} = 1$, which satisfies (2.17) with equality. To prove (A.56), we notice that by Component 5

$$Y_{k;S_I\to d_k}(\tau) = 1_{\{T^*(\tau)=S_I\cup\{k\}\}} \cdot 1_{\{k\in S_{\text{rx}}(\tau)\}}. \tag{A.87}$$

[4]↑The limits in (A.84) to (A.86) may not converge. If so, we simply use a convergent subsequence of the original sequence $\{t_m\}$ and the rest of the proof holds verbatim.

Since the channel is memoryless, by taking the expectation of (A.87) and by comparing (A.84) and (A.85), inequality (A.56) thus holds with equality. By the packet movement rule in Component 5, we also have

$$\sum_{S_O \in 2^{S_I \cup S_X}} Y^{[S_X]}_{k;S_I \to S_O}(\tau) = 1_{\{T^*(\tau)=S_I \cup \{k\}\}} \cdot 1_{\{S_{\text{rx}}(\tau)=S_X\}}. \tag{A.88}$$

Again by the memorylessness of the channel and by comparing (A.84) and (A.86), inequality (A.57) holds with equality.

To prove (A.55), we notice that by (A.61), we can bound $\Delta Q_S^{(k)}$ by the queue length difference

$$\Delta Q_S^{(k)}(t) \le Q_S^{(k)}(t+1) - Q_S^{(k)}(t).$$

Therefore we have

$$\begin{aligned} &\lim_{m\to\infty} \frac{1}{t_m} \sum_{\tau=0}^{t_m-1} \mathsf{E}\left\{\Delta Q_S^{(k)}(\tau)\right\} \\ \le &\lim_{m\to\infty} \frac{1}{t_m} \left(\mathsf{E}\left\{Q_S^{(k)}(t_m)\right\} - \mathsf{E}\left\{Q_S^{(k)}(0)\right\}\right) \\ \le &\lim_{m\to\infty} \frac{D}{t_m} = 0 \end{aligned} \tag{A.89}$$

where (A.89) is by (A.83). By comparing the definition of $\Delta Q_S^{(k)}$ in (A.60) with the definitions of $\vec{y}$ in (A.85) and (A.86), the computed $\vec{y}$ variables must satisfy (A.55). The proof of Proposition 2.4.3 is complete.

A.7 Proof of Lemma 3

We prove Lemma 3 by showing that Lemma 3 holds for each iteration of $(k_0, \mathrm{i}_0) \in \mathcal{N}_0$ in Algorithm 1. Since changing the order of columns for all the matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ is equivalent to changing the labels of arrival packets, Lemma 3 holds after moving the column of index (k_0, i_0) to the left in Line 7 of Algorithm 1.

Recall that $\mathcal{N}_0$ is the collection of the old (k, i) that no longer appears in any of the headers in the queue; and $\mathbf{y}(t)$ is the linear combination of row vectors in $\mathbf{U}^{(k)}$, $k \in [K]$, where $\mathbf{U}^{(k)}$ is the matrix derived from the columns that still in the $\mathbf{V}^{(k)}$ being considered. Since after the column swap in Line 7 the first column is $(k_0, \mathrm{i}_0) \in \mathcal{N}_0$, we can represent $\mathbf{y}(t) = \begin{bmatrix} 0 & \tilde{\mathbf{y}}(t) \end{bmatrix}$. If the first column (k_0, i_0) is also a zero column in $\mathbf{V}^{(k)}$ then it is clear that Lemma 3 holds after removing (k_0, i_0) in both $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$. On the other hand, if the first column is not a zero column, we apply Gaussian elimination to $\mathbf{V}^{(k)}$, which does not alter the row space $\langle \mathbf{V}^{(k)} \rangle$. We can then write the resulting $\tilde{\mathbf{V}}^{(k)}$ and the corresponding $\tilde{\mathbf{U}}^{(k)}$ by

$$\mathbf{U}^{(k)} = \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{U}}^{(k)} \end{bmatrix} \text{ and } \mathbf{V}^{(k)} = \begin{bmatrix} 1 & \mathbf{v} \\ \mathbf{0} & \tilde{\mathbf{V}}^{(k)} \end{bmatrix}.$$

Therefore $\mathbf{y}(t) \in \langle \mathbf{U}^{(k)} \rangle \oplus \langle \mathbf{V}^{(k)} \rangle$ if and only if there exists a coding vector $\mathbf{c} \triangleq [\mathbf{c}_1, \gamma, \mathbf{c}_2]$ such that

$$\begin{aligned} \mathbf{y}(t) = \begin{bmatrix} 0 & \tilde{\mathbf{y}}(t) \end{bmatrix} = \mathbf{c} \begin{bmatrix} \mathbf{U}^{(k)} \\ \mathbf{V}^{(k)} \end{bmatrix} &= \begin{bmatrix} \mathbf{c}_1 & \gamma & \mathbf{c}_2 \end{bmatrix} \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{U}}^{(k)} \\ 1 & \mathbf{v} \\ \mathbf{0} & \tilde{\mathbf{V}}^{(k)} \end{bmatrix} \\ &= \begin{bmatrix} \gamma & \mathbf{c}_1 \tilde{\mathbf{U}}^{(k)} + \gamma \mathbf{v} + \mathbf{c}_2 \tilde{\mathbf{V}}^{(k)} \end{bmatrix}. \end{aligned}$$

Examining the above equality then leads to $\gamma = 0$ and

$$\tilde{\mathbf{y}}(t) = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}}^{(k)} \\ \tilde{\mathbf{V}}^{(k)} \end{bmatrix} \in \langle \tilde{\mathbf{U}}^{(k)} \rangle \oplus \langle \tilde{\mathbf{V}}^{(k)} \rangle. \tag{A.90}$$

Therefore Lemma 3 also holds after removing the first row and first column of $\mathbf{V}^{(k)}$.

Let $\mathbf{V}_{\text{ref}}^{(k)}$ be the row-echelon form of $\mathbf{V}^{(k)}$ after removing all the zero rows. Since $\langle \mathbf{V}_{\text{ref}}^{(k)} \rangle = \langle \mathbf{V}^{(k)} \rangle$, Lemma 3 holds after Line 14 of Algorithm 1. The proof of Lemma 3 is thus complete.

B. SUPPLEMENTARY MATERIALS FOR CHAPTER 3

B.1 Proof of Proposition 3.3.2

Section 3.3.2 shows that any achievable 4-dimensional rate vector $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$ must satisfy the 28 inequalities for (O-1) to (IV-6), which can be succinctly summarized into the following two groups.

Group A: Bounds of a single variable, which combine Instances 0 to 2.

$$R_{(1,1)} \geq a_1 \triangleq \max(0, F_1 - M_1, F_1 - M_2) \tag{A1}$$

$$R_{(1,2)} \geq a_2 \triangleq \max(0, F_1 - M_1, F_2 - M_2, F_1 + F_2 - M_1 - M_2) \tag{A2}$$

$$R_{(2,1)} \geq a_3 \triangleq \max(0, F_2 - M_1, F_1 - M_2, F_1 + F_2 - M_1 - M_2) \tag{A3}$$

$$R_{(2,2)} \geq a_4 \triangleq \max(0, F_2 - M_1, F_2 - M_2) \tag{A4}$$

Group B: Bounds of two variables, which combine Instances 3 and 4.

$$R_{(1,1)} + R_{(1,2)} \geq b_1 \triangleq \max(F_1 + F_2 - M_2, \\ 2F_1 + F_2 - M_1 - M_2) \tag{B1}$$

$$R_{(1,1)} + R_{(2,1)} \geq b_2 \triangleq \max(F_1 + F_2 - M_1, \\ 2F_1 + F_2 - M_1 - M_2) \tag{B2}$$

$$R_{(1,1)} + R_{(2,2)} \geq b_3 \triangleq \max(F_1 + F_2 - M_1, \\ F_1 + F_2 - M_2) \tag{B3}$$

$$R_{(1,2)} + R_{(2,1)} \geq b_4 \triangleq \max(F_1 + F_2 - M_1, F_1 + F_2 - M_2, \\ 2F_1 + F_2 - M_1 - M_2, F_1 + 2F_2 - M_1 - M_2) \tag{B4}$$

$$R_{(1,2)} + R_{(2,2)} \geq b_5 \triangleq \max(F_1 + F_2 - M_1, \\ F_1 + 2F_2 - M_1 - M_2) \tag{B5}$$

$$R_{(2,1)} + R_{(2,2)} \geq b_6 \triangleq \max(F_1 + F_2 - M_2, \\ F_1 + 2F_2 - M_1 - M_2) \tag{B6}$$

Note that the values a_1 to a_4 and b_1 to b_6 are computed by evaluating the max operations in (A1) to (B6). For example, if $M_2 < M_1 < F_1$, then $a_1 = F_1 - M_2$ in (A1). However, if $F_1 < M_2 < M_1$, then $a_1 = 0$ in (A1). The key observation is that once we fix the (F_1, F_2, M_1, M_2) value, the 28 linear inequalities immediately collapse to 10 linear inequalities.

We now discuss some perquisite of the detailed proof.

Tight inequalities: There are 10 inequalities in (A1) to (B6). Each corner point in this 4-dimensional polytope must satisfy at least 4 of them with equalities and sometimes more. If an inequality is satisfied with equality, we say such an inequality is *tight*. Therefore, we need to have at least 4 tight inequalities. One main contribution of this proof is to analyze the relationship among these 10 inequalities for any arbitrary (F_1, F_2, M_1, F_2) so that we do not need to exhaustively examining all $\binom{10}{4}$ combinations for every (F_1, F_2, M_1, F_2). For simplicity we use the notation $\overline{(\cdot)}$ to represent an inequality being tight. For example, $\overline{\text{(A1)}}$

represents (A1) being tight. Another example is that the four equalities $\overline{\text{(A1)}}$, $\overline{\text{(A3)}}$, $\overline{\text{(B1)}}$, and $\overline{\text{(B6)}}$ jointly imply $R_{(1,1)} = a_1$, $R_{(1,2)} = b_1 - a_1$, $R_{(2,1)} = a_3$, and $R_{(2,2)} = b_6 - a_3$.

Global conditions: Without loss of generality, we assume implicitly the following conditions throughout Appendix B.1.

$$M_2 \geq 0 \qquad \text{(G1)} \qquad\qquad F_2 \geq 0 \qquad \text{(G2)}$$

$$M_1 \geq M_2 \qquad \text{(G3)} \qquad\qquad F_1 \geq F_2 \qquad \text{(G4)}$$

$$M_1 \leq F_1 + F_2 \qquad \text{(G5)}$$

These technical assumptions are without loss of generality. Specifically, (G1) and (G2) ensure non-negativity; (G3) and (G4) always hold after swapping the user and file indices; and (G5) holds since there is no need to store more than the total file size $F_1 + F_2$. In the future, we refer these 5 inequalities as the global conditions $\mathcal{G}$:

$$\mathcal{G} \triangleq \{\text{(G1)}, \text{(G2)}, \text{(G3)}, \text{(G4)}, \text{(G5)}\}$$

Additional notation: For any set of (linear) inequalities $\mathcal{A}$, we use $\vec{\mathcal{R}}_{\mathcal{A}}$ to denote the set of $\vec{R}$ vectors that satisfy simultaneously *all* inequalities of $\mathcal{A}$. For any two sets of inequalities $\mathcal{A}$ and $\mathcal{B}$, we say $\mathcal{A}$ *implies* $\mathcal{B}$ if $\vec{\mathcal{R}}_{\mathcal{A}} \subseteq \vec{\mathcal{R}}_{\mathcal{B}}$. We use $\mathcal{A} \Rightarrow \mathcal{B}$ as shorthand.

We say the two sets of inequalities $\mathcal{A}$ and $\mathcal{B}$ are *equivalent*, denoted by $\mathcal{A} \Leftrightarrow \mathcal{B}$, if $\mathcal{A} \Rightarrow \mathcal{B}$ and $\mathcal{B} \Rightarrow \mathcal{A}$. Sometimes the equivalence and implication relationships hold only under some additional conditions $\mathcal{C}$. To that end, we use

$$\mathcal{A} \overset{\mathcal{C}}{\Rightarrow} \mathcal{B}$$

to represent $\mathcal{A}$ implies $\mathcal{B}$ under conditions[1] $\mathcal{C}$. Similarly, the notation $\mathcal{A} \overset{\mathcal{C}}{\Leftrightarrow} \mathcal{B}$ represents conditional equivalence under $\mathcal{C}$.

[1]↑A more rigorous notation of conditional implication should be $(\mathcal{A} \cup \mathcal{C}) \Rightarrow \mathcal{B}$. However, by writing $\mathcal{A} \overset{\mathcal{C}}{\Rightarrow} \mathcal{B}$ it is clearer what are the inequalities of interest (i.e., $\mathcal{A}$ and $\mathcal{B}$) and what are extra conditions being considered (i.e., $\mathcal{C}$).

Case 1: We assume

$$M_1 \leq F_2. \tag{c1}$$

Ineq. (c1) and $\mathcal{G}$ jointly describe the scenario when the (M_1, M_2) value falls into the lower-left triangle in Fig. 3.2 with solid edges and being marked as "Case 1". In this case, the a_1 to b_6 values of (A1) to (B6) become

$$\begin{aligned} &a_1 = F_1 - M_2, \quad a_2 = a_3 = F_1 + F_2 - M_1 - M_2, \\ &a_4 = F_2 - M_2, \quad b_1 = b_2 = b_4 = 2F_1 + F_2 - M_1 - M_2, \\ &b_3 = F_1 + F_2 - M_2, \quad b_5 = b_6 = F_1 + 2F_2 - M_1 - M_2. \end{aligned} \tag{c1.ab}$$

Using our previous notation and the definition of a_1 to b_6, the above statement can be summarized as $\{(\text{c1})\} \cup \mathcal{G} \Rightarrow \{(\text{c1.ab})\}$. We now further divide this case into two sub-cases.
Case 1.1: We assume

$$M_1 + M_2 \leq F_2 \tag{c1.1}$$

and **Case 1.2:** We assume

$$M_1 + M_2 > F_2. \tag{c1.2}$$

Cases 1.1 and 1.2 further divide the solid lower-left triangle of Fig. 3.2 by a dotted line. In the following we focus on Case 1.1, the left sub-triangle.

Case 1.1: We consider the following 5 subcases.

Case 1.1.1 (A1) is tight. i.e., $\overline{(\text{A1})}$ holds. Under conditions $\mathcal{G}$, (c1) and (c1.1), we can prove the following relationship

$$\begin{aligned} &\{\overline{(\text{A1})}, (\text{B1}), (\text{B2}), (\text{B3})\} \overset{\mathcal{G},(\text{c1}),(\text{c1.1})}{\Longrightarrow} \\ &\{(\text{A2}), (\text{A3}), (\text{A4}), (\text{B4}), (\text{B5}), (\text{B6})\}. \end{aligned} \tag{B.1}$$

The above relationship is derived by first noting $\{(\text{c1})\} \cup \mathcal{G} \Rightarrow \{(\text{c1.ab})\}$ and by the following intermediate steps

$$\{\overline{(\text{A1})}, (\text{B1})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A2}) \tag{B.2}$$

$$\{\overline{(\text{A1})}, (\text{B2})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A3}) \tag{B.3}$$

$$\{\overline{(\text{A1})}, (\text{B3})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A4}) \tag{B.4}$$

$$\{\overline{(\text{A1})}, (\text{B1}), (\text{B2})\} \overset{(\text{c1.ab}),(\text{G1}),(\text{c1})}{\Longrightarrow} (\text{B4}) \tag{B.5}$$

$$\{\overline{(\text{A1})}, (\text{B1}), (\text{B3})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{B5}) \tag{B.6}$$

$$\{\overline{(\text{A1})}, (\text{B2}), (\text{B3})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{B6}). \tag{B.7}$$

Each intermediate step can be verified by straightforward algebraic operations. For example, part of (c1.ab) ensures that $a_1 = F_1 - M_2$, $a_2 = F_1 + F_2 - M_1 - M_2$, and $b_1 = 2F_1 + F_2 - M_1 - M_2$. Under these a_1, a_2, and b_1 values, $\overline{(\text{A1})}$, (A2), and (B1) become

$$R_{(1,1)} = F_1 - M_2 \tag{B.8}$$

$$R_{(1,2)} \geq F_1 + F_2 - M_1 - M_2 \tag{B.9}$$

$$R_{(1,1)} + R_{(1,2)} \geq 2F_1 + F_2 - M_1 - M_2 \tag{B.10}$$

Subtracting $\overline{(\text{A1})}$ (i.e., (B.8)) from (B1) (i.e., (B.10)), we have $R_{(1,2)} \geq F_1 + F_2 - M_1$ which implies (A2) (i.e., (B.9)) under condition (G1). We thus prove the intermediate step (B.2).

Similarly, (c1.ab) implies that (B2), and (B4) become

$$R_{(1,1)} + R_{(2,1)} \geq 2F_1 + F_2 - M_1 - M_2 \tag{B.11}$$

$$R_{(1,2)} + R_{(2,1)} \geq 2F_1 + F_2 - M_1 - M_2 \tag{B.12}$$

Adding up (B1) and (B2) (i.e., (B.10) and (B.11)) and subtracting $\overline{(\text{A1})}$ (i.e., (B.8)) twice, we have $R_{(1,2)} + R_{(2,1)} \geq 2F_1 + 2F_2 - 2M_1$ which implies (B4) (i.e., (B.12)), provided both (G1) and (c1) hold simultaneously. We have thus proven the intermediate step (B.5). Since

the proofs of other intermediate steps (B.3), (B.4), (B.6), and (B.7) are very similar and straightforward, we omit their details.

By (B.1), the four *tight* linear inequalities in Case-1.1.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (B1), (B2), and (B3). Solving these equations, the corresponding corner point is *Vertex 1* $(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$ listed in Table 3.3.

Case 1.1.2: (A2) is tight. i.e., $\overline{\text{(A2)}}$ holds. We can then prove the following relationship

$$\{\overline{\text{(A2)}}, \text{(A3)}, \text{(B1)}, \text{(B5)}\} \overset{\mathcal{G},\text{(c1)},\text{(c1.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A4)}, \text{(B2)}, \text{(B3)}, \text{(B4)}, \text{(B6)}\}. \tag{B.13}$$

The above relationship is derived by $\{\text{(c1)}\} \cup \mathcal{G} \Rightarrow \{\text{(c1.ab)}\}$ and by the following intermediate steps

$$\{\overline{\text{(A2)}}, \text{(B1)}\} \overset{\text{(c1.ab)},\text{(G1)}}{\Longrightarrow} \text{(A1)} \tag{B.14}$$

$$\{\overline{\text{(A2)}}, \text{(A3)}\} \overset{\text{(c1.ab)},\text{(c1.1)}}{\Longrightarrow} \text{(B4)} \tag{B.15}$$

$$\{\overline{\text{(A2)}}, \text{(B5)}\} \overset{\text{(c1.ab)},\text{(G1)}}{\Longrightarrow} \text{(A4)} \tag{B.16}$$

$$\{\overline{\text{(A2)}}, \text{(A3)}, \text{(B1)}\} \overset{\text{(c1.ab)}}{\Longrightarrow} \text{(B2)} \tag{B.17}$$

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B5)}\} \overset{\text{(c1.ab)},\text{(G1)}}{\Longrightarrow} \text{(B3)} \tag{B.18}$$

$$\{\overline{\text{(A2)}}, \text{(A3)}, \text{(B5)}\} \overset{\text{(c1.ab)}}{\Longrightarrow} \text{(B6)}. \tag{B.19}$$

We omit the detailed proofs of the intermediate steps as they are extremely similar to the two examples discussed in the proof of Case 1.1.1.

By (B.13), the four tight linear inequalities in Case-1.1.2 can only be (A2) (thus $\overline{\text{(A2)}}$), (A3), (B1), and (B5). Solving these equations, the corresponding corner point is *Vertex 2* $(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$ listed in Table 3.3.

Case 1.1.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can then prove the following relationship

$$\{\overline{\text{(A3)}}, \text{(A2)}, \text{(B2)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c1)},\text{(c1.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A4)}, \text{(B1)}, \text{(B3)}, \text{(B4)}, \text{(B5)}\} \tag{B.20}$$

by the following straightforward intermediate steps

$$\{\overline{(A3)}, (B2)\} \overset{(c1.ab),(G1)}{\Longrightarrow} (A1) \tag{B.21}$$

$$\{\overline{(A3)}, (A2)\} \overset{(c1.ab),(c1.1)}{\Longrightarrow} (B4) \tag{B.22}$$

$$\{\overline{(A3)}, (B6)\} \overset{(c1.ab),(G1)}{\Longrightarrow} (A4) \tag{B.23}$$

$$\{\overline{(A3)}, (A2), (B2)\} \overset{(c1.ab)}{\Longrightarrow} (B1) \tag{B.24}$$

$$\{\overline{(A3)}, (B2), (B6)\} \overset{(c1.ab),(G1)}{\Longrightarrow} (B3) \tag{B.25}$$

$$\{\overline{(A3)}, (A2), (B6)\} \overset{(c1.ab)}{\Longrightarrow} (B5). \tag{B.26}$$

By (B.20), the four tight linear inequalities in Case-1.1.3 can only be (A3) (thus $\overline{(A3)}$), (A2), (B2), and (B6). Solving these equations, the corresponding corner point is Vertex 2 $(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$ listed in Table 3.3.

Case 1.1.4: (A4) is tight. i.e., $\overline{(A4)}$ holds. We can then prove the following relationship

$$\{\overline{(A4)}, (B3), (B5), (B6)\} \overset{\mathcal{G},(c1),(c1.1)}{\Longrightarrow} \{(A1), (A2), (A3), (B1), (B2), (B4)\} \tag{B.27}$$

by the following straightforward intermediate steps

$$\{\overline{(A4)}, (B3)\} \overset{(c1.ab),(G1)}{\Longrightarrow} (A1) \tag{B.28}$$

$$\{\overline{(A4)}, (B5)\} \overset{(c1.ab),(G1)}{\Longrightarrow} (A2) \tag{B.29}$$

$$\{\overline{(A4)}, (B6)\} \overset{(c1.ab),(G1)}{\Longrightarrow} (A3) \tag{B.30}$$

$$\{\overline{(A4)}, (B3), (B5)\} \overset{(c1.ab),(G1)}{\Longrightarrow} (B1) \tag{B.31}$$

$$\{\overline{(A4)}, (B3), (B6)\} \overset{(c1.ab),(G1)}{\Longrightarrow} (B2) \tag{B.32}$$

$$\{\overline{(A4)}, (B5), (B6)\} \overset{(c1.ab),(G1),(c1)}{\Longrightarrow} (B4). \tag{B.33}$$

By (B.27), the four tight linear inequalities in Case-1.1.4 can only be (A4) (thus $\overline{(A4)}$), (B3), (B5), and (B6). Solving these equations, the corresponding corner point is *Vertex 3* $(F_1, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2 - M_2)$ listed in Table 3.3.

Case 1.1.5: None of (A1) to (A4) is tight. Recall that in all the discussion of Case 1.1 and its subcases, we assume $\mathcal{G}$, (c1), (c1.1), and (c1.ab). Since

$$\{(\text{A2}), (\text{A3})\} \overset{(\text{c1.ab}),(\text{c1.1})}{\Longrightarrow} (\text{B4}), \tag{B.34}$$

any corner point that is *loose* for all 4 inequalities (A1) to (A4) (and thus being loose for (A2) and (A3)) must also be loose for (B4). Therefore the corner point must be decided by 4 out of the 5 remaining inequalities (B1), (B2), (B3), (B5), and (B6). By (c1.ab), the inequalities corresponding to (B1), (B2), (B5), and (B6) become

$$R_{(1,1)} + R_{(1,2)} \geq b_1 = 2F_1 + F_2 - M_1 - M_2 \tag{B.35}$$

$$R_{(1,1)} + R_{(2,1)} \geq b_2 = 2F_1 + F_2 - M_1 - M_2 \tag{B.36}$$

$$R_{(1,2)} + R_{(2,2)} \geq b_5 = F_1 + 2F_2 - M_1 - M_2 \tag{B.37}$$

$$R_{(2,1)} + R_{(2,2)} \geq b_6 = F_1 + 2F_2 - M_1 - M_2 \tag{B.38}$$

We observe that the tight versions (equalities) of these inequalities are linearly dependent. As a result, any three of them being tight implies the fourth one is also tight. We hereby say that these 4 inequalities are *co-dependent*.

Since (B1), (B2), (B5), and (B6) are co-dependent and since a corner point requires 4 tight linearly *independent* inequalities, all 5 inequalities (B1), (B2), (B3), (B5), and (B6) must be tight simultaneously and jointly they yield exactly one corner point *Vertex 4* $(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$ listed in Table 3.3.

The proof of Case 1.1.5 is completed by further proving that Vertex 4 is a legitimate corner point that satisfies (A1) to (A4) as well. The detailed verification steps are

$$\text{Vertex 4} \overset{(\text{c1.ab}),(\text{G1})}{\Rightarrow} (\text{A1}); \quad \text{Vertex 4} \overset{(\text{c1.ab}),(\text{G1})}{\Rightarrow} (\text{A2}); \tag{B.39}$$

$$\text{Vertex 4} \overset{(\text{c1.ab}),(\text{G1})}{\Rightarrow} (\text{A3}); \quad \text{Vertex 4} \overset{(\text{c1.ab}),(\text{G1})}{\Rightarrow} (\text{A4}). \tag{B.40}$$

Case 1.2: In this case we assume both (c1) and (c1.2) are true. This sub-case is the right sub-triangle above the dotted line in the solid lower-left triangle (Case 1) of Fig. 3.2. We now consider the following 6 subcases of Case 1.2.

Case 1.2.1: (A1) is tight, i.e., $\overline{\text{(A1)}}$ holds. Since the intermediate steps of Case 1.1.1, i.e., (B.2)-(B.7) does not require condition (c1.2), the statement in (B.1) holds even if we swap out (c1.1) by (c1.2). The rest of the analysis is verbatim to Case 1.1.1 and the corner point is also Vertex 1.

Case 1.2.2: (A2) is tight. i.e., $\overline{\text{(A2)}}$ holds. Note that we cannot reuse the derivation in Case 1.1.2 since (B.15) requires (c1.1) being true but in this case we only have (c1.2). That said, we can still prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}, \text{(B5)}\} \overset{\mathcal{G},\text{(c1)},\text{(c1.2)}}{\Longrightarrow}\\ &\{\text{(A1)}, \text{(A3)}, \text{(A4)}, \text{(B2)}, \text{(B3)}, \text{(B6)}\}\end{aligned} \tag{B.41}$$

by reusing (B.14), (B.16), (B.18), and the following straightforward intermediate steps

$$\{\overline{\text{(A2)}}, \text{(B4)}\} \overset{\text{(c1.ab)},\text{(c1.2)}}{\Longrightarrow} \text{(A3)} \tag{B.42}$$

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}\} \overset{\text{(c1.ab)},\text{(c1.2)}}{\Longrightarrow} \text{(B2)} \tag{B.43}$$

$$\{\overline{\text{(A2)}}, \text{(B4)}, \text{(B5)}\} \overset{\text{(c1.ab)},\text{(c1.2)}}{\Longrightarrow} \text{(B6)}. \tag{B.44}$$

By (B.41), the four tight linear inequalities in Case-1.2.2 can only be (A2) (thus $\overline{\text{(A2)}}$), (B1), (B4), and (B5). Solving these equations, the corresponding corner point is Vertex 5 $(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$ listed in Table 3.3.

Case 1.2.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can then prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c1)},\text{(c1.2)}}{\Longrightarrow}\\ &\{\text{(A1)}, \text{(A2)}, \text{(A4)}, \text{(B1)}, \text{(B3)}, \text{(B5)}\}\end{aligned} \tag{B.45}$$

by reusing (B.21), (B.23), (B.25), and the following straightforward intermediate steps

$$\{\overline{(A3)}, (B4)\} \overset{(c1.ab),(c1.2)}{\Longrightarrow} (A2) \tag{B.46}$$

$$\{\overline{(A3)}, (B2), (B4)\} \overset{(c1.ab),(c1.2)}{\Longrightarrow} (B1) \tag{B.47}$$

$$\{\overline{(A3)}, (B4), (B6)\} \overset{(c1.ab),(c1.2)}{\Longrightarrow} (B5). \tag{B.48}$$

By (B.45), the four tight linear inequalities in Case-1.2.3 can only be (A3) (thus $\overline{(A3)}$), (B2), (B4), and (B6). Solving these equations, the corresponding corner point is Vertex 6 $(F_1, F_1, F_1 + F_2 - M_1 - M_2, F_2)$ listed in Table 3.3.

Case 1.2.4: (A4) is tight, i.e., $\overline{(A4)}$ holds. Since the intermediate steps of Case 1.1.4, i.e., (B.28)-(B.33) does not require condition (c1.2), the statement in (B.27) holds even if we swap out (c1.1) by (c1.2). The rest of the analysis is verbatim to Case 1.1.4 and the corner point is also Vertex 3.

Case 1.2.5: None of (A1) to (A4) is tight, but (B4) is tight, i.e., $\overline{(B4)}$ holds. We can prove

$$\{(B1), (B2), (B5), (B6), \overline{(B4)}\} \overset{\mathcal{G},(c1),(c1.2)}{\Rightarrow} \{(B3)\} \tag{B.49}$$

using the following intermediate step

$$\{(B1), (B6), \overline{(B4)}\} \overset{(c1.ab),(c1)}{\Rightarrow} \{(B3)\}. \tag{B.50}$$

The statement (B.49) implies that when none of (A1) to (A4) is tight but $\overline{(B4)}$ holds, the corner point is decided solely by the inequalities (B1), (B2), (B5), and (B6) and we do not need to check whether (B3) is tight or not.

Recall that under the a_1 to b_6 values in (c1.ab), the inequalities corresponding to (B1), (B2), (B5), and (B6) are *co-dependent* as shown in Case 1.1.5. Therefore, the statement (B.49) further implies that the corner point must be tight for all 5 inequalities (B1), (B2), (B5), (B6), $\overline{(B4)}$. Solving these 5 joint equations (four of them are codependent), we obtain

the corner point *Vertex 7* $(F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, F_2 + \frac{F_2-M_1-M_2}{2})$ listed in Table 3.3.

The proof of Case 1.2.5 is completed by further proving that Vertex 7 is a legitimate corner point that satisfies (A1) to (A4) as well. The detailed verification steps are

$$\text{Vertex 7} \overset{\text{(c1.ab)},\text{(G1)},\text{(c1)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 7} \overset{\text{(c1.ab)},\text{(c1.2)}}{\Rightarrow} \text{(A2)}; \tag{B.51}$$

$$\text{Vertex 7} \overset{\text{(c1.ab)},\text{(c1.2)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 7} \overset{\text{(c1.ab)},\text{(G1)},\text{(c1)}}{\Rightarrow} \text{(A4)}. \tag{B.52}$$

Case 1.2.6: None of (A1) to (A4) is tight, nor is (B4). If we retrace the proof of Case 1.1.5, we notice that (B.34) ensures that when in Case 1.1.5, we always have (B4) being loose. Since (B.34) holds only under (c1.1), ineq. (B4) can be tight or loose in Case 1.2. That is why in Case 1.2.5, we discussed the case when (B4) is tight and in this case we assume (B4) is loose. Since the arguments in Case 1.1.5 after (B.34) no longer uses the condition (c1.1), we can use the same argument verbatim and prove that the corner point in Case 1.2.6 is the Vertex 4 $(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$ listed in Table 3.3.

Case 2: We assume

$$M_2 \le F_2 < M_1 \le F_1. \tag{c2}$$

Ineq. (c2) and $\mathcal{G}$ jointly describe the scenario when the (M_1, M_2) value falls into the lower-mid rectangle in Fig. 3.2 with solid edges and being marked as "Case 2". In this case, the a_1 to b_6 values of (A1) to (B6) become

$$\begin{aligned} &a_1 = a_3 = F_1 - M_2, \quad a_2 = F_1 + F_2 - M_1 - M_2, \\ &a_4 = F_2 - M_2, \quad b_1 = b_2 = b_4 = 2F_1 + F_2 - M_1 - M_2, \\ &b_3 = b_6 = F_1 + F_2 - M_2, \quad b_5 = F_1 + 2F_2 - M_1 - M_2. \end{aligned} \tag{c2.ab}$$

Note that only a_3 and b_6 are different in (c1.ab) and (c2.ab), and the rest 8 values are identical. Therefore we can reuse the equations related to (c1.ab) when (A3) and (B6) are not involved. We now further divide this case into two sub-cases. **Case 2.1:** We assume

$$F_2 + M_2 < M_1 \tag{c2.1}$$

and **Case 2.2:** We assume

$$F_2 + M_2 \geq M_1. \tag{c2.2}$$

Cases 2.1 and 2.2 further divide the solid lower-mid rectangle of Fig. 3.2 by a 45-degree line. In the following we focus on Case 2.1, the right subregion below the dotted line.

Case 2.1: We consider the following 5 subcases.

Case 2.1.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A1)}}, \text{(A3)}, \text{(B1)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c2)},\text{(c2.1)}}{\Longrightarrow} \\ &\{\text{(A2)}, \text{(A4)}, \text{(B2)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\}\end{aligned} \tag{B.53}$$

by reusing (B.2), (B.4), (B.6), and the following straightforward intermediate steps

$$\{\overline{\text{(A1)}}, \text{(A3)}\} \overset{\text{(c2.ab)},\text{(c2.1)}}{\Longrightarrow} \text{(B2)} \tag{B.54}$$

$$\{\overline{\text{(A1)}}, \text{(A3)}, \text{(B1)}\} \overset{\text{(c2.ab)}}{\Longrightarrow} \text{(B4)} \tag{B.55}$$

$$\{\overline{\text{(A1)}}, \text{(A3)}, \text{(B3)}\} \overset{\text{(c2.ab)}}{\Longrightarrow} \text{(B6)}. \tag{B.56}$$

By (B.53), the four tight linear inequalities in Case-2.1.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (A3), (B1), and (B3). Solving these equations, the corresponding corner point is *Vertex 8* $(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 2.1.2: (A2) is tight. i.e., $\overline{\text{(A2)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}, \text{(B5)}\} \overset{\mathcal{G},\text{(c2)},\text{(c2.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A3)}, \text{(A4)}, \text{(B2)}, \text{(B3)}, \text{(B6)}\} \tag{B.57}$$

by reusing (B.14), (B.16), (B.18), and the following straightforward intermediate steps

$$\{\overline{\text{(A2)}}, \text{(B4)}\} \overset{\text{(c2.ab)},\text{(G1)}}{\Longrightarrow} \text{(A3)} \tag{B.58}$$

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}\} \overset{\text{(c2.ab)},\text{(G1)},\text{(c2)}}{\Longrightarrow} \text{(B2)} \tag{B.59}$$

$$\{\overline{\text{(A2)}}, \text{(B4)}, \text{(B5)}\} \overset{\text{(c2.ab)},\text{(G1)}}{\Longrightarrow} \text{(B6)}. \tag{B.60}$$

By (B.57), the four tight linear inequalities in Case-2.1.2 can only be (A2) (thus $\overline{\text{(A2)}}$), (B1), (B4), and (B5). Solving these equations, the corresponding corner point is *Vertex 9* $(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$ listed in Table 3.3.

Case 2.1.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A3)}}, \text{(A1)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c2)},\text{(c2.1)}}{\Longrightarrow} \{\text{(A2)}, \text{(A4)}, \text{(B1)}, \text{(B2)}, \text{(B3)}, \text{(B5)}\} \tag{B.61}$$

by the following straightforward intermediate steps

$$\{\overline{\text{(A3)}}, \text{(A1)}\} \overset{\text{(c2.ab)},\text{(c2.1)}}{\Longrightarrow} \text{(B2)} \tag{B.62}$$

$$\{\overline{\text{(A3)}}, \text{(B4)}\} \overset{\text{(c2.ab)},\text{(G1)}}{\Longrightarrow} \text{(A2)} \tag{B.63}$$

$$\{\overline{\text{(A3)}}, \text{(B6)}\} \overset{\text{(c2.ab)},\text{(G1)}}{\Longrightarrow} \text{(A4)} \tag{B.64}$$

$$\{\overline{\text{(A3)}}, \text{(A1)}, \text{(B4)}\} \overset{\text{(c2.ab)}}{\Longrightarrow} \text{(B1)} \tag{B.65}$$

$$\{\overline{\text{(A3)}}, \text{(A1)}, \text{(B6)}\} \overset{\text{(c2.ab)}}{\Longrightarrow} \text{(B3)} \tag{B.66}$$

$$\{\overline{\text{(A3)}}, \text{(B4)}, \text{(B6)}\} \overset{\text{(c2.ab)},\text{(G1)}}{\Longrightarrow} \text{(B5)}. \tag{B.67}$$

By (B.61), the four tight linear inequalities in Case-2.1.3 can only be (A3) (thus $\overline{\text{(A3)}}$), (A1), (B4), and (B6). Solving these equations, the corresponding corner point is Vertex 8 $(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 2.1.4: (A4) is tight. i.e., $\overline{\text{(A4)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B5)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c2)},\text{(c2.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A2)}, \text{(A3)}, \text{(B1)}, \text{(B2)}, \text{(B4)}\} \tag{B.68}$$

by reusing (B.28), (B.29), (B.31), and the following straightforward intermediate steps

$$\{\overline{\text{(A4)}}, \text{(B6)}\} \overset{\text{(c2.ab)},\text{(G1)}}{\Longrightarrow} \text{(A3)} \tag{B.69}$$

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B6)}\} \overset{\text{(c2.ab)},\text{(G1)},\text{(c2)}}{\Longrightarrow} \text{(B2)} \tag{B.70}$$

$$\{\overline{\text{(A4)}}, \text{(B5)}, \text{(B6)}\} \overset{\text{(c2.ab)},\text{(G1)}}{\Longrightarrow} \text{(B4)}. \tag{B.71}$$

By (B.68), the four tight linear inequalities in Case-2.1.4 can only be (A4) (thus $\overline{\text{(A4)}}$), (B3), (B5), and (B6). Solving these equations, the corresponding corner point is *Vertex 10* $(F_1, F_1 + F_2 - M_1, F_1, F_2 - M_2)$ listed in Table 3.3.

Case 2.1.5: None of (A1) to (A4) is tight. Recall that in all the discussion of Case 2.1 and its subcases, we assume $\mathcal{G}$, (c2), (c2.1), and (c2.ab). Since

$$\{\text{(A1)}, \text{(A3)}\} \overset{\text{(c2.ab)},\text{(c2.1)}}{\Longrightarrow} \text{(B2)}, \tag{B.72}$$

any corner point that is *loose* for all 4 inequalities (A1) to (A4) (and thus being loose for (A1) and (A3)) must also be loose for (B2). Therefore the corner point must be decided by 4 out of the 5 remaining inequalities (B1), (B3), (B4), (B5), and (B6). By (c2.ab), the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent* such that a corner point requires all 5 inequalities (B1), (B3), (B4), (B5), and (B6) to be tight simultaneously. The 5 inequalities jointly yield exactly one corner point *Vertex 11* $(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 -$

$\frac{M_2}{2}, F_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2}$) listed in Table 3.3. Vertex 11 is a legitimate corner point since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 11} \overset{\text{(c2.ab)},\text{(G1)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 11} \overset{\text{(c2.ab)},\text{(G1)}}{\Rightarrow} \text{(A2)};$$
$$\text{Vertex 11} \overset{\text{(c2.ab)},\text{(G1)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 11} \overset{\text{(c2.ab)},\text{(G1)}}{\Rightarrow} \text{(A4)}.$$

Case 2.2: In this case we assume both (c2) and (c2.2) are true. This sub-case is the left subregion above the dotted line in the solid lower-mid rectangle (Case 2) of Fig. 3.2. We now consider the following 6 subcases of Case 2.2.

Case 2.2.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c2)},\text{(c2.2)}}{\Longrightarrow}$$
$$\{\text{(A2)}, \text{(A3)}, \text{(A4)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\} \tag{B.73}$$

by reusing (B.2), (B.4), (B.6), and the following straightforward intermediate steps

$$\{\overline{\text{(A1)}}, \text{(B2)}\} \overset{\text{(c2.ab)},\text{(c2.2)}}{\Longrightarrow} \text{(A3)} \tag{B.74}$$
$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}\} \overset{\text{(c2.ab)},\text{(c2.2)}}{\Longrightarrow} \text{(B4)}. \tag{B.75}$$
$$\{\overline{\text{(A1)}}, \text{(B2)}, \text{(B3)}\} \overset{\text{(c2.ab)},\text{(c2.2)}}{\Longrightarrow} \text{(B6)}. \tag{B.76}$$

By (B.73), the four tight linear inequalities in Case-2.2.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (B1), (B2), and (B3). Solving these equations, the corresponding corner point is Vertex 1 ($F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2$) listed in Table 3.3.

Case 2.2.2: (A2) is tight, i.e., $\overline{\text{(A2)}}$ holds. Since the intermediate steps of Case 2.1.2 does not require condition (c2.2), the statement in (B.57) holds even if we swap out (c2.1) by (c2.2). The rest of the analysis is verbatim to Case 2.1.2 and the corner point is also Vertex 9.

Case 2.2.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c2)},\text{(c2.2)}}{\Longrightarrow} \{\text{(A1)}, \text{(A2)}, \text{(A4)}, \text{(B1)}, \text{(B3)}, \text{(B5)}\} \tag{B.77}$$

by reusing (B.63), (B.64), (B.67), and the following straightforward intermediate steps

$$\{\overline{\text{(A3)}}, \text{(B2)}\} \overset{\text{(c2.ab)},\text{(c2.2)}}{\Longrightarrow} \text{(A1)} \tag{B.78}$$

$$\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B4)}\} \overset{\text{(c2.ab)},\text{(c2.2)}}{\Longrightarrow} \text{(B1)} \tag{B.79}$$

$$\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B6)}\} \overset{\text{(c2.ab)},\text{(c2.2)}}{\Longrightarrow} \text{(B3)} \tag{B.80}$$

By (B.77), the four tight linear inequalities in Case-2.2.3 can only be (A3) (thus $\overline{\text{(A3)}}$), (B2), (B4), and (B6). Solving these equations, the corresponding corner point is *Vertex 12* $(F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 2.2.4: (A4) is tight, i.e., $\overline{\text{(A4)}}$ holds. Since the intermediate steps of Case 2.1.4 does not require condition (c2.2), the statement in (B.68) holds even if we swap out (c2.1) by (c2.2). The rest of the analysis is verbatim to Case 2.1.4 and the corner point is also Vertex 10.

Case 2.2.5: None of (A1) to (A4) is tight, but (B2) is tight, i.e., $\overline{\text{(B2)}}$ holds. We can prove

$$\{\text{(B1)}, \overline{\text{(B2)}}, \text{(B3)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c2)},\text{(c2.2)}}{\Rightarrow} \{\text{(B5)}\} \tag{B.81}$$

using the following intermediate step

$$\{\text{(B1)}, \text{(B6)}, \overline{\text{(B2)}}\} \overset{\text{(c2.ab)},\text{(c2)}}{\Rightarrow} \{\text{(B5)}\}. \tag{B.82}$$

That is, in this case, we do not need to check whether (B5) is tight or not. Recall that under the a_1 to b_6 values in (c2.ab), the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent* as shown in Case 2.1.5. Therefore, the corner point must be tight for all 5

inequalities (B1), $\overline{\text{(B2)}}$, (B3), (B4), (B6). Solving these 5 joint equations (four of them are codependent), we obtain the corner point *Vertex 13* ($F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2}$) listed in Table 3.3. Vertex 13 is a legitimate corner point since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 13} \overset{\text{(c2.ab)},\text{(c2.2)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 13} \overset{\text{(c2.ab)},\text{(G1)},\text{(c2)}}{\Rightarrow} \text{(A2)};$$
$$\text{Vertex 13} \overset{\text{(c2.ab)},\text{(c2.2)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 13} \overset{\text{(c2.ab)},\text{(G1)},\text{(c2)}}{\Rightarrow} \text{(A4)}.$$

Case 2.2.6: None of (A1) to (A4) is tight, nor is (B2). Since the arguments in Case 2.1.5 after (B.72) no longer uses the condition (c2.1), we can use the same argument verbatim and prove that the corner point in Case 2.2.6 is the Vertex 11 ($F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2}$) listed in Table 3.3.

Case 3: We assume

$$M_2 \le F_2 \le F_1 < M_1. \tag{c3}$$

Ineq. (c3) and $\mathcal{G}$ jointly describe the scenario when the (M_1, M_2) value falls into the lower-right square in Fig. 3.2 with solid edges and being marked as "Case 3". In this case, the a_1 to b_6 values of (A1) to (B6) become

$$\begin{aligned} &a_1 = a_3 = F_1 - M_2, \quad a_2 = a_4 = F_2 - M_2, \\ &b_1 = b_3 = b_4 = b_6 = F_1 + F_2 - M_2, \\ &b_2 = 2F_1 + F_2 - M_1 - M_2, \quad b_5 = F_1 + 2F_2 - M_1 - M_2. \end{aligned} \tag{c3.ab}$$

Note that only a_2, b_1, and b_4 are different in (c2.ab) and (c3.ab), and the rest 7 values are identical. Therefore we can reuse the equations related to (c2.ab) (or (c1.ab) referred in Case 2) when (A2), (B1) and (B4) are not involved. We now further divide this case into three sub-cases. **Case 3.1:** We assume

$$F_1 + M_2 < M_1, \tag{c3.1}$$

Case 3.2: We assume

$$F_2 + M_2 < M_1 \leq F_1 + M_2, \tag{c3.2}$$

and **Case 3.3:** We assume

$$M_1 \leq F_2 + M_2. \tag{c3.3}$$

Cases 3.1 and 3.2 further divide the solid lower-right square of Fig. 3.2a by a 45-degree line. Cases 3.1, 3.2, and 3.3 further divide the the solid lower-right square of Fig. 3.2b by two 45-degree lines. In the following we focus on Case 3.1, the rightmost subregion below the dotted line.

Case 3.1: We consider the following 5 subcases.

Case 3.1.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A1)}}, \text{(A3)}, \text{(B1)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c3)},\text{(c3.1)}}{\Longrightarrow} \\ &\{\text{(A2)}, \text{(A4)}, \text{(B2)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\}\end{aligned} \tag{B.83}$$

by reusing (B.4), (B.56), and the following straightforward intermediate steps

$$\{\overline{\text{(A1)}}, \text{(B1)}\} \overset{\text{(c3.ab)},\text{(G1)}}{\Longrightarrow} \text{(A2)} \tag{B.84}$$

$$\{\overline{\text{(A1)}}, \text{(A3)}\} \overset{\text{(c3.ab)},\text{(G4)},\text{(c3.1)}}{\Longrightarrow} \text{(B2)} \tag{B.85}$$

$$\{\overline{\text{(A1)}}, \text{(A3)}, \text{(B1)}\} \overset{\text{(c3.ab)}}{\Longrightarrow} \text{(B4)} \tag{B.86}$$

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B3)}\} \overset{\text{(c3.ab)},\text{(G1)},\text{(c3)}}{\Longrightarrow} \text{(B5)} \tag{B.87}$$

By (B.83), the four tight linear inequalities in Case-3.1.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (A3), (B1), and (B3). Solving these equations, the corresponding corner point is *Vertex 14* $(F_1 - M_2, F_2, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 3.1.2: (A2) is tight. i.e., $\overline{\text{(A2)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A2)}}, \text{(A4)}, \text{(B1)}, \text{(B4)}\} \overset{\mathcal{G},\text{(c3)},\text{(c3.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A3)}, \text{(B2)}, \text{(B3)}, \text{(B5)}, \text{(B6)}\} \tag{B.88}$$

by the following straightforward intermediate steps

$$\{\overline{\text{(A2)}}, \text{(B1)}\} \overset{\text{(c3.ab)},\text{(G1)}}{\Longrightarrow} \text{(A1)} \tag{B.89}$$

$$\{\overline{\text{(A2)}}, \text{(B4)}\} \overset{\text{(c3.ab)},\text{(G1)}}{\Longrightarrow} \text{(A3)} \tag{B.90}$$

$$\{\overline{\text{(A2)}}, \text{(A4)}\} \overset{\text{(c3.ab)},\text{(c3.1)}}{\Longrightarrow} \text{(B5)} \tag{B.91}$$

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}\} \overset{\text{(c3.ab)},\text{(G1)},\text{(c3)}}{\Longrightarrow} \text{(B2)} \tag{B.92}$$

$$\{\overline{\text{(A2)}}, \text{(A4)}, \text{(B1)}\} \overset{\text{(c3.ab)}}{\Longrightarrow} \text{(B3)} \tag{B.93}$$

$$\{\overline{\text{(A2)}}, \text{(A4)}, \text{(B4)}\} \overset{\text{(c3.ab)}}{\Longrightarrow} \text{(B6)}. \tag{B.94}$$

By (B.88), the four tight linear inequalities in Case-3.1.2 can only be (A2) (thus $\overline{\text{(A2)}}$), (A2), (B1), and (B4). Solving these equations, the corresponding corner point is *Vertex 15* $(F_1, F_2 - M_2, F_1, F_2 - M_2)$ listed in Table 3.3.

Case 3.1.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A3)}}, \text{(A1)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c3)},\text{(c3.1)}}{\Longrightarrow} \{\text{(A2)}, \text{(A4)}, \text{(B1)}, \text{(B2)}, \text{(B3)}, \text{(B5)}\} \tag{B.95}$$

by reusing (B.62), (B.64), (B.66), and the following straightforward intermediate steps

$$\{\overline{\text{(A3)}}, \text{(A1)}\} \overset{\text{(c3.ab)},\text{(G4)},\text{(c3.1)}}{\Longrightarrow} \text{(B2)} \tag{B.96}$$

$$\{\overline{\text{(A3)}}, \text{(B4)}\} \overset{\text{(c3.ab)},\text{(G1)}}{\Longrightarrow} \text{(A2)} \tag{B.97}$$

$$\{\overline{\text{(A3)}}, \text{(A1)}, \text{(B4)}\} \overset{\text{(c3.ab)}}{\Longrightarrow} \text{(B1)} \tag{B.98}$$

$$\{\overline{\text{(A3)}}, \text{(B4)}, \text{(B6)}\} \overset{\text{(c3.ab)},\text{(G1)},\text{(c3)}}{\Longrightarrow} \text{(B5)}. \tag{B.99}$$

By (B.95), the four tight linear inequalities in Case-3.1.3 can only be (A3) (thus $\overline{\text{(A3)}}$), (A1), (B4), and (B6). Solving these equations, the corresponding corner point is Vertex 14 $(F_1 - M_2, F_2, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 3.1.4: (A4) is tight. i.e., $\overline{\text{(A4)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A4)}}, \text{(A2)}, \text{(B3)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c3)},\text{(c3.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A3)}, \text{(B1)}, \text{(B2)}, \text{(B4)}, \text{(B5)}\} \tag{B.100}$$

by reusing (B.28), (B.69), (B.70), and the following straightforward intermediate steps

$$\{\overline{\text{(A4)}}, \text{(A2)}\} \overset{\text{(c3.ab)},\text{(c3.1)}}{\Longrightarrow} \text{(B5)} \tag{B.101}$$

$$\{\overline{\text{(A4)}}, \text{(A2)}, \text{(B3)}\} \overset{\text{(c3.ab)},\text{(G4)}}{\Longrightarrow} \text{(B1)} \tag{B.102}$$

$$\{\overline{\text{(A4)}}, \text{(A2)}, \text{(B6)}\} \overset{\text{(c3.ab)},\text{(G4)}}{\Longrightarrow} \text{(B4)}. \tag{B.103}$$

By (B.100), the four tight linear inequalities in Case-3.1.4 can only be (A4) (thus $\overline{\text{(A4)}}$), (A2), (B3), and (B6). Solving these equations, the corresponding corner point is Vertex 15 $(F_1, F_2 - M_2, F_1, F_2 - M_2)$ listed in Table 3.3.

Case 3.1.5: None of (A1) to (A4) is tight. Recall that in all the discussion of Case 3.1 and its subcases, we assume $\mathcal{G}$, (c3), (c3.1), and (c3.ab). Since

$$\{\text{(A1)}, \text{(A3)}\} \overset{\text{(c3.ab)},\text{(G4)},\text{(c3.1)}}{\Longrightarrow} \text{(B2)}, \tag{B.104}$$

$$\{\text{(A2)}, \text{(A4)}\} \overset{\text{(c3.ab)},\text{(c3.1)}}{\Longrightarrow} \text{(B5)}, \tag{B.105}$$

any corner point that is *loose* for all 4 inequalities (A1) to (A4) must also be loose for (B2) and (B5). Therefore the corner point must be decided by the 4 remaining inequalities (B1), (B3), (B4), and (B5). By (c3.ab), the inequalities corresponding to (B1), (B3), (B4), and (B5) are *co-dependent* such that they are not possible to determine a corner point. As the result, there is no corner point in Case-3.1.5.

Case 3.2: In this case we assume both (c3) and (c3.2) are true. This sub-case is the left subregion above the dotted line in the solid lower-right square (Case 3) of Fig. 3.2a and mid

subregion between the two dotted lines in the solid lower-right square (Case 3) of Fig. 3.2b. We now consider the following 5 subcases of Case 3.2.

Case 3.2.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. Since the intermediate step (B.85) in Case-3.1.1 only require $F_2 + M_2 \leq M_1$, we can substitute $\{\text{(G4)}, \text{(c3.1)}\}$ with (c3.2) in (B.85). The rest of the analysis is verbatim to Case-3.1.1 and the corner point is also Vertex 14.

Case 3.2.2: (A2) is tight. i.e., $\overline{\text{(A2)}}$ holds. We can prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}, \text{(B5)}\} \overset{\mathcal{G},\text{(c3)},\text{(c3.2)}}{\Longrightarrow} \\ &\{\text{(A1)}, \text{(A3)}, \text{(A4)}, \text{(B2)}, \text{(B3)}, \text{(B6)}\}\end{aligned} \tag{B.106}$$

by reusing (B.89), (B.90), (B.92), and the following straightforward intermediate steps

$$\{\overline{\text{(A2)}}, \text{(B5)}\} \overset{\text{(c3.ab)},\text{(c3.2)}}{\Longrightarrow} \text{(A4)} \tag{B.107}$$

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B5)}\} \overset{\text{(c3.ab)},\text{(c3.2)}}{\Longrightarrow} \text{(B3)} \tag{B.108}$$

$$\{\overline{\text{(A2)}}, \text{(B4)}, \text{(B5)}\} \overset{\text{(c3.ab)},\text{(c3.2)}}{\Longrightarrow} \text{(B6)}. \tag{B.109}$$

By (B.106), the four tight linear inequalities in Case-3.2.2 can only be (A2) (thus $\overline{\text{(A2)}}$), (B1), (B4), and (B5). Solving these equations, the corresponding corner point is *Vertex 16* $(F_1, F_2 - M_2, F_1, F_1 + F_2 - M_1)$ listed in Table 3.3.

Case 3.2.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. Since the intermediate step (B.96) in Case-3.1.3 only require $F_2 + M_2 \leq M_1$, we can substitute $\{\text{(G4)}, \text{(c3.1)}\}$ with (c3.2) in (B.96). The rest of the analysis is verbatim to Case-3.1.3 and the corner point is also Vertex 14.

Case 3.2.4: (A4) is tight. i.e., $\overline{\text{(A4)}}$ holds. We can prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B5)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c3)},\text{(c3.2)}}{\Longrightarrow} \\ &\{\text{(A1)}, \text{(A2)}, \text{(A3)}, \text{(B1)}, \text{(B2)}, \text{(B4)}\}\end{aligned} \tag{B.110}$$

by reusing (B.28), (B.69), (B.70), and the following straightforward intermediate steps

$$\{\overline{(A4)}, (B5)\} \overset{(c3.ab),(c3.2)}{\Longrightarrow} (A2) \tag{B.111}$$

$$\{\overline{(A4)}, (B3), (B5)\} \overset{(c3.ab),(c3.2)}{\Longrightarrow} (B1) \tag{B.112}$$

$$\{\overline{(A4)}, (B5), (B6)\} \overset{(c3.ab),(c3.2)}{\Longrightarrow} (B4). \tag{B.113}$$

By (B.110), the four tight linear inequalities in Case-3.2.4 can only be (A4) (thus $\overline{(A4)}$), (B3), (B5), and (B6). Solving these equations, the corresponding corner point is Vertex 10 $(F_1, F_1 + F_2 - M_1, F_1, F_2 - M_2)$ listed in Table 3.3.

Case 3.2.5: None of (A1) to (A4) is tight. Recall that in all the discussion of Case 3.2 and its subcases, we assume $\mathcal{G}$, (c3), (c3.2), and (c3.ab). Since

$$\{(A1), (A3)\} \overset{(c3.ab),(c3.2)}{\Longrightarrow} (B2), \tag{B.114}$$

the corner point must be decided by 4 out of the 5 remaining inequalities (B1), (B3), (B4), (B5), and (B6). By (c3.ab), the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent* such that a corner point requires all 5 inequalities (B1), (B3), (B4), (B5), and (B6) to be tight simultaneously. The 5 inequalities jointly yield exactly one corner point *Vertex 17* $(\frac{F_1+M_1-M_2}{2}, F_2 + \frac{F_1-M_1-M_2}{2}, \frac{F_1+M_1-M_2}{2}, F_2 + \frac{F_1-M_1-M_2}{2})$ listed in Table 3.3. Vertex 17 is a legitimate corner point since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 17} \overset{(c3.ab),(G1),(c3)}{\Rightarrow} (A1);\ \text{Vertex 17} \overset{(c3.ab),(c3.2)}{\Rightarrow} (A2); \tag{B.115}$$

$$\text{Vertex 17} \overset{(c3.ab),(G1),(c3)}{\Rightarrow} (A3);\ \text{Vertex 17} \overset{(c3.ab),(c3.2)}{\Rightarrow} (A4). \tag{B.116}$$

Case 3.3: In this case we assume both (c3) and (c3.3) are true. This sub-case is the leftmost subregion above the dotted line in the solid lower-right square (Case 3) of Fig. 3.2b. We now consider the following 6 subcases of Case 3.3.

Case 3.3.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c3)},\text{(c3.3)}}{\Longrightarrow} \{\text{(A2)}, \text{(A3)}, \text{(A4)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\} \tag{B.117}$$

by reusing (B.84), (B.87), and the following straightforward intermediate steps

$$\{\overline{\text{(A1)}}, \text{(B2)}\} \overset{\text{(c3.ab)},\text{(c3.3)}}{\Longrightarrow} \text{(A3)} \tag{B.118}$$

$$\{\overline{\text{(A1)}}, \text{(B3)}\} \overset{\text{(c3.ab)},\text{(G1)}}{\Longrightarrow} \text{(A4)} \tag{B.119}$$

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}\} \overset{\text{(c3.ab)},\text{(c3.3)}}{\Longrightarrow} \text{(B4)} \tag{B.120}$$

$$\{\overline{\text{(A1)}}, \text{(B2)}, \text{(B3)}\} \overset{\text{(c3.ab)},\text{(c3.3)}}{\Longrightarrow} \text{(B6)} \tag{B.121}$$

By (B.117), the four tight linear inequalities in Case-3.3.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (B1), (B2), and (B3). Solving these equations, the corresponding corner point is *Vertex 18* $(F_1 - M_2, F_2, F_1 + F_2 - M_1, F_2)$ listed in Table 3.3.

Case 3.3.2: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. Since the intermediate steps (B.107) to (B.109) in Case-3.2.2 only require $M_1 \leq F_1 + M_2$, we can substitute (c3.2) with $\{\text{(G4)}, \text{(c3.3)}\}$ in Case-3.2.2. The rest of the analysis is verbatim to Case-3.2.2 and the corner point is also Vertex 16.

Case 3.3.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c3)},\text{(c3.3)}}{\Longrightarrow} \{\text{(A1)}, \text{(A2)}, \text{(A4)}, \text{(B1)}, \text{(B3)}, \text{(B5)}\} \tag{B.122}$$

by reusing (B.97), (B.99), and the following straightforward intermediate steps

$$\{\overline{(A3)}, (B2)\} \overset{(c3.ab),(c3.3)}{\Longrightarrow} (A1) \tag{B.123}$$

$$\{\overline{(A3)}, (B6)\} \overset{(c3.ab),(G1)}{\Longrightarrow} (A4) \tag{B.124}$$

$$\{\overline{(A3)}, (B2), (B4)\} \overset{(c3.ab),(c3.3)}{\Longrightarrow} (B1) \tag{B.125}$$

$$\{\overline{(A3)}, (B2), (B6)\} \overset{(c3.ab),(c3.3)}{\Longrightarrow} (B3) \tag{B.126}$$

By (B.122), the four tight linear inequalities in Case-3.3.3 can only be (A3) (thus $\overline{(A3)}$), (B2), (B4), and (B6). Solving these equations, the corresponding corner point is *Vertex 19* $(F_1 + F_2 - M_1, F_2, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 3.3.4: (A4) is tight. i.e., $\overline{(A4)}$ holds. Since the intermediate steps (B.111) to (B.113) in Case-3.2.4 only require $M_1 \leq F_1 + M_2$, we can substitute (c3.2) with $\{(G4), (c3.3)\}$ in Case-3.2.4. The rest of the analysis is verbatim to Case-3.2.4 and the corner point is also Vertex 10.

Case 3.3.5: None of (A1) to (A4) is tight, but (B2) is tight, i.e., $\overline{(B2)}$ holds. We can prove

$$\{(B1), \overline{(B2)}, (B3), (B4), (B6)\} \overset{\mathcal{G},(c3),(c3.3)}{\Rightarrow} \{(B5)\} \tag{B.127}$$

using the following intermediate step

$$\{(B1), (B6), \overline{(B2)}\} \overset{(c3.ab),(c3)}{\Rightarrow} \{(B5)\}. \tag{B.128}$$

Recall that under the a_1 to b_6 values in (c3.ab), the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent* as shown in Case-3.2.5. Therefore, the corner point must be tight for all 5 inequalities (B1), $\overline{(B2)}$, (B3), (B4), (B6). Solving these 5 joint equations (four of them are co-dependent), we obtain the corner point *Vertex 20* $(F_1 +$

$\frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2}, F_1+\frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2}$) listed in Table 3.3. Vertex 20 is a legitimate corner point since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 20} \overset{\text{(c3.ab)},\text{(c3.3)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 20} \overset{\text{(c3.ab)},\text{(c3)}}{\Rightarrow} \text{(A2)};$$
$$\text{Vertex 20} \overset{\text{(c3.ab)},\text{(c3.3)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 20} \overset{\text{(c3.ab)},\text{(c3)}}{\Rightarrow} \text{(A4)}.$$

Case 3.3.6: None of (A1) to (A4) is tight, nor is (B2). Since the arguments in Case-3.2.5 after (B.104) only require the condition $M_1 \le F_1 + M_2$, we can change (c3.2) in (B.115) and (B.116) with (c3.3) and use the same argument verbatim to prove that the corner point in Case-3.3.6 is the Vertex 17 ($\frac{F_1+M_1-M_2}{2}, F_2+\frac{F_1-M_1-M_2}{2}, \frac{F_1+M_1-M_2}{2}, F_2+\frac{F_1-M_1-M_2}{2}$) listed in Table 3.3.

Case 4: We assume

$$F_2 < M_2 \le M_1 \le F_1. \tag{c4}$$

Ineq. (c4) and $\mathcal{G}$ jointly describe the scenario when the (M_1, M_2) value falls into the mid-left triangle in Fig. 3.2 with solid edges and being marked as "Case 4". In this case, the a_1 to b_6 values of (A1) to (B6) become

$$\begin{aligned} &a_1 = a_3 = F_1 - M_2, \quad a_2 = F_1 - M_1, \quad a_4 = 0, \\ &b_1 = b_2 = b_4 = 2F_1 + F_2 - M_1 - M_2, \\ &b_3 = b_6 = F_1 + F_2 - M_2, \quad b_5 = F_1 + F_2 - M_1. \end{aligned} \tag{c4.ab}$$

Note that only a_2, a_4, and b_5 are different in (c2.ab) and (c4.ab), and the rest 7 values are identical. Therefore we can reuse the equations related to (c2.ab) when (A2), (A4) and (B5) are not involved. We now further divide this case into two sub-cases. **Case 4.1:** We assume

$$F_2 + M_2 < M_1, \tag{c4.1}$$

and **Case 4.2:** We assume

$$F_2 + M_2 \geq M_1. \tag{c4.2}$$

Cases 4.1 and 4.2 further divide the solid mid-left triangle of Fig. 3.2a by a 45-degree line. Note that (c4.1) and (c4.2) are identical to (c2.1) and (c2.2), respectively, such that we can reuse the equations in Case 2 by directly changing (c2.1) (resp. (c2.2)) to (c4.1) (resp.(c4.2)). In the following we focus on Case 4.1, the right subregion below the dotted line.

Case 4.1: We consider the following 5 subcases.

Case 4.1.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A1)}}, \text{(A3)}, \text{(B1)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c4)},\text{(c4.1)}}{\Longrightarrow}\\ &\{\text{(A2)}, \text{(A4)}, \text{(B2)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\}\end{aligned} \tag{B.129}$$

by reusing (B.54), (B.55), (B.56), and the following straightforward intermediate steps

$$\{\overline{\text{(A1)}}, \text{(B1)}\} \overset{\text{(c4.ab)},\text{(G2)}}{\Longrightarrow} \text{(A2)} \tag{B.130}$$

$$\{\overline{\text{(A1)}}, \text{(B3)}\} \overset{\text{(c4.ab)},\text{(G2)}}{\Longrightarrow} \text{(A4)} \tag{B.131}$$

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B3)}\} \overset{\text{(c4.ab)},\text{(G2)}}{\Longrightarrow} \text{(B5)} \tag{B.132}$$

By (B.129), the four tight linear inequalities in Case-4.1.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (A3), (B1), and (B3). Solving these equations, the corresponding corner point is Vertex 8 $(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 4.1.2: (A2) is tight. i.e., $\overline{\text{(A2)}}$ holds. We can prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}, \text{(B5)}\} \overset{\mathcal{G},\text{(c4)},\text{(c4.1)}}{\Longrightarrow}\\ &\{\text{(A1)}, \text{(A3)}, \text{(A4)}, \text{(B2)}, \text{(B3)}, \text{(B6)}\}\end{aligned} \tag{B.133}$$

by the following straightforward intermediate steps

$$\{\overline{(A2)}, (B1)\} \overset{(c4.ab),(G2)}{\Longrightarrow} (A1) \tag{B.134}$$

$$\{\overline{(A2)}, (B4)\} \overset{(c4.ab),(G2)}{\Longrightarrow} (A3) \tag{B.135}$$

$$\{\overline{(A2)}, (B5)\} \overset{(c4.ab),(G2)}{\Longrightarrow} (A4) \tag{B.136}$$

$$\{\overline{(A2)}, (B1), (B4)\} \overset{(c4.ab),(G2),(G3)}{\Longrightarrow} (B2) \tag{B.137}$$

$$\{\overline{(A2)}, (B1), (B5)\} \overset{(c4.ab),(G2)}{\Longrightarrow} (B3) \tag{B.138}$$

$$\{\overline{(A2)}, (B4), (B5)\} \overset{(c4.ab),(G2)}{\Longrightarrow} (B6). \tag{B.139}$$

By (B.133), the four tight linear inequalities in Case-4.1.2 can only be (A2) (thus $\overline{(A2)}$), (B1), (B4), and (B5). Solving these equations, the corresponding corner point is *Vertex 21* $(F_1 + F_2 - M_2, F_1 - M_1, F_1 + F_2 - M_2, F_2)$ listed in Table 3.3.

Case 4.1.3: (A3) is tight. i.e., $\overline{(A3)}$ holds. We can prove the following relationship

$$\{\overline{(A3)}, (A1), (B4), (B6)\} \overset{\mathcal{G},(c4),(c4.1)}{\Longrightarrow}$$
$$\{(A2), (A4), (B1), (B2), (B3), (B5)\} \tag{B.140}$$

by reusing (B.62), (B.65), (B.66), and the following straightforward intermediate steps

$$\{\overline{(A3)}, (B4)\} \overset{(c4.ab),(G2)}{\Longrightarrow} (A2) \tag{B.141}$$

$$\{\overline{(A3)}, (B6)\} \overset{(c4.ab),(G2)}{\Longrightarrow} (A4) \tag{B.142}$$

$$\{\overline{(A3)}, (B4), (B6)\} \overset{(c4.ab),(G2)}{\Longrightarrow} (B5). \tag{B.143}$$

By (B.140), the four tight linear inequalities in Case-4.1.3 can only be (A3) (thus $\overline{(A3)}$), (A1), (B4), and (B6). Solving these equations, the corresponding corner point is Vertex 8 $(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 4.1.4: (A4) is tight. i.e., $\overline{\text{(A4)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B5)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c4)},\text{(c4.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A2)}, \text{(A3)}, \text{(B1)}, \text{(B2)}, \text{(B4)}\} \tag{B.144}$$

by the following straightforward intermediate steps

$$\{\overline{\text{(A4)}}, \text{(B3)}\} \overset{\text{(c4.ab)},\text{(G2)}}{\Longrightarrow} \text{(A1)} \tag{B.145}$$

$$\{\overline{\text{(A4)}}, \text{(B5)}\} \overset{\text{(c4.ab)},\text{(G2)}}{\Longrightarrow} \text{(A2)} \tag{B.146}$$

$$\{\overline{\text{(A4)}}, \text{(B6)}\} \overset{\text{(c4.ab)},\text{(G2)}}{\Longrightarrow} \text{(A3)} \tag{B.147}$$

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B5)}\} \overset{\text{(c4.ab)},\text{(G2)}}{\Longrightarrow} \text{(B1)} \tag{B.148}$$

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B6)}\} \overset{\text{(c4.ab)},\text{(G2)},\text{(G3)}}{\Longrightarrow} \text{(B2)} \tag{B.149}$$

$$\{\overline{\text{(A4)}}, \text{(B5)}, \text{(B6)}\} \overset{\text{(c4.ab)},\text{(G2)}}{\Longrightarrow} \text{(B4)}. \tag{B.150}$$

By (B.144), the four tight linear inequalities in Case-4.1.4 can only be (A4) (thus $\overline{\text{(A4)}}$), (B3), (B5), and (B6). Solving these equations, the corresponding corner point is *Vertex 22* $(F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0)$ listed in Table 3.3.

Case 4.1.5: None of (A1) to (A4) is tight. Recall that in all the discussion of Case 4.1 and its subcases, we assume $\mathcal{G}$, (c4), (c4.1), and (c4.ab). Since

$$\{\text{(A1)}, \text{(A3)}\} \overset{\text{(c4.ab)},\text{(c4.1)}}{\Longrightarrow} \text{(B2)}, \tag{B.151}$$

and the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent*, a corner point requires all the remaining 5 inequalities (B1), (B3), (B4), (B5), and (B6) to be tight simultaneously. The 5 inequalities jointly yield exactly one corner point *Vertex 23* $(F_1 +$

$\frac{F_2}{2} - M_2, F_1 + \frac{F_2}{2} - M_1, F_1 + \frac{F_2}{2} - M_2, \frac{F_2}{2}$) listed in Table 3.3. Vertex 23 is a legitimate corner point since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 23} \overset{\text{(c4.ab)},\text{(G2)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 23} \overset{\text{(c4.ab)},\text{(G2)}}{\Rightarrow} \text{(A2)};$$
$$\text{Vertex 23} \overset{\text{(c4.ab)},\text{(G2)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 23} \overset{\text{(c4.ab)},\text{(G2)}}{\Rightarrow} \text{(A4)}.$$

Case 4.2: In this case we assume both (c4) and (c4.2) are true. This sub-case is the left subregion above the dotted line in the solid mid-left triangle (Case 4) of Fig. 3.2a. We now consider the following 6 subcases of Case 4.2.

Case 4.2.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c4)},\text{(c4.2)}}{\Longrightarrow}$$
$$\{\text{(A2)}, \text{(A3)}, \text{(A4)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\} \tag{B.152}$$

by reusing (B.130), (B.74), (B.131), (B.75) (B.132), and (B.76). By (B.152), the four tight linear inequalities in Case-4.2.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (B1), (B2), and (B3). Solving these equations, the corresponding corner point is Vertex 1 ($F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2$) listed in Table 3.3.

Case 4.2.2: (A2) is tight, i.e., $\overline{\text{(A2)}}$ holds. Since the intermediate steps of Case 4.1.2 does not require condition (c4.2), the statement in (B.133) holds even if we swap out (c4.1) by (c4.2). The rest of the analysis is verbatim to Case 4.1.2 and the corner point is also Vertex 21.

Case 4.2.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c4)},\text{(c4.2)}}{\Longrightarrow}$$
$$\{\text{(A1)}, \text{(A2)}, \text{(A4)}, \text{(B1)}, \text{(B3)}, \text{(B5)}\} \tag{B.153}$$

by reusing (B.78), (B.141), (B.142) (B.79), (B.80), and (B.143). By (B.153), the four tight linear inequalities in Case-4.2.3 can only be (A3) (thus $\overline{\text{(A3)}}$), (B2), (B4), and (B6). Solving

these equations, the corresponding corner point is Vertex 12 $(F_1+F_2-M_1, F_1+F_2-M_1, F_1-M_2, F_2)$ listed in Table 3.3.

Case 4.2.4: (A4) is tight, i.e., $\overline{\text{(A4)}}$ holds. Since the intermediate steps of Case 4.1.4 does not require condition (c4.2), the statement in (B.144) holds even if we swap out (c4.1) by (c4.2). The rest of the analysis is verbatim to Case 4.1.4 and the corner point is also Vertex 22.

Case 4.2.5: None of (A1) to (A4) is tight, but (B2) is tight, i.e., $\overline{\text{(B2)}}$ holds. We can prove

$$\{\text{(B1)}, \overline{\text{(B2)}}, \text{(B3)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c4)},\text{(c4.2)}}{\Rightarrow} \{\text{(B5)}\} \tag{B.154}$$

using the following intermediate step

$$\{\text{(B1)}, \text{(B6)}, \overline{\text{(B2)}}\} \overset{\text{(c4.ab)},\text{(G3)}}{\Rightarrow} \{\text{(B5)}\}. \tag{B.155}$$

That is, in this case, we do not need to check whether (B5) is tight or not. Recall that under the a_1 to b_6 values in (c4.ab), the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent* as shown in Case 4.1.5. Therefore, the corner point must be tight for all 5 inequalities (B1), $\overline{\text{(B2)}}$, (B3), (B4), (B6). Solving these 5 joint equations, the corresponding corner point is Vertex 13 $(F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2})$ listed in Table 3.3. Vertex 13 is a legitimate corner point in this case since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 13} \overset{\text{(c4.ab)},\text{(c4.2)}}{\Rightarrow} \text{(A1)}; \ \text{Vertex 13} \overset{\text{(c4.ab)},\text{(G2)},\text{(G3)}}{\Rightarrow} \text{(A2)};$$
$$\text{Vertex 13} \overset{\text{(c4.ab)},\text{(c4.2)}}{\Rightarrow} \text{(A3)}; \ \text{Vertex 13} \overset{\text{(c4.ab)},\text{(G2)},\text{(G3)}}{\Rightarrow} \text{(A4)}.$$

Case 4.2.6: None of (A1) to (A4) is tight, nor is (B2). Since the arguments in Case 4.1.5 after (B.151) no longer uses the condition (c4.1), we can use the same argument verbatim and prove that the corner point in Case 4.2.6 is the Vertex 23 $(F_1 + \frac{F_2}{2} - M_2, F_1 + \frac{F_2}{2} - M_1, F_1 + \frac{F_2}{2} - M_2, \frac{F_2}{2})$ listed in Table 3.3.

Case 5: We assume

$$F_2 < M_2 \leq F_1 < M_1. \tag{c5}$$

Ineq. (c5) and $\mathcal{G}$ jointly describe the scenario when the (M_1, M_2) value falls into the mid-right rectangle in Fig. 3.2 with solid edges and being marked as "Case 5". In this case, the a_1 to b_6 values of (A1) to (B6) become

$$\begin{aligned} &a_1 = a_3 = F_1 - M_2, \quad a_2 = a_4 = 0, \\ &b_1 = b_3 = b_4 = b_6 = F_1 + F_2 - M_2, \\ &b_2 = 2F_1 + F_2 - M_1 - M_2, \quad b_5 = F_1 + F_2 - M_1. \end{aligned} \tag{c5.ab}$$

Note that only a_2, b_1, and b_4 are different in (c4.ab) and (c5.ab), and the rest 7 values are identical. Therefore we can reuse the equations related to (c4.ab) (or (c2.ab) referred in Case 4) when (A2), (B1) and (B4) are not involved. We now further divide this case into two sub-cases. **Case 5.1:** We assume

$$F_2 + M_2 < M_1, \tag{c5.1}$$

and **Case 5.2:** We assume

$$F_2 + M_2 \geq M_1. \tag{c5.2}$$

Cases 5.1 and 5.2 further divide the solid mid-right rectangle of Fig. 3.2 by a 45-degree line. Note that (c5.1) and (c5.2) are identical to (c4.1) and (c4.2), respectively, such that we can reuse the equations in Case 4 by directly changing (c4.1) (resp. (c4.2)) to (c5.1) (resp.(c5.2)). In the following we focus on Case 5.1, the right subregion below the dotted line.

Case 5.1: We consider the following 5 subcases.

Case 5.1.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A1)}}, \text{(A3)}, \text{(B1)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c5)},\text{(c5.1)}}{\Longrightarrow} \{\text{(A2)}, \text{(A4)}, \text{(B2)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\} \tag{B.156}$$

by reusing (B.54), (B.131), (B.56), and the following straightforward intermediate steps

$$\{\overline{\text{(A1)}}, \text{(B1)}\} \overset{\text{(c5.ab)},\text{(G2)}}{\Longrightarrow} \text{(A2)} \tag{B.157}$$

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}\} \overset{\text{(c5.ab)}}{\Longrightarrow} \text{(B4)} \tag{B.158}$$

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B3)}\} \overset{\text{(c5.ab)},\text{(G2)},\text{(c5)}}{\Longrightarrow} \text{(B5)} \tag{B.159}$$

By (B.156), the four tight linear inequalities in Case-5.1.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (A3), (B1), and (B3). Solving these equations, the corresponding corner point is Vertex 14 $(F_1 - M_2, F_2, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 5.1.2: (A2) is tight. i.e., $\overline{\text{(A2)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}, \text{(B5)}\} \overset{\mathcal{G},\text{(c5)},\text{(c5.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A3)}, \text{(A4)}, \text{(B2)}, \text{(B3)}, \text{(B6)}\} \tag{B.160}$$

by the following straightforward intermediate steps

$$\{\overline{\text{(A2)}}, \text{(B1)}\} \overset{\text{(c5.ab)},\text{(G2)}}{\Longrightarrow} \text{(A1)} \tag{B.161}$$

$$\{\overline{\text{(A2)}}, \text{(B4)}\} \overset{\text{(c5.ab)},\text{(G2)}}{\Longrightarrow} \text{(A3)} \tag{B.162}$$

$$\{\overline{\text{(A2)}}, \text{(B5)}\} \overset{\text{(c5.ab)},\text{(G5)}}{\Longrightarrow} \text{(A4)} \tag{B.163}$$

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}\} \overset{\text{(c5.ab)},\text{(G2)},\text{(G3)}}{\Longrightarrow} \text{(B2)} \tag{B.164}$$

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B5)}\} \overset{\text{(c5.ab)},\text{(G5)}}{\Longrightarrow} \text{(B3)} \tag{B.165}$$

$$\{\overline{\text{(A2)}}, \text{(B4)}, \text{(B5)}\} \overset{\text{(c5.ab)},\text{(G5)}}{\Longrightarrow} \text{(B6)}. \tag{B.166}$$

By (B.160), the four tight linear inequalities in Case-5.1.2 can only be (A2) (thus $\overline{\text{(A2)}}$), (B1), (B4), and (B5). Solving these equations, the corresponding corner point is *Vertex 24* $(F_1+F_2-M_2, 0, F_1+F_2-M_2, F_1+F_2-M_1)$ listed in Table 3.3.

Case 5.1.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A3)}}, \text{(A1)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c5)},\text{(c5.1)}}{\Longrightarrow} \{\text{(A2)}, \text{(A4)}, \text{(B1)}, \text{(B2)}, \text{(B3)}, \text{(B5)}\} \tag{B.167}$$

by reusing (B.62), (B.142), (B.66), and the following straightforward intermediate steps

$$\{\overline{\text{(A3)}}, \text{(B4)}\} \overset{\text{(c5.ab)},\text{(G2)}}{\Longrightarrow} \text{(A2)} \tag{B.168}$$

$$\{\overline{\text{(A3)}}, \text{(A1)}, \text{(B4)}\} \overset{\text{(c5.ab)}}{\Longrightarrow} \text{(B1)} \tag{B.169}$$

$$\{\overline{\text{(A3)}}, \text{(B4)}, \text{(B6)}\} \overset{\text{(c5.ab)},\text{(G2)},\text{(c5)}}{\Longrightarrow} \text{(B5)}. \tag{B.170}$$

By (B.167), the four tight linear inequalities in Case-5.1.3 can only be (A3) (thus $\overline{\text{(A3)}}$), (A1), (B4), and (B6). Solving these equations, the corresponding corner point is Vertex 14 $(F_1-M_2, F_2, F_1-M_2, F_2)$ listed in Table 3.3.

Case 5.1.4: (A4) is tight. i.e., $\overline{\text{(A4)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B5)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c5)},\text{(c5.1)}}{\Longrightarrow} \{\text{(A1)}, \text{(A2)}, \text{(A3)}, \text{(B1)}, \text{(B2)}, \text{(B4)}\} \tag{B.171}$$

by reusing (B.145), (B.147), (B.149), and the following straightforward intermediate steps

$$\{\overline{\text{(A4)}}, \text{(B5)}\} \overset{\text{(c5.ab)},\text{(G2)}}{\Longrightarrow} \text{(A2)} \tag{B.172}$$

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B5)}\} \overset{\text{(c5.ab)},\text{(G5)}}{\Longrightarrow} \text{(B1)} \tag{B.173}$$

$$\{\overline{\text{(A4)}}, \text{(B5)}, \text{(B6)}\} \overset{\text{(c5.ab)},\text{(G5)}}{\Longrightarrow} \text{(B4)}. \tag{B.174}$$

By (B.171), the four tight linear inequalities in Case-5.1.4 can only be (A4) (thus $\overline{\text{(A4)}}$), (B3), (B5), and (B6). Solving these equations, the corresponding corner point is Vertex 22 $(F_1+F_2-M_2, F_1+F_2-M_1, F_1+F_2-M_2, 0)$ listed in Table 3.3.

Case 5.1.5: None of (A1) to (A4) is tight. Recall that in all the discussion of Case 5.1 and its subcases, we assume $\mathcal{G}$, (c5), (c5.1), and (c5.ab). Since

$$\{\text{(A1)}, \text{(A3)}\} \overset{\text{(c5.ab)},\text{(c5.1)}}{\Longrightarrow} \text{(B2)}, \tag{B.175}$$

and the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent*, a corner point requires all the remaining 5 inequalities (B1), (B3), (B4), (B5), and (B6) to be tight simultaneously. The 5 inequalities jointly yield exactly one corner point *Vertex 25* $(\frac{F_1+F_2+M_1}{2}-M_2, \frac{F_1+F_2-M_1}{2}, \frac{F_1+F_2+M_1}{2}-M_2, \frac{F_1+F_2-M_1}{2})$ listed in Table 3.3. Vertex 25 is a legitimate corner point since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 25} \overset{\text{(c5.ab)},\text{(G2)},\text{(c5)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 25} \overset{\text{(c5.ab)},\text{(G5)}}{\Rightarrow} \text{(A2)};$$
$$\text{Vertex 25} \overset{\text{(c5.ab)},\text{(G2)},\text{(c5)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 25} \overset{\text{(c5.ab)},\text{(G5)}}{\Rightarrow} \text{(A4)}.$$

Case 5.2: In this case we assume both (c5) and (c5.2) are true. This sub-case is the left subregion above the dotted line in the solid mid-right rectangle (Case 5) of Fig. 3.2. We now consider the following 6 subcases of Case 5.2.

Case 5.2.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c5)},\text{(c5.2)}}{\Longrightarrow}$$
$$\{\text{(A2)}, \text{(A3)}, \text{(A4)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\} \tag{B.176}$$

by reusing (B.157), (B.74), (B.131), (B.158), and the following straightforward intermediate steps

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}\} \overset{\text{(c5.ab)},\text{(c5.2)}}{\Longrightarrow} \text{(B4)} \tag{B.177}$$
$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B3)}\} \overset{\text{(c5.ab)},\text{(c5.2)}}{\Longrightarrow} \text{(B6)}. \tag{B.178}$$

By (B.176), the four tight linear inequalities in Case-5.2.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (B1), (B2), and (B3). Solving these equations, the corresponding corner point is Vertex 18 $(F_1 - M_2, F_2, F_1 + F_2 - M_1, F_2)$ listed in Table 3.3.

Case 5.2.2: (A2) is tight, i.e., $\overline{\text{(A2)}}$ holds. Since the intermediate steps of Case 5.1.2 does not require condition (c5.2), the statement in (B.160) holds even if we swap out (c5.1) by (c5.2). The rest of the analysis is verbatim to Case 5.1.2 and the corner point is also Vertex 24.

Case 5.2.3: (A3) is tight. i.e., $\overline{\text{(A3)}}$ holds. We can prove the following relationship

$$\begin{aligned}&\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c5)},\text{(c5.2)}}{\Longrightarrow} \\ &\{\text{(A1)}, \text{(A2)}, \text{(A4)}, \text{(B1)}, \text{(B3)}, \text{(B5)}\}\end{aligned} \tag{B.179}$$

by reusing (B.78), (B.168), (B.142), (B.170), and the following straightforward intermediate steps

$$\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B4)}\} \overset{\text{(c5.ab)},\text{(c5.2)}}{\Longrightarrow} \text{(B1)} \tag{B.180}$$

$$\{\overline{\text{(A3)}}, \text{(B2)}, \text{(B6)}\} \overset{\text{(c5.ab)},\text{(c5.2)}}{\Longrightarrow} \text{(B3)}. \tag{B.181}$$

By (B.179), the four tight linear inequalities in Case-5.2.3 can only be (A3) (thus $\overline{\text{(A3)}}$), (B2), (B4), and (B6). Solving these equations, the corresponding corner point is Vertex 19 $(F_1 + F_2 - M_1, F_2, F_1 - M_2, F_2)$ listed in Table 3.3.

Case 5.2.4: (A4) is tight, i.e., $\overline{\text{(A4)}}$ holds. Since the intermediate steps of Case 5.1.4 does not require condition (c5.2), the statement in (B.171) holds even if we swap out (c5.1) by (c5.2). The rest of the analysis is verbatim to Case 5.1.4 and the corner point is also Vertex 22.

Case 5.2.5: None of (A1) to (A4) is tight, but (B2) is tight, i.e., $\overline{\text{(B2)}}$ holds. We can prove

$$\{\text{(B1)}, \overline{\text{(B2)}}, \text{(B3)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c5)},\text{(c5.2)}}{\Rightarrow} \{\text{(B5)}\} \tag{B.182}$$

using the following intermediate step

$$\{(\text{B1}), (\text{B6}), \overline{(\text{B2})}\} \overset{(\text{c5.ab}),(\text{c5})}{\Rightarrow} \{(\text{B5})\}. \tag{B.183}$$

That is, in this case, we do not need to check whether (B5) is tight or not. Recall that under the a_1 to b_6 values in (c5.ab), the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent* as shown in Case 5.1.5. Therefore, the corner point must be tight for all 5 inequalities (B1), $\overline{(\text{B2})}$, (B3), (B4), (B6). Solving these 5 joint equations, the corresponding corner point is Vertex 20 $(F_1 + \frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2})$ listed in Table 3.3. Vertex 20 is a legitimate corner point in this case since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 20} \overset{(\text{c5.ab}),(\text{c5.2})}{\Rightarrow} (\text{A1}); \ \text{Vertex 20} \overset{(\text{c5.ab}),(\text{G2}),(\text{G3})}{\Rightarrow} (\text{A2});$$
$$\text{Vertex 20} \overset{(\text{c5.ab}),(\text{c5.2})}{\Rightarrow} (\text{A3}); \ \text{Vertex 20} \overset{(\text{c5.ab}),(\text{G2}),(\text{G3})}{\Rightarrow} (\text{A4}).$$

Case 5.2.6: None of (A1) to (A4) is tight, nor is (B2). Since the arguments in Case 5.1.5 after (B.175) no longer uses the condition (c5.1), we can use the same argument verbatim and prove that the corner point in Case 5.2.6 is the Vertex 25 $(\frac{F_1+F_2+M_1}{2} - M_2, \frac{F_1+F_2-M_1}{2}, \frac{F_1+F_2+M_1}{2} - M_2, \frac{F_1+F_2-M_1}{2})$ listed in Table 3.3.

Case 6: We assume

$$F_1 < M_2 \leq M_1. \tag{c6}$$

Ineq. (c6) and $\mathcal{G}$ jointly describe the scenario when the (M_1, M_2) value falls into the upper triangle in Fig. 3.2 with solid edges and being marked as "Case 6". In this case, the a_1 to b_6 values of (A1) to (B6) become

$$\begin{aligned} &a_1 = a_2 = a_3 = a_4 = 0, \\ &b_1 = b_3 = b_4 = b_6 = F_1 + F_2 - M_2, \\ &b_2 = b_5 = F_1 + F_2 - M_1. \end{aligned} \tag{c6.ab}$$

Note that only a_1, a_3, and b_2 are different in (c6.ab) and (c5.ab), and the rest 7 values are identical. Therefore we can reuse the equations related to (c5.ab) when (A1), (A3) and (B2) are not involved. We consider the following 6 subcases.

Case 6.1: (A1) is tight. i.e., $\overline{\text{(A1)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}, \text{(B3)}\} \overset{\mathcal{G},\text{(c6)}}{\Longrightarrow} \{\text{(A2)}, \text{(A3)}, \text{(A4)}, \text{(B4)}, \text{(B5)}, \text{(B6)}\} \tag{B.184}$$

by and the following straightforward intermediate steps

$$\{\overline{\text{(A1)}}, \text{(B1)}\} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Longrightarrow} \text{(A2)} \tag{B.185}$$

$$\{\overline{\text{(A1)}}, \text{(B2)}\} \overset{\text{(c6.ab)},\text{(G5)}}{\Longrightarrow} \text{(A3)} \tag{B.186}$$

$$\{\overline{\text{(A1)}}, \text{(B3)}\} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Longrightarrow} \text{(A4)} \tag{B.187}$$

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B2)}\} \overset{\text{(c6.ab)},\text{(G5)}}{\Longrightarrow} \text{(B4)} \tag{B.188}$$

$$\{\overline{\text{(A1)}}, \text{(B2)}, \text{(B3)}\} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Longrightarrow} \text{(B5)} \tag{B.189}$$

$$\{\overline{\text{(A1)}}, \text{(B1)}, \text{(B3)}\} \overset{\text{(c6.ab)},\text{(G5)}}{\Longrightarrow} \text{(B6)}. \tag{B.190}$$

By (B.184), the four tight linear inequalities in Case-6.1 can only be (A1) (thus $\overline{\text{(A1)}}$), (B1), (B2), and (B3). Solving these equations, the corresponding corner point is *Vertex 26* $(0, F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2)$ listed in Table 3.3.

Case 6.2: (A2) is tight. i.e., $\overline{\text{(A2)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A2)}}, \text{(B1)}, \text{(B4)}, \text{(B5)}\} \overset{\mathcal{G},\text{(c6)}}{\Longrightarrow} \{\text{(A1)}, \text{(A3)}, \text{(A4)}, \text{(B2)}, \text{(B3)}, \text{(B6)}\} \tag{B.191}$$

by reusing (B.163), (B.165), (B.166), and the following straightforward intermediate steps

$$\{\overline{(A2)}, (B1)\} \overset{(c6.ab),(G3),(G5)}{\Longrightarrow} (A1) \tag{B.192}$$

$$\{\overline{(A2)}, (B4)\} \overset{(c6.ab),(G3),(G5)}{\Longrightarrow} (A3) \tag{B.193}$$

$$\{\overline{(A2)}, (B1), (B4)\} \overset{(c6.ab),(G3),(G5)}{\Longrightarrow} (B2) \tag{B.194}$$

By (B.191), the four tight linear inequalities in Case-6.2 can only be (A2) (thus $\overline{(A2)}$), (B1), (B4), and (B5). Solving these equations, the corresponding corner point is Vertex 24 $(F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2, F_1 + F_2 - M_1)$ listed in Table 3.3.

Case 6.3: (A3) is tight. i.e., $\overline{(A3)}$ holds. We can prove the following relationship

$$\{\overline{(A3)}, (B2), (B4), (B6)\} \overset{\mathcal{G},(c6)}{\Longrightarrow}$$
$$\{(A1), (A2), (A4), (B1), (B3), (B5)\} \tag{B.195}$$

by reusing the following straightforward intermediate steps

$$\{\overline{(A3)}, (B2)\} \overset{(c6.ab),(G5)}{\Longrightarrow} (A1) \tag{B.196}$$

$$\{\overline{(A3)}, (B4)\} \overset{(c6.ab),(G3),(G5)}{\Longrightarrow} (A2) \tag{B.197}$$

$$\{\overline{(A3)}, (B6)\} \overset{(c6.ab),(G3),(G5)}{\Longrightarrow} (A4) \tag{B.198}$$

$$\{\overline{(A3)}, (B2), (B4)\} \overset{(c6.ab),(G5)}{\Longrightarrow} (B1) \tag{B.199}$$

$$\{\overline{(A3)}, (B2), (B6)\} \overset{(c6.ab),(G5)}{\Longrightarrow} (B3) \tag{B.200}$$

$$\{\overline{(A3)}, (B4), (B6)\} \overset{(c6.ab),(G3),(G5)}{\Longrightarrow} (B5). \tag{B.201}$$

By (B.195), the four tight linear inequalities in Case-6.3 can only be (A3) (thus $\overline{(A3)}$), (B2), (B4), and (B6). Solving these equations, the corresponding corner point is *Vertex 27* $(F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2)$ listed in Table 3.3.

Case 6.4: (A4) is tight. i.e., $\overline{\text{(A4)}}$ holds. We can prove the following relationship

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B5)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c6)}}{\Longrightarrow} \{\text{(A1)}, \text{(A2)}, \text{(A3)}, \text{(B1)}, \text{(B2)}, \text{(B4)}\} \tag{B.202}$$

by reusing (B.172), (B.173), (B.174) and the following straightforward intermediate steps

$$\{\overline{\text{(A4)}}, \text{(B3)}\} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Longrightarrow} \text{(A1)} \tag{B.203}$$

$$\{\overline{\text{(A4)}}, \text{(B6)}\} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Longrightarrow} \text{(A3)} \tag{B.204}$$

$$\{\overline{\text{(A4)}}, \text{(B3)}, \text{(B6)}\} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Longrightarrow} \text{(B2)} \tag{B.205}$$

By (B.202), the four tight linear inequalities in Case-6.4 can only be (A4) (thus $\overline{\text{(A4)}}$), (B3), (B5), and (B6). Solving these equations, the corresponding corner point is Vertex 22 $(F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0)$ listed in Table 3.3.

Case 6.5: None of (A1) to (A4) is tight, but (B2) is tight, i.e., $\overline{\text{(B2)}}$ holds. We can prove

$$\{\text{(B1)}, \overline{\text{(B2)}}, \text{(B3)}, \text{(B4)}, \text{(B6)}\} \overset{\mathcal{G},\text{(c6)}}{\Rightarrow} \{\text{(B5)}\} \tag{B.206}$$

using the following intermediate step

$$\{\text{(B1)}, \text{(B6)}, \overline{\text{(B2)}}\} \overset{\text{(c6.ab)},\text{(G3)}}{\Rightarrow} \{\text{(B5)}\}. \tag{B.207}$$

That is, in this case, we do not need to check whether (B5) is tight or not. Since under the a_1 to b_6 values in (c6.ab), the inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent*, the corner point must be tight for all 5 inequalities (B1), $\overline{\text{(B2)}}$, (B3), (B4), (B6). Solving these 5 joint equations, the corresponding corner point is *Vertex 28*

Table B.1. Vertex 1 $(F_1-M_2, F_1+F_2-M_1, F_1+F_2-M_1, F_2)$ with applicable range: $M_2 \le M_1 \le \min(F_1, F_2+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	M_1-M_2	M_1-M_2	M_1-M_2	0	M_1-M_2	M_1-M_2	M_1-M_2	M_1-M_2
1.1.Cov	$F_1-M_1+M_2$	$F_2-M_1+M_2$	M_2	M_2	F_1-M_1	$F_1+F_2-2M_1+M_2$	$F_1+F_2-2M_1+M_2$	$F_2-M_1+M_2$
Total	F_1	F_2	M_1	M_2	F_1-M_2	$F_1+F_2-M_1$	$F_1+F_2-M_1$	F_2

$(\frac{F_1+F_2-M_1}{2}, \frac{F_1+F_2+M_1}{2} - M_2, \frac{F_1+F_2-M_1}{2}, \frac{F_1+F_2+M_1}{2} - M_2)$ listed in Table 3.3. Vertex 28 is a legitimate corner point in this case since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 28} \overset{\text{(c6.ab)},\text{(G5)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 28} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Rightarrow} \text{(A2)};$$
$$\text{Vertex 28} \overset{\text{(c6.ab)},\text{(G5)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 28} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Rightarrow} \text{(A4)}.$$

Case 6.6: None of (A1) to (A4) is tight, nor is (B2). Recall that under the a_1 to b_6 values in (c6.ab), inequalities corresponding to (B1), (B3), (B4), and (B6) are *co-dependent* in Case-6.5, a corner point requires all the remaining 5 inequalities (B1), (B3), (B4), (B5), and (B6) to be tight simultaneously. The 5 inequalities jointly yield exactly one corner point Vertex 25 $(\frac{F_1+F_2+M_1}{2} - M_2, \frac{F_1+F_2-M_1}{2}, \frac{F_1+F_2+M_1}{2} - M_2, \frac{F_1+F_2-M_1}{2})$ listed in Table 3.3. Vertex 25 is a legitimate corner point since it also satisfies (A1) to (A4) as well, i.e.,

$$\text{Vertex 25} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 25} \overset{\text{(c6.ab)},\text{(G5)}}{\Rightarrow} \text{(A2)};$$
$$\text{Vertex 25} \overset{\text{(c6.ab)},\text{(G3)},\text{(G5)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 25} \overset{\text{(c6.ab)},\text{(G5)}}{\Rightarrow} \text{(A4)}.$$

B.2 Proof of Proposition 3.3.3

In the following, we will prove that each of the 28 vertices can be achieved by space sharing among the 7 basic achievable schemes listed in Table 3.2.

Vertex 1: As summarized in Fig. 3.2, Vertex-1 rate vector $(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$ is the corner point for 4 out of 11 sub-regions. Table B.1 describes an achievable scheme that attains Vertex-1 rate vector as long as the following Applicable Range (AR) holds

$$\text{[AR]:} \quad M_2 \le M_1 \le \min(F_1, F_2 + M_2), \tag{B.208}$$

which is the union of the 4 desired sub-regions.

Each row of Table B.1 describes one basic scheme and the last row describes the total (combined) effect after space sharing. Each basic scheme takes parts of files 1 and 2 and stores coded data in parts of memories 1 and 2. The columns of f_1, f_2, m_1, and m_2 correspond to the amount of files 1 and 2 and memories 1 and 2 of each basic scheme. The columns of $R_{(1,1)}$, $R_{(1,2)}$, $R_{(2,1)}$, and $R_{(2,2)}$ correspond to the rates contributed by each basic scheme under each request pattern. The intersection of the "Total" row and the four rate columns thus represents the achievable rate vector of the overall scheme.

To ensure that Table B.1 indeed describes a legitimate scheme, one needs to verify the following three conditions:

1. *All the file sizes and the memory sizes are non-negative.* For example, 1.1.Cov uses $F_1 - M_1 + M_2$ of file 1 and $F_2 - M_1 + M_2$ of file 2 to encode. The AR (B.208) ensures that both sub-file sizes are non-negative.

2. *For any given row, the assigned subfile sizes and the assigned memory sizes satisfy the required condition of the basic scheme listed in Table 3.2.* For example, as summarized in Table 3.2 the 1.1.Cov scheme requires that $\max(m_1, m_2) \leq f_1$. As a result, in the row of 1.1.Cov in Table B.1 we must satisfy $\max(M_2, M_2) \leq F_1 - M_1 + M_2$, which is ensured by the AR (B.208).

3. *For any given row, the delivery rate vector is computed correctly according to Table 3.2.* For example, as summarized in Table 3.2 the 1.1.Cov scheme achieves $R_{(1,1)} = f_1 - \min(m_1, m_2)$. As a result, correct rate computation in the row of 1.1.Cov of Table B.1 requires the equality $F_1 - M_1 = (F_1 - M_1 + M_2) - \min(M_2, M_2)$ to hold, which is ensured by[2] the AR (B.208).

Verifying these three statements are very straightforward and we thus omit the details here.

Remark: The main reason that the Vertex-1 scheme in Table B.1 is companied by an AR condition (B.208) is to ensure that the above three conditions about file/memory sizes and rate computation are properly met.

[2]↑In this example, it is trivially true regardless whether the AR condition (B.208) holds or not.

Table B.2. Vertex 2 $(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$ with applicable range: $M_1 + M_2 \leq \min(F_1, F_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	M_1	M_1	M_1	0	M_1	M_1	M_1	M_1
Emp.Mix	M_2	M_2	0	M_2	M_2	M_2	M_2	M_2
1.1.Cov	$F_1-M_1-M_2$	$F_2-M_1-M_2$	0	0	$F_1-M_1-M_2$	$F_1+F_2-2M_1-2M_2$	$F_1+F_2-2M_1-2M_2$	$F_2-M_1-M_2$
Total	F_1	F_2	M_1	M_2	F_1	$F_1+F_2-M_1-M_2$	$F_1+F_2-M_1-M_2$	F_2

Table B.3. Vertex 3 $(F_1, F_1+F_2-M_1, F_1+F_2-M_1, F_2-M_2)$ with applicable range: $M_2 \le M_1 \le \min(F_1+M_2, F_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	M_1-M_2	M_1-M_2	M_1-M_2	0	M_1-M_2	M_1-M_2	M_1-M_2	M_1-M_2
2.2.Cov	$F_1-M_1+M_2$	$F_2-M_1+M_2$	M_2	M_2	$F_1-M_1+M_2$	$F_1+F_2-2M_1+M_2$	$F_1+F_2-2M_1+M_2$	F_2-M_1
Total	F_1	F_2	M_1	M_2	F_1	$F_1+F_2-M_1$	$F_1+F_2-M_1$	F_2-M_2

Table B.4. Vertex 4 $(F_1 - {}^1\!/_2 M_2, F_1 + F_2 - M_1 - {}^1\!/_2 M_2, F_1 + F_2 - M_1 - {}^1\!/_2 M_2, F_2 - {}^1\!/_2 M_2)$ with applicable range: $M_2 \le M_1 \le \min(F_1, F_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	M_1-M_2	M_1-M_2	M_1-M_2	0	M_1-M_2	M_1-M_2	M_1-M_2	M_1-M_2
Ha.Fi	M_2	M_2	M_2	M_2	$M_2/2$	$M_2/2$	$M_2/2$	$M_2/2$
1.1.Cov	F_1-M_1	F_2-M_1	0	0	F_1-M_1	$F_1+F_2-2M_1$	$F_1+F_2-2M_1$	F_2-M_1
Total	F_1	F_2	M_1	M_2	$F_1-\frac{1}{2}M_2$	$F_1+F_2-M_1-\frac{1}{2}M_2$	$F_1+F_2-M_1-\frac{1}{2}M_2$	$F_2-\frac{1}{2}M_2$

Vertex 2: As summarized in Fig. 3.2, Vertex-2 rate vector $(F_1, F_1+F_2-M_1-M_2, F_1+F_2-M_1-M_2, F_2)$ is the corner point of 1 out of 11 sub-regions. Table B.2 describes the corresponding space-sharing scheme that attains Vertex-2 rate vector. One can easily verify that the applicable range listed in Table B.2, i.e. $M_1+M_2 \le \min(F_1, F_2)$, completely covers the desired sub-region. Furthermore, the applicable range ensures that the three conditions on the file/memory sizes and rate computation are met, also see the discussion of Vertex 1. Since the verification step is straightforward, we omit the details. The proof of Vertex-2 achievability is thus complete.

Remark: As will be seen later, each of our schemes, described in the corresponding table, is associated with an applicable range. To prove the legitimacy of the scheme, we always have to check (i) the applicable range covers the sub-regions of the corresponding corner point; and (ii) the applicable range ensures that the three conditions on the file/memory sizes and rate computation are met. Since checking (i) and (ii) can all be verified very easily, we will not repeatedly emphasize these important verification steps in the sequel. Instead we only describe the schemes and the corresponding applicable ranges.

Vertex 3: Per Fig. 3.2, Vertex-3 rate vector $(F_1, F_1+F_2-M_1, F_1+F_2-M_1, F_2-M_2)$ is the corner point of 2 out of 11 sub-regions. Table B.3 describes an achievable scheme that attains Vertex-3 rate vector and its applicable range.

Table B.5. Vertex 5 $(F_1, F_1+F_2-M_1-M_2, F_1, F_2)$ with applicable range: $\max(M_1, M_2) \leq F_2 \leq \min(F_1, M_1+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
Emp.Mix	F_2-M_1	F_2-M_1	0	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1
1.2.Cov	F_1-2F_2 $+M_1+M_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	F_1-2F_2 $+M_1+M_2$	F_1-F_2	F_1-2F_2 $+M_1+M_2$	$M_1+M_2-F_2$
Total	F_1	F_2	M_1	M_2	F_1	$F_1+F_2-M_1-M_2$	F_1	F_2

Table B.6. Vertex 6 $(F_1, F_1, F_1+F_2-M_1-M_2, F_2)$ with applicable range: $\max(M_1, M_2) \leq F_2 \leq \min(F_1, M_1+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
Emp.Mix	F_2-M_1	F_2-M_1	0	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1
2.1.Cov	F_1-2F_2 $+M_1+M_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	F_1-2F_2 $+M_1+M_2$	F_1-2F_2 $+M_1+M_2$	F_1-F_2	$M_1+M_2-F_2$
Total	F_1	F_2	M_1	M_2	F_1	F_1	$F_1+F_2-M_1-M_2$	F_2

Table B.7. Vertex 7 $(F_1+1/2(F_2-M_1-M_2), F_1+1/2(F_2-M_1-M_2), F_1+1/2(F_2-M_1-M_2), F_2+1/2(F_2-M_1-M_2))$ with applicable range: $\max(M_1, M_2) \leq F_2 \leq \min(F_1, M_1+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
Emp.Mix	F_2-M_1	F_2-M_1	0	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1
Ha.Fi	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$\frac{M_1+M_2-F_2}{2}$	$\frac{M_1+M_2-F_2}{2}$	$\frac{M_1+M_2-F_2}{2}$	$\frac{M_1+M_2-F_2}{2}$
1.1.Cov	F_1-F_2	0	0	0	F_1-F_2	F_1-F_2	F_1-F_2	0
Total	F_1	F_2	M_1	M_2	$F_1+\frac{F_2-M_1-M_2}{2}$	$F_1+\frac{F_2-M_1-M_2}{2}$	$F_1+\frac{F_2-M_1-M_2}{2}$	$F_2+\frac{F_2-M_1-M_2}{2}$

Vertex 4: Per Fig. 3.2, Vertex-4 rate vector $(F_1-\frac{M_2}{2}, F_1+F_2-M_1-\frac{M_2}{2}, F_1+F_2-M_1-\frac{M_2}{2}, F_2-\frac{M_2}{2})$ is the corner point of 2 out of 11 sub-regions. Table B.4 describes an achievable scheme that attains Vertex-4 rate vector and its applicable range.

Vertex 5: Per Fig. 3.2, Vertex-5 rate vector $(F_1, F_1+F_2-M_1-M_2, F_1, F_2)$ is the corner point of 1 out of 11 sub-regions. Table B.5 describes an achievable scheme that attains Vertex-5 rate vector and its applicable range.

Vertex 6: Per Fig. 3.2, Vertex-6 rate vector $(F_1, F_1, F_1+F_2-M_1-M_2, F_2)$ is the corner point of 1 out of 11 sub-regions. Table B.6 describes an achievable scheme that attains Vertex-6 rate vector and its applicable range.

Table B.8. Vertex 8 $(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ with applicable range: $F_2 + M_2 \leq M_1 \leq F_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2	F_2	F_2	0	F_2	F_2	F_2	F_2
1.1.Cov	F_1-F_2	0	M_1-F_2	M_2	$F_1-F_2-M_2$	F_1-M_1	$F_1-F_2-M_2$	0
Total	F_1	F_2	M_1	M_2	F_1-M_2	$F_1+F_2-M_1$	F_1-M_2	F_2

Table B.9. Vertex 9 $(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$ with applicable range: $M_2 \leq F_2 \leq M_1 \leq F_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
1.2.Cov	$F_1-F_2+M_2$	M_2	$M_1+M_2-F_2$	M_2	$F_1-F_2+M_2$	F_1-M_1	$F_1-F_2+M_2$	M_2
Total	F_1	F_2	M_1	M_2	F_1	$F_1+F_2-M_1-M_2$	F_1	F_2

Table B.10. Vertex 10 $(F_1, F_1 + F_2 - M_1, F_1, F_2 - M_2)$ with applicable range: $M_2 \leq F_2 \leq M_1 \leq F_1 + M_2$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
1.1.Cov	$F_1-F_2+M_2$	0	M_1-F_2	0	$F_1-F_2+M_2$	$F_1-M_1+M_2$	$F_1-F_2+M_2$	0
2.2.Cov	0	M_2	M_2	M_2	0	0	0	0
Total	F_1	F_2	M_1	M_2	F_1	$F_1+F_2-M_1$	F_1	F_2-M_2

Vertex 7: Per Fig. 3.2, Vertex-7 rate vector $(F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, F_2 + \frac{F_2-M_1-M_2}{2})$ is the corner point of 1 out of 11 sub-regions. Table B.7 describes an achievable scheme that attains Vertex-7 rate vector and its applicable range.

Vertex 8: Per Fig. 3.2, Vertex-8 rate vector $(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$ is the corner point of 2 out of 11 sub-regions when $F_1 \geq 2F_2$ and of 1 out of 11 sub-regions when $F_1 < 2F_2$. Table B.8 describes an achievable scheme that attains Vertex-8 rate vector and its applicable range.

Vertex 9: Per Fig. 3.2, Vertex-9 rate vector $(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$ is the corner point of 2 out of 11 sub-regions. Table B.9 describes an achievable scheme that attains Vertex-9 rate vector and its applicable range.

Vertex 10: Per Fig. 3.2, Vertex-10 rate vector $(F_1, F_1 + F_2 - M_1, F_1, F_2 - M_2)$ is the corner point of 3 out of 11 sub-regions when $F_1 \geq 2F_2$ and of 4 out of 11 sub-regions when $F_1 < 2F_2$. Table B.10 describes an achievable scheme that attains Vertex-10 rate vector and its applicable range.

Table B.11. Vertex 11 ($F_1 - 1/2 M_2, F_1 + F_2 - M_1 - 1/2 M_2, F_1 - 1/2 M_2, F_2 - 1/2 M_2$) with applicable range: $M_2 \leq F_2 \leq M_1 \leq F_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	$F_2 - M_2$	$F_2 - M_2$	$F_2 - M_2$	0	$F_2 - M_2$	$F_2 - M_2$	$F_2 - M_2$	$F_2 - M_2$
Ha.Fi	M_2	M_2	M_2	M_2	$M_2/2$	$M_2/2$	$M_2/2$	$M_2/2$
1.1.Cov	$F_1 - F_2$	0	$M_1 - F_2$	0	$F_1 - F_2$	$F_1 - M_1$	$F_1 - F_2$	0
Total	F_1	F_2	M_1	M_2	$F_1 - \frac{1}{2} M_2$	$F_1 + F_2 - M_1 - \frac{1}{2} M_2$	$F_1 - \frac{1}{2} M_2$	$F_2 - \frac{1}{2} M_2$

Table B.12. Vertex 12 $(F_1+F_2-M_1, F_1+F_2-M_1, F_1-M_2, F_2)$ with applicable range: $\max(M_2, F_2) \le M_1 \le \min(F_1, F_2+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	M_1-M_2	M_1-M_2	M_1-M_2	0	M_1-M_2	M_1-M_2	M_1-M_2	M_1-M_2
1.1.Cov	F_1-F_2	0	M_1-F_2	M_1-F_2	F_1-M_1	F_1-M_1	F_1-M_1	0
2.1.Cov	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	0	$F_2-M_1+M_2$
Total	F_1	F_2	M_1	M_2	$F_1+F_2-M_1$	$F_1+F_2-M_1$	F_1-M_2	F_2

Table B.13. Vertex 13 $(F_1+1/2(F_2-M_1-M_2), F_1+1/2(F_2-M_1-M_2), F_1+1/2(F_2-M_1-M_2), 1/2(F_2+M_1-M_2))$ with applicable range: $\max(M_2, F_2) \le M_1 \le \min(F_1, F_2+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	M_1-M_2	M_1-M_2	M_1-M_2	0	M_1-M_2	M_1-M_2	M_1-M_2	M_1-M_2
Ha.Fi	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$\frac{F_2-M_1+M_2}{2}$	$\frac{F_2-M_1+M_2}{2}$	$\frac{F_2-M_1+M_2}{2}$	$\frac{F_2-M_1+M_2}{2}$
1.1.Cov	F_1-F_2	0	M_1-F_2	M_1-F_2	F_1-M_1	F_1-M_1	F_1-M_1	0
Total	F_1	F_2	M_1	M_2	$F_1+\frac{F_2-M_1-M_2}{2}$	$F_1+\frac{F_2-M_1-M_2}{2}$	$F_1+\frac{F_2-M_1-M_2}{2}$	$\frac{F_2+M_1-M_2}{2}$

Table B.14. Vertex 14 $(F_1-M_2, F_2, F_1-M_2, F_2)$ with applicable range: $\max(F_1, F_2+M_2) \le M_1 \le F_1+F_2$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	0	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$
1.1.Cov	M_1-F_2	0	M_1-F_2	M_2	$M_1-F_2-M_2$	0	$M_1-F_2-M_2$	0
2.2.Cov	0	M_1-F_1	M_1-F_1	0	0	M_1-F_1	0	M_1-F_1
Total	F_1	F_2	M_1	M_2	F_1-M_2	F_2	F_1-M_2	F_2

Vertex 11: Per Fig. 3.2, Vertex-11 rate vector $(F_1-\frac{M_2}{2}, F_1+F_2-M_1-\frac{M_2}{2}, F_1-\frac{M_2}{2}, F_2-\frac{M_2}{2})$ is the corner point of 2 out of 11 sub-regions. Table B.11 describes an achievable scheme that attains Vertex-11 rate vector and its applicable range.

Vertex 12: Per Fig. 3.2, Vertex-12 rate vector $(F_1+F_2-M_1, F_1+F_2-M_1, F_1-M_2, F_2)$ is the corner point of 2 out of 11 sub-regions. Table B.12 describes an achievable scheme that attains Vertex-12 rate vector and its applicable range.

Vertex 13: Per Fig. 3.2, Vertex-13 rate vector $(F_1+\frac{F_2-M_1-M_2}{2}, F_1+\frac{F_2-M_1-M_2}{2}, F_1+\frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2})$ is the corner point of 2 out of 11 sub-regions. Table B.13 describes an achievable scheme that attains Vertex-13 rate vector and its applicable range.

Vertex 14: Per Fig. 3.2, Vertex-14 rate vector $(F_1-M_2, F_2, F_1-M_2, F_2)$ is the corner point of 3 out of 11 sub-regions. Table B.14 describes an achievable scheme that attains Vertex-14 rate vector and its applicable range.

Table B.15. Vertex 15 $(F_1, F_2 - M_2, F_1, F_2 - M_2)$ with applicable range: $M_2 \leq F_2 \leq F_1 + M_2 \leq M_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
1.1.Cov	$F_1-F_2+M_2$	0	$F_1-F_2+M_2$	0	$F_1-F_2+M_2$	0	$F_1-F_2+M_2$	0
2.2.Cov	0	M_2	M_2	M_2	0	0	0	0
Total	F_1	F_2	$F_1 + M_2$	M_2	F_1	F_2-M_2	F_1	F_2-M_2

Table B.16. Vertex 16 $(F_1, F_2 - M_2, F_1, F_1 + F_2 - M_1)$ with applicable range: $F_2 - F_1 \leq M_2 \leq F_2$ and $F_1 \leq M_1 \leq F_1 + M_2$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
1.2.Cov	$F_1-F_2+M_2$	$F_1-M_1+M_2$	$F_1-F_2+M_2$	$F_1-M_1+M_2$	$F_1-F_2+M_2$	0	$F_1-F_2+M_2$	$F_1-M_1+M_2$
2.2.Cov	0	M_1-F_1	M_1-F_1	M_1-F_1	0	0	0	0
Total	F_1	F_2	M_1	M_2	F_1	F_2-M_2	F_1	$F_1+F_2-M_1$

Table B.17. Vertex 17 $(1/2(F_1+M_1-M_2), F_2+1/2(F_1-M_1-M_2), 1/2(F_1+M_1-M_2), F_2+1/2(F_1-M_1-M_2))$ with applicable range: $M_2 \leq F_2 \leq M_1 \leq F_1+M_2$ and $F_1 \leq M_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
Ha.Fi	$F_1-M_1+M_2$	$F_1-M_1+M_2$	$F_1-M_1+M_2$	$F_1-M_1+M_2$	$\frac{F_1-M_1+M_2}{2}$	$\frac{F_1-M_1+M_2}{2}$	$\frac{F_1-M_1+M_2}{2}$	$\frac{F_1-M_1+M_2}{2}$
1.1.Cov	M_1-F_2	0	M_1-F_2	0	M_1-F_2	0	M_1-F_2	0
2.2.Cov	0	M_1-F_1	M_1-F_1	M_1-F_1	0	0	0	0
Total	F_1	F_2	M_1	M_2	$\frac{F_1+M_1-M_2}{2}$	$F_2+\frac{F_1-M_1-M_2}{2}$	$\frac{F_1+M_1-M_2}{2}$	$F_2+\frac{F_1-M_1-M_2}{2}$

Table B.18. Vertex 18 $(F_1 - M_2, F_2, F_1 + F_2 - M_1, F_2)$ with applicable range: $M_2 \leq F_1 \leq M_1 \leq F_2 + M_2$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_1-M_2	F_1-M_2	F_1-M_2	0	F_1-M_2	F_1-M_2	F_1-M_2	F_1-M_2
1.1.Cov	M_2	0	M_2	M_2	0	0	0	0
2.2.Cov	0	$F_2-F_1+M_2$	M_1-F_1	0	0	$F_2-F_1+M_2$	$F_2-M_1+M_2$	$F_2-F_1+M_2$
Total	F_1	F_2	M_1	M_2	F_1-M_2	F_2	$F_1+F_2-M_1$	F_2

Vertex 15: Per Fig. 3.2, Vertex-15 rate vector $(F_1, F_2 - M_2, F_1, F_2 - M_2)$ is the corner point of 1 out of 11 sub-regions. Table B.15 describes an achievable scheme that attains Vertex-15 rate vector and its applicable range.

Vertex 16: Per Fig. 3.2, Vertex-16 rate vector $(F_1, F_2 - M_2, F_1, F_1 + F_2 - M_1)$ is the corner point of 1 out of 11 sub-regions when $F_1 \geq 2F_2$ and of 2 out of 11 sub-regions when $F_1 < 2F_2$. Table B.16 describes an achievable scheme that attains Vertex-16 rate vector and its applicable range.

Vertex 17: Per Fig. 3.2, Vertex-17 rate vector $(\frac{F_1+M_1-M_2}{2}, F_2 + \frac{F_1-M_1-M_2}{2}, \frac{F_1+M_1-M_2}{2}, F_2 + \frac{F_1-M_1-M_2}{2})$ is the corner point of 1 out of 11 sub-regions when $F_1 \geq 2F_2$ and of 2 out of 11 sub-regions when $F_1 < 2F_2$. Table B.17 describes an achievable scheme that attains Vertex-17 rate vector and its applicable range.

Table B.19. Vertex 19 $(F_1+F_2-M_1, F_2, F_1-M_2, F_2)$ with applicable range: $M_2 \leq F_1 \leq M_1 \leq F_2+M_2$ and $F_2 \leq M_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_1-M_2	F_1-M_2	F_1-M_2	0	F_1-M_2	F_1-M_2	F_1-M_2	F_1-M_2
1.1.Cov	M_1-F_2	0	M_1-F_2	M_1-F_2	0	0	0	0
2.1.Cov	$F_2-M_1+M_2$	$F_2-F_1+M_2$	$F_2-F_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-F_1+M_2$	0	$F_2-F_1+M_2$
Total	F_1	F_2	M_1	M_2	$F_1+F_2-M_1$	F_2	F_1-M_2	F_2

Table B.20. Vertex 20 $(F_1+1/2(F_2-M_1-M_2), 1/2(F_2+M_1-M_2), F_1+1/2(F_2-M_1-M_2), 1/2(F_2+M_1-M_2))$ with applicable range: $M_2 \leq F_1 \leq M_1 \leq F_2+M_2$ and $F_2 \leq M_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_1-M_2	F_1-M_2	F_1-M_2	0	F_1-M_2	F_1-M_2	F_1-M_2	F_1-M_2
Ha.Fi	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$F_2-M_1+M_2$	$\frac{F_2-M_1+M_2}{2}$	$\frac{F_2-M_1+M_2}{2}$	$\frac{F_2-M_1+M_2}{2}$	$\frac{F_2-M_1+M_2}{2}$
1.1.Cov	M_1-F_2	0	M_1-F_2	M_1-F_2	0	0	0	0
2.2.Cov	0	M_1-F_1	M_1-F_1	0	0	M_1-F_1	0	M_1-F_1
Total	F_1	F_2	M_1	M_2	$F_1+\frac{F_2-M_1-M_2}{2}$	$\frac{F_2+M_1-M_2}{2}$	$F_1+\frac{F_2-M_1-M_2}{2}$	$\frac{F_2+M_1-M_2}{2}$

Table B.21. Vertex 21 $(F_1+F_2-M_2, F_1-M_1, F_1+F_2-M_2, F_2)$ with applicable range: $F_2 \leq M_2 \leq M_1 \leq F_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
1.1.Cov	F_1-F_2	0	M_1-F_2	M_2-F_2	F_1-M_2	F_1-M_1	F_1-M_2	0
1.2.Cov	F_2	F_2	F_2	F_2	F_2	0	F_2	F_2
Total	F_1	F_2	M_1	M_2	$F_1+F_2-M_2$	F_1-M_1	$F_1+F_2-M_2$	F_2

Vertex 18: Per Fig. 3.2, Vertex-18 rate vector $(F_1 - M_2, F_2, F_1 + F_2 - M_1, F_2)$ is the corner point of 1 out of 11 sub-regions when $F_1 \geq 2F_2$ and of 2 out of 11 sub-regions when $F_1 < 2F_2$. Table B.18 describes an achievable scheme that attains Vertex-18 rate vector and its applicable range.

Vertex 19: Per Fig. 3.2, Vertex-19 rate vector $(F_1 + F_2 - M_1, F_2, F_1 - M_2, F_2)$ is the corner point of 1 out of 11 sub-regions when $F_1 \geq 2F_2$ and of 2 out of 11 sub-regions when $F_1 < 2F_2$. Table B.19 describes an achievable scheme that attains Vertex-19 rate vector and its applicable range.

Vertex 20: Per Fig. 3.2, Vertex-20 rate vector $(F_1+\frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2}, F_1+\frac{F_2-M_1-M_2}{2}, \frac{F_2+M_1-M_2}{2})$ is the corner point of 1 out of 11 sub-regions when $F_1 \geq 2F_2$ and of 2 out of 11 sub-regions when $F_1 < 2F_2$. Table B.20 describes an achievable scheme that attains Vertex-20 rate vector and its applicable range.

Table B.22. Vertex 22 $(F_1+F_2-M_2, F_1+F_2-M_1, F_1+F_2-M_2, 0)$ with applicable range: $F_2 \le M_2 \le M_1 \le F_1+F_2$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
1.1.Cov	F_1	0	M_1-F_2	M_2-F_2	$F_1+F_2-M_2$	$F_1+F_2-M_1$	$F_1+F_2-M_2$	0
2.2.Cov	0	F_2	F_2	F_2	0	0	0	0
Total	F_1	F_2	M_1	M_2	$F_1+F_2-M_2$	$F_1+F_2-M_1$	$F_1+F_2-M_2$	0

Table B.23. Vertex 23 $(F_1+1/2F_2-M_2, F_1+1/2F_2-M_1, F_1+1/2F_2-M_2, 1/2F_2)$ with applicable range: $F_2 \le M_2 \le M_1 \le F_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Ha.Fi	F_2	F_2	F_2	F_2	$F_2/2$	$F_2/2$	$F_2/2$	$F_2/2$
1.1.Cov	F_1-F_2	0	M_1-F_2	M_2-F_2	F_1-M_2	F_1-M_1	F_1-M_2	0
Total	F_1	F_2	M_1	M_2	$F_1+\frac{1}{2}F_2-M_2$	$F_1+\frac{1}{2}F_2-M_1$	$F_1+\frac{1}{2}F_2-M_2$	$\frac{1}{2}F_2$

Table B.24. Vertex 24 $(F_1+F_2-M_2, 0, F_1+F_2-M_2, F_1+F_2-M_1)$ with applicable range: $F_2 \le M_2 \le M_1 \le F_1+F_2$ and $F_1 \le M_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
1.1.Cov	M_1-F_2	0	M_1-F_2	M_2-F_2	M_1-M_2	0	M_1-M_2	0
1.2.Cov	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	0	$F_1+F_2-M_1$	$F_1+F_2-M_1$
2.2.Cov	0	M_1-F_1	M_1-F_1	M_1-F_1	0	0	0	0
Total	F_1	F_2	M_1	M_2	$F_1+F_2-M_2$	0	$F_1+F_2-M_2$	$F_1+F_2-M_1$

Vertex 21: Per Fig. 3.2, Vertex-21 rate vector $(F_1+F_2-M_2, F_1-M_1, F_1+F_2-M_2, F_2)$ is the corner point of 2 out of 11 sub-regions when $F_1 \ge 2F_2$ and of 1 out of 11 sub-regions when $F_1 < 2F_2$. Table B.21 describes an achievable scheme that attains Vertex-21 rate vector and its applicable range.

Vertex 22: Per Fig. 3.2, Vertex-22 rate vector $(F_1+F_2-M_2, F_1+F_2-M_1, F_1+F_2-M_2, 0)$ is the corner point of 5 out of 11 sub-regions when $F_1 \ge 2F_2$ and of 4 out of 11 sub-regions when $F_1 < 2F_2$. Table B.22 describes an achievable scheme that attains Vertex-22 rate vector and its applicable range.

Vertex 23: Per Fig. 3.2, Vertex-23 rate vector $(F_1+\frac{F_2}{2}-M_2, F_1+\frac{F_2}{2}-M_1, F_1+\frac{F_2}{2}-M_2, \frac{F_2}{2})$ is the corner point of 2 out of 11 sub-regions when $F_1 \ge 2F_2$ and of 1 out of 11 sub-regions when $F_1 < 2F_2$. Table B.23 describes an achievable scheme that attains Vertex-23 rate vector and its applicable range.

Table B.25. Vertex 25 $(1/2(F_1+F_2+M_1)-M_2, 1/2(F_1+F_2-M_1), 1/2(F_1+F_2+M_1)-M_2, 1/2(F_1+F_2-M_1))$ with applicable range: $F_2 \le M_2 \le M_1 \le F_1+F_2$ and $F_1 \le M_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Ha.Fi	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2-M_1}{2}$
1.1.Cov	M_1-F_2	0	M_1-F_2	M_2-F_2	M_1-M_2	0	M_1-M_2	0
2.2.Cov	0	M_1-F_1	M_1-F_1	M_1-F_1	0	0	0	0
Total	F_1	F_2	M_1	M_2	$\frac{F_1+F_2+M_1}{2}-M_2$	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2+M_1}{2}-M_2$	$\frac{F_1+F_2-M_1}{2}$

Table B.26. Vertex 26 $(0, F_1+F_2-M_2, F_1+F_2-M_1, F_1+F_2-M_2)$ with applicable range: $F_1 \le M_2 \le M_1 \le F_1+F_2$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
1.1.Cov	F_1	0	F_1	F_1	0	0	0	0
2.2.Cov	0	F_2	M_1-F_1	M_2-F_1	0	$F_1+F_2-M_2$	$F_1+F_2-M_1$	$F_1+F_2-M_2$
Total	F_1	F_2	M_1	M_2	0	$F_1+F_2-M_2$	$F_1+F_2-M_1$	$F_1+F_2-M_2$

Table B.27. Vertex 27 $(F_1+F_2-M_1, F_1+F_2-M_2, 0, F_1+F_2-M_2)$ with applicable range: $F_1 \le M_2 \le M_1 \le F_1+F_2$ and $F_2 \le M_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
1.1.Cov	M_1-F_2	0	M_1-F_2	M_1-F_2	0	0	0	0
2.1.Cov	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	0	$F_1+F_2-M_1$
2.2.Cov	0	M_1-F_1	M_1-F_1	M_2-F_1	0	M_1-M_2	0	M_1-M_2
Total	F_1	F_2	M_1	M_2	$F_1+F_2-M_1$	$F_1+F_2-M_2$	0	$F_1+F_2-M_2$

Vertex 24: Per Fig. 3.2, Vertex-24 rate vector $(F_1+F_2-M_2, 0, F_1+F_2-M_2, F_1+F_2-M_1)$ is the corner point of 3 out of 11 sub-regions. Table B.24 describes an achievable scheme that attains Vertex-24 rate vector and its applicable range.

Vertex 25: Per Fig. 3.2, Vertex-25 rate vector $(\frac{F_1+F_2+M_1}{2}-M_2, \frac{F_1+F_2-M_1}{2}, \frac{F_1+F_2+M_1}{2}-M_2, \frac{F_1+F_2-M_1}{2})$ is the corner point of 3 out of 11 sub-regions. Table B.25 describes an achievable scheme that attains Vertex-25 rate vector and its applicable range.

Vertex 26: Per Fig. 3.2, Vertex-26 rate vector $(0, F_1+F_2-M_2, F_1+F_2-M_1, F_1+F_2-M_2)$ is the corner point of 1 out of 11 sub-regions. Table B.26 describes an achievable scheme that attains Vertex-26 rate vector and its applicable range.

Vertex 27: Per Fig. 3.2, Vertex-27 rate vector $(F_1+F_2-M_1, F_1+F_2-M_2, 0, F_1+F_2-M_2)$ is the corner point of 1 out of 11 sub-regions. Table B.27 describes an achievable scheme that attains Vertex-27 rate vector and its applicable range.

Table B.28. Vertex 28 $(1/2(F_1+F_2-M_1), 1/2(F_1+F_2+M_1)-M_2, 1/2(F_1+F_2-M_1), 1/2(F_1+F_2+M_1)-M_2)$ with applicable range: $F_1 \leq M_2 \leq M_1 \leq F_1+F_2$ and $F_2 \leq M_1$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Ha.Fi	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$F_1+F_2-M_1$	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2-M_1}{2}$
1.1.Cov	M_1-F_2	0	M_1-F_2	M_1-F_2	0	0	0	0
2.2.Cov	0	M_1-F_1	M_1-F_1	M_2-F_1	0	M_1-M_2	0	M_1-M_2
Total	F_1	F_2	M_1	M_2	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2+M_1}{2}$ $-M_2$	$\frac{F_1+F_2-M_1}{2}$	$\frac{F_1+F_2+M_1}{2}$ $-M_2$

Vertex 28: Per Fig. 3.2, Vertex-28 rate vector $(\frac{F_1+F_2-M_1}{2}, \frac{F_1+F_2+M_1}{2} - M_2, \frac{F_1+F_2-M_1}{2}, \frac{F_1+F_2+M_1}{2} - M_2)$ is the corner point of 1 out of 11 sub-regions. Table B.28 describes an achievable scheme that attains Vertex-28 rate vector and its applicable range.

B.3 Proof of Corollary 3

The uniform average rate capacity $\tilde{R} = \frac{1}{4}(R_{(1,1)} + R_{(1,2)} + R_{(2,1)} + R_{(2,2)})$ is a special case of the average rate with popularity $p_{\vec{d}} = 0.25$ for all $\vec{d} \in \{1, 2\}$. In the following, we use Proposition 3.3.1 to derive the closed-form expression of the uniform average rate capacity as in Fig. 3.7.

We combine (B1) and (B6) to obtain

$$\begin{aligned}\tilde{R} \geq \frac{1}{4} \max\{&2F_1 + 2F_2 - 2M_2, 3F_1 + 2F_2 - M_1 - 2M_2, \\ &2F_1 + 3F_2 - M_1 - 2M_2, 3F_1 + 3F_2 - 2M_1 - 2M_2\}\end{aligned} \tag{B.209}$$

and combine (B2) and (B5) to obtain

$$\begin{aligned}\tilde{R} \geq \frac{1}{4} \max\{&2F_1 + 2F_2 - 2M_1, 3F_1 + 2F_2 - 2M_1 - M_2, \\ &2F_1 + 3F_2 - 2M_1 - M_2, 3F_1 + 3F_2 - 2M_1 - 2M_2\}.\end{aligned} \tag{B.210}$$

Therefore, the uniform average rate $\tilde{R}$ must simultaneously satisfy (B.209) and (B.210), which can be expanded as a set of 7 inequalities[3] as follows.

$$\tilde{R} \geq \frac{F_1 + F_2 - M_1}{2} \tag{P1}$$
$$\tilde{R} \geq \frac{F_1 + F_2 - M_2}{2} \tag{P2}$$
$$\tilde{R} \geq \frac{3F_1 + 2F_2 - 2M_1 - M_2}{4} \tag{P3}$$
$$\tilde{R} \geq \frac{2F_1 + 3F_2 - 2M_1 - M_2}{4} \tag{P4}$$
$$\tilde{R} \geq \frac{3F_1 + 2F_2 - M_1 - 2M_2}{4} \tag{P5}$$
$$\tilde{R} \geq \frac{2F_1 + 3F_2 - M_1 - 2M_2}{4} \tag{P6}$$
$$\tilde{R} \geq \frac{3F_1 + 3F_2 - 2M_1 - 2M_2}{4}. \tag{P7}$$

Now we show that the set of 7 inequalities is sufficient to describe the uniform average rate capacity. Note that the joint 7 inequalities (P1) to (P7) are symmetric with respect to the file and user indices; therefore without loss of generality, we assume $F_1 \geq F_2$ and $M_1 \geq M_2$. Under this assumption, (P1), (P3), (P4) and (P6) are always loose and the remaining 3 inequalities (P2), (P5), and (P7) jointly form the lower-triangular region ($M_1 \geq M_2$) in Fig. 3.7.

We now provide the proof of the achievability part. If we describe each corner point of the triangular region of $M_1 \geq M_2$ by the corresponding tuple $(M_1, M_2, \bar{R})$, then there are 7 vertices in the region, and they are

$$\begin{aligned}
&v_1 = (0, 0, \frac{3F_1 + 3F_2}{4}),\ v_2 = (F_2, 0, \frac{3F_1 + F_2}{4}),\\
&v_3 = (F_2, F_2, \frac{3F_1 - F_2}{4}),\ v_4 = (F_1, 0, \frac{F_1 + F_2}{2}),\\
&v_5 = (F_1, F_1, \frac{F_2}{2}),\ v_6 = (F_1 + F_2, 0, \frac{F_1 + F_2}{2}),\\
&\text{and } v_7 = (F_1 + F_2, F_1 + F_2, 0).
\end{aligned} \tag{B.211}$$

[3]↑When assuming $F_1 \geq F_2$, (P4) and (P6) are always loose. The remaining 5 inequalities are indeed the 5 facets described in Fig. 3.7.

The vertices v_1 and v_7 can be achieved by some trivial schemes. The rest of them, v_2, v_3, v_4, v_5, and v_6, can be achieved by Vertices 1, 4, 8, 23, and 14 described in Tables B.1, B.4, B.8, B.23, and B.14, respectively.

B.4 Re-derivation of Worst-Case Rate Capacity in [41]

Corollary 7. *The 2-user/2-file zero-error worst-case capacity is characterized by the following 9 inequalities:*

$$R^* \geq \frac{F_1 + F_2 - M_1}{2} \tag{Q1}$$

$$R^* \geq \frac{F_1 + F_2 - M_2}{2} \tag{Q2}$$

$$R^* \geq F_1 - M_1 \tag{Q3}$$

$$R^* \geq F_1 - M_2 \tag{Q4}$$

$$R^* \geq F_2 - M_1 \tag{Q5}$$

$$R^* \geq F_2 - M_2 \tag{Q6}$$

$$R^* \geq \frac{2F_1 + F_2 - M_1 - M_2}{2} \tag{Q7}$$

$$R^* \geq \frac{F_1 + 2F_2 - M_1 - M_2}{2} \tag{Q8}$$

$$R^* \geq F_1 + F_2 - M_1 - M_2. \tag{Q9}$$

If we further assume $F_1 \geq F_2$, then (Q5), (Q6), and (Q8) are always loose. The corresponding worst-case capacity R^* is described by Fig. B.1, which consists of 11 vertices and the 6 planes (Q1), (Q2), (Q3), (Q4), (Q7), and (Q9).

Recall that the worst-case rate objective (3.4) is convex and the PRCR is characterized by 28 linear constraints (O-1) to (IV-6). The problem of minimizing R^* subject to PRCR

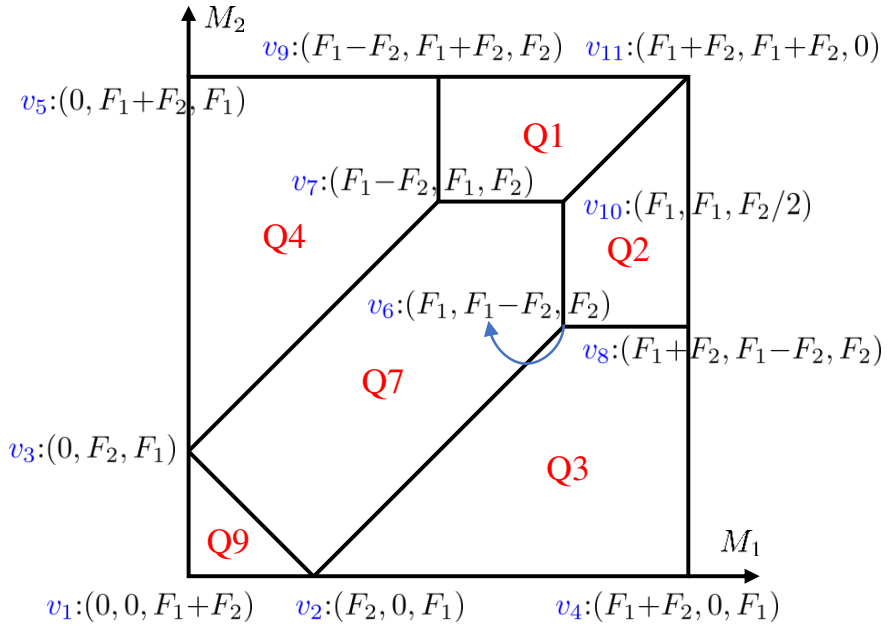


Figure B.1. The capacity of worst-case rate under the assumption $F_1 \geq F_2$. Each corner point is labeled by a tuple (M_1, M_2, R^*), where (M_1, M_2) describe the location and the third coordinate describe the corresponding exact worst-case rate capacity R^*.

is therefore a convex optimization problem. We solve the convex optimization problem by first converting to the equivalent linear programming problem as follows.

$$\begin{aligned} &\min_{\vec{R}} R^* \\ \text{subject to } & R^* \geq R_{(1,1)} && \text{(B.212)} \\ & R^* \geq R_{(1,2)} && \text{(B.213)} \\ & R^* \geq R_{(2,1)} && \text{(B.214)} \\ & R^* \geq R_{(2,2)} && \text{(B.215)} \\ & \text{(A1) to (B6).} \end{aligned}$$

By plugging in (A1) to (B6), the linear programming problem can be further converted to the following equivalent form

$$\begin{aligned} &\min_{\vec{R}} R^* \\ \text{subject to } & R^* \geq a_{\mathrm{i}}, \quad \mathrm{i} \in \{1,2,3,4\} && \text{(B.216)} \\ & 2R^* \geq b_{\mathrm{j}}, \quad \mathrm{j} \in \{1,2,3,4,5,6\}. && \text{(B.217)} \end{aligned}$$

Note that the values of a_1 to b_6 are calculated by max operations and we can expand each $\geq$ inequality with max operation on the right-hand side to multiple inequalities. If we expand the max operation in (B.216), we will have (Q3) to (Q6), and if we expand the max operation in (B.217), we will have (Q1), (Q2), (Q7) to (Q9). The converse proof is thus complete.

For the achievability part, without loss of generality, we only consider the corner points in the region $F_1 \geq F_2$ and $M_1 \geq M_2$, and there are 7 vertices v_1, v_2, v_4, v_6, v_8, v_{10}, and v_{11} as shown in Fig. B.1. The vertices v_1 and v_{11} can be achieved by some trivial schemes. The rest of them, v_2, v_4, v_6, v_8, and v_{10}, can be achieved by Vertices 1, 14, 1, 14, and 23 described in Tables B.1, B.14, B.1, B.14, and B.23, respectively.

Since the worst-case objective (3.4) is not a linear function of rates $R_{\vec{d}}$, $\vec{d} \in [N]^K$, the facets of the worst-case rate region may not match the facets of PRCR. For example, it can be observed that the edge (v_6, v_8) in Fig. B.1 is a new edge that did not appear in Fig. 3.2.

C. SUPPLEMENTARY MATERIALS FOR CHAPTER 4

C.1 Proof of Proposition 4.4.1

We first prove the selfish and unselfish designs achieves the same average rate $\tilde{R}$. We construct the lower bounds following the cut-set bounds in [31] for disjoint Θ_k, $k \in [K]$. That is, for all $k \in [K]$, $S_k \subseteq \Theta_k$, and $T = \prod_{k\in K} |S_k|$, we have

$$\sum_{\vec{d}:d_k\in S_k} R_{\vec{d}} + \sum_{k\in[K]} \frac{T}{|S_k|} M_k \geq \sum_{k\in[K]} \frac{T}{|S_k|} \left(\sum_{\mathrm{i}\in S_k} F_{\mathrm{i}} \right). \tag{C.1}$$

If we let $F_{\mathrm{i}} = F$ for all $\mathrm{i} \in [N]$ and substitute the uniform average rate (4.5) according to Definition 4.2.5 in (C.1), we obtain

$$\tilde{R} \geq \sum_{k\in[K]} \left(F - \frac{M_k}{|\Theta_k|} \right)^+. \tag{C.2}$$

The achievable scheme is as follows. In the placement phase, each user k caches $M_k/|\Theta_k|$ size of each file W_{i}, $\mathrm{i} \in \Theta_k$, and in delivery phase, for any demand $\vec{d}$, the server transmits the remain $(F-\frac{M_k}{|\Theta_k|})^+$ fraction of file for each demanded file W_{d_k} with rate $R_{\vec{d}} = \sum_{k\in[K]}(F-\frac{M_k}{|\Theta_k|})^+$ and hence the uniform-average rate is achieved with equality in (C.2).

For the worst-case rate R^*, it is clear that $R^* \geq \tilde{R}$ such that we also have

$$R^* \geq \sum_{k\in[K]} \left(F - \frac{M_k}{|\Theta_k|} \right)^+. \tag{C.3}$$

Since the described achievable scheme has same $R_{\vec{d}} = \sum_{k\in[K]}(F - \frac{M_k}{|\Theta_k|})^+$ for all $\vec{d}$, the worst-case rate is achieved with equality in (C.3).

C.2 Proof of Proposition 4.4.2

We first prove the lower bounds (P0) to (P5) and then show that the corner points in Fig. 4.2 are achievable. The bound (P0) is trivial and the bounds (P1), (P2) can be obtained by the cut-set bounds in [31].

Without loss of generality, we assume that $N_2 \geq N_1$. For simplicity, we relabel the $N = N_1 + N_2 - 1$ files as $\{-N_1+1, -N_1+2, \ldots, -1, 0, 1, \ldots, N_2-2, N_2-1\}$ such that $\Theta_1 = \{-N_1+1, \ldots, 0\}$, $\Theta_2 = \{0, \ldots, N_2-1\}$ with $\alpha = 1$ overlapped file W_0. In the following, we prove (P3) under the condition $N_2 \geq N_1 - 1$, (P4) under the condition $N_2 \geq N_1 + 1$, and (P5) under the condition $N_2 \geq N_1$. By symmetry, we then prove (P3) to (P5) for $N_2 \geq N_1$.

Lemma 10. *Consider $K = 2$ users and $N = N_1 + N_2 - 1$ files with FDS $\Theta_1 = \{-N_1 + 1, \ldots, 0\}$, $\Theta_2 = \{0, \ldots, N_2 - 1\}$ and overlapped file W_0, we have the following inequality*

$$\begin{aligned}
&(N_2-1)H(Z_1, W_0) + \sum_{\mathrm{i}=1}^{N_1-1} \sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} \\
\geq &(N_2-1)H(Z_1, [W_{-k}]_{k=0}^{N_1-1}) \\
&+ \sum_{\mathrm{i}=1}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1} | Z_1, [W_{-k}]_{k=0}^{N_1-1})
\end{aligned} \tag{C.4}$$

Proof. To prove inequality (C.4), we first derive

$$(N_2-1)H(Z_1, W_0) + \sum_{\mathrm{i}=1}^{N_1-1} \sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} \tag{C.5}$$

$$= \sum_{\mathrm{j}=1}^{N_2-1} \Big(H(Z_1, W_0) + R_{(-1,\mathrm{j})} \Big) + \sum_{\mathrm{i}=2}^{N_1-1} \sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} \tag{C.6}$$

$$\geq \sum_{\mathrm{j}=1}^{N_2-1} H(X_{(-1,\mathrm{j})}, Z_1, W_0, W_{-1}) + \sum_{\mathrm{i}=2}^{N_1-1} \sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} \tag{C.7}$$

$$\begin{aligned}
&\geq \sum_{\mathrm{i}=2}^{N_1-1} \sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} + (N_2-1)H(Z_1, W_0, W_{-1}) \\
&\quad + H([X_{(-1,\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1} | Z_1, W_0, W_{-1})
\end{aligned} \tag{C.8}$$

$$\begin{aligned}
&\geq (N_2-1)H(Z_1, [W_{-l}]_{l=0}^{N_1-1}) \\
&\quad + \sum_{\mathrm{i}=1}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1} | Z_1, [W_{-k}]_{k=0}^{\mathrm{i}})
\end{aligned} \tag{C.9}$$

where (C.7) follows from user 1 is able to decode W_{-1} from $X_{(-1,\mathrm{j})}$ and Z_1; (C.8) follows from iteratively applying Shannon-type inequality

$$\begin{aligned}&H(X_{(-1,k+1)}, Z_1, W_0, W_{-1}) + H([X_{(-1,\mathrm{j})}]_{\mathrm{j}=1}^{k}, Z_1, W_0, W_{-1})\\ &\geq H(Z_1, W_0, W_{-1}) + H([X_{(-1,\mathrm{j})}]_{\mathrm{j}=1}^{k+1}, Z_1, W_0, W_{-1})\end{aligned}$$

and (C.9) follows from iteratively applying (C.5) to (C.8) such that

$$\begin{aligned}&(N_2-1)H(Z_1, [W_k]_{k=0}^{l-1}) + \sum_{\mathrm{i}=l}^{N_1-1}\sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})}\\ \geq&(N_2-1)H(Z_1, [W_k]_{k=0}^{l}) + H([X_{(-1,\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1, [W_k]_{k=0}^{l})\\ &+\sum_{\mathrm{i}=l+1}^{N_1-1}\sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})}\end{aligned}$$

Finally, due to

$$\begin{aligned}&\sum_{\mathrm{i}=1}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1, [W_{-k}]_{k=0}^{\mathrm{i}})\\ \geq&\sum_{\mathrm{i}=1}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1, [W_{-k}]_{k=0}^{N_1-1})\end{aligned}$$

we have inequality (C.4) and hence complete the proof of Lemma 10. □

Lemma 11. *Consider $K = 2$ users and $N = N_1 + N_2 - 1$ files with FDS $\Theta_1 = \{-N_1 + 1, \ldots, 0\}$, $\Theta_2 = \{0, \ldots, N_2 - 1\}$, where $N_2 \geq N_1 - 1$, the uniform-average-rate $\tilde{R}$ satisfied (P3), or equivalently*

$$N_2M_1 + (N_1-1)M_2 + N_1N_2\tilde{R} \geq (2N_1N_2 - N_2)F \tag{C.10}$$

Proof. We derive inequality (C.10) from

$$
\begin{aligned}
&N_2M_1 + (N_1-1)M_2 + N_1N_2\tilde{R} \\
=&N_2M_1 + (N_1-1)M_2 + \sum_{\mathrm{i}=0}^{N_1-1}\sum_{\mathrm{j}=0}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} && \text{(C.11)}\\
\geq&N_2M_1 + \sum_{\mathrm{j}=0}^{N_2-1} R_{(0,\mathrm{j})} + (N_1-1)M_2 + \sum_{\mathrm{i}=1}^{N_1-1} R_{(-\mathrm{i},0)} \\
&+ \sum_{\mathrm{i}=1}^{N_1-1}\sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} && \text{(C.12)}\\
\geq&(N_1-2)H(Z_2,W_0) + (N_2-1)H(Z_1,[W_{-k}]_{k=0}^{N_1-1}) \\
&+ \sum_{\mathrm{i}=1}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1,[W_{-k}]_{k=0}^{N_1-1}) + (N_1+N_2)F && \text{(C.13)}\\
\geq&(N_2-N_1+1)H([W_{-\mathrm{i}}]_{\mathrm{i}=0}^{N_1-1}) + (N_1-2)H(W_0) \\
&+ (N_1-2)H([W_{\mathrm{i}}]_{\mathrm{i}=-N_1+1}^{N_2-1}) + (N_1+N_2)F && \text{(C.14)}\\
=&(2N_1N_2-N_2)F && \text{(C.15)}
\end{aligned}
$$

where equality (C.11) holds from the definition of uniform-average-rate in (4.5).

Inequality (C.13) can be shown by first deriving

$$
\begin{aligned}
&N_2M_1 + \sum_{\mathrm{j}=0}^{N_2-1} R_{(0,\mathrm{j})} + (N_1-1)M_2 + \sum_{\mathrm{i}=1}^{N_1-1} R_{(-\mathrm{i},0)} \\
\geq&\sum_{\mathrm{j}=0}^{N_2-1} H(X_{(0,\mathrm{j})},Z_1,W_0) + \sum_{\mathrm{i}=1}^{N_1-1} H(X_{(-\mathrm{i},0)},Z_2,W_0) && \text{(C.16)}\\
\geq&(N_2-1)H(Z_1,W_0) + H(Z_1,W_0,[X_{(0,\mathrm{j})}]_{\mathrm{j}=0}^{N_2-1}) \\
&+ (N_1-2)H(Z_2,W_0) + H(Z_2,W_0,[X_{(-\mathrm{i},0)}]_{\mathrm{i}=0}^{N_1-1}) && \text{(C.17)}\\
\geq&H(W_0) + H(Z_1,Z_2,W_0,[X_{(0,\mathrm{j})}]_{\mathrm{j}=0}^{N_2-1},[X_{(-\mathrm{i},0)}]_{\mathrm{i}=1}^{N_1-1}) \\
&+ (N_2-1)H(Z_1,W_0) + (N_1-2)H(Z_2,W_0) && \text{(C.18)}\\
=&(N_2-1)H(Z_1,W_0) + (N_1-2)H(Z_2,W_0) \\
&+ (N_1+N_2)F && \text{(C.19)}
\end{aligned}
$$

where (C.16) follows from user 1 can decode W_0 from $(X_{(0,\mathrm{j})}, Z_1)$ and user 2 can decode W_0 from $(X_{(-\mathrm{i},0)}, Z_2)$; (C.17) follows from iteratively applying Shannon-type inequality for the two summation terms; (C.18) follows from another Shannon-type inequality; and (C.19) follows from

$$\begin{aligned} H(Z_1, Z_2, W_0, [X_{(0,\mathrm{j})}]_{\mathrm{j}=0}^{N_2-1}, [X_{(-\mathrm{i},0)}]_{\mathrm{i}=1}^{N_1-1}) &= H([X_{(0,\mathrm{j})}]_{\mathrm{j}=0}^{N_2-1}, [X_{(-\mathrm{i},0)}]_{\mathrm{i}=1}^{N_1-1}) \\ &= (N_1 + N_2 - 1)F \end{aligned}$$

Combining the inequality (C.19), the remaining term $\sum_{\mathrm{i}=1}^{N_1-1}\sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})}$ in (C.12), and Lemma 10, we show (C.13).

Inequality (C.14) can be derived from

$$\begin{aligned} &(N_1-2)H(Z_2, W_0) + (N_2-1)H(Z_1, [W_{-k}]_{k=0}^{N_1-1}) + \\ &+ \sum_{\mathrm{i}=1}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1} | Z_1, [W_{-k}]_{k=0}^{N_1-1}) \qquad \text{(C.20)} \\ =& \sum_{\mathrm{i}=1}^{N_1-2} \Big(H(Z_2, W_0) + H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}, Z_1, [W_{-k}]_{k=0}^{N_1-1}) \Big) \\ &+ (N_2 - N_1 + 1) H(Z_1, [W_{-k}]_{k=0}^{N_1-1}) \qquad \text{(C.21)} \\ \geq& \sum_{\mathrm{i}=1}^{N_1-2} \Big(H(W_0) + H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}, Z_1, Z_2, [W_{-k}]_{k=0}^{N_1-1}) \Big) \\ &+ (N_2 - N_1 + 1) H(Z_1, [W_{-k}]_{k=0}^{N_1-1}) \qquad \text{(C.22)} \\ =& (N_1-2)H(W_0) + (N_1-2)H([W_{\mathrm{j}}]_{\mathrm{j}=1}^{N_2-1}, [W_k]_{k=0}^{N_1-1}) \\ &+ (N_2 - N_1 + 1) H([W_{-k}]_{k=0}^{N_1-1}) \qquad \text{(C.23)} \end{aligned}$$

where (C.21) follows from the property of conditional entropy and the condition $N_2 \geq N_1 - 1$; (C.22) follows from the Shannon-type inequality; and (C.23) follows from that user 2 can decode $[W_{\mathrm{j}}]_{\mathrm{j}=1}^{N_2-1}$ from $[X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}$ and Z_2.

Finally (C.15) follows from the equality $(N_2 - N_1 + 1)N_1 F + (N_1 - 2)(N_1 + N_2)F + (N_1 + N_2)F = (2N_1N_2 - N_2)F$. Thus we complete the proof of inequality (C.10). □

Lemma 12. *Consider $K = 2$ users and $N = N_1 + N_2 - 1$ files with FDS $\Theta_1 = \{-N_1 + 1, \ldots, 0\}$, $\Theta_2 = \{0, \ldots, N_2 - 1\}$, where $N_2 \geq N_1 + 1$, the uniform-average-rate $\tilde{R}$ satisfied (P4), or equivalently*

$$(N_2 - 1)M_1 + N_1 M_2 + N_1 N_2 \tilde{R} \geq (2N_1 N_2 - N_1)F \tag{C.24}$$

Proof. We derive inequality (C.24) from

$$\begin{aligned}
&(N_2 - 1)M_1 + N_1 M_2 + N_1 N_2 \tilde{R} \\
=&(N_2 - 1)M_1 + N_1 M_2 + \sum_{\mathrm{i}=0}^{N_1-1} \sum_{\mathrm{j}=0}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} && \text{(C.25)} \\
=&(N_2 - 1)M_1 + \sum_{\mathrm{j}=1}^{N_2-1} R_{(0,\mathrm{j})} + N_1 M_2 + \sum_{\mathrm{i}=0}^{N_1-1} R_{(-\mathrm{i},0)} \\
&+ \sum_{\mathrm{i}=1}^{N_1-1} \sum_{\mathrm{j}=1}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} && \text{(C.26)} \\
\geq&N_1 H(Z_2, W_0) + (N_2 - 1)H(Z_1, [W_{-k}]_{k=0}^{N_1-1}) \\
&+ \sum_{\mathrm{i}=0}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1} | Z_1, [W_{-k}]_{k=0}^{N_1-1}) && \text{(C.27)} \\
\geq&N_1 H(W_0) + N_1 H([W_{\mathrm{j}}]_{\mathrm{j}=1}^{N_2-1}, [W_{-k}]_{k=0}^{N_1-1}) \\
&+ (N_2 - N_1 - 1)H([W_{-k}]_{k=0}^{N_1-1}) && \text{(C.28)} \\
=&(2N_1 N_2 - N_1)F && \text{(C.29)}
\end{aligned}$$

where equality (C.25) follows from the definition of uniform-average-rate in (4.5).

Inequality (C.27) can be shown by first deriving

$$
\begin{aligned}
&(N_2-1)M_1+\sum_{\mathrm{j}=1}^{N_2-1}R_{(0,\mathrm{j})}+N_1M_2+\sum_{\mathrm{i}=0}^{N_1-1}R_{(-\mathrm{i},0)}\\
\geq&\sum_{\mathrm{j}=1}^{N_2-1}H(X_{(0,\mathrm{j})},Z_1,W_0)+\sum_{\mathrm{i}=0}^{N_1-1}H(X_{(-\mathrm{i},0)},Z_2,W_0) &\text{(C.30)}\\
\geq&(N_2-1)H(Z_1,W_0)+H([X_{(0,\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1,W_0)\\
&+N_1H(Z_2,W_0) &\text{(C.31)}\\
\geq&(N_2-1)H(Z_1,W_0)+H([X_{(0,\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1,[W_{-k}]_{k=0}^{N_1-1})\\
&+N_1H(Z_2,W_0) &\text{(C.32)}
\end{aligned}
$$

where (C.30) follows from user 1 can decode W_0 from $(X_{(0,\mathrm{j})},Z_1)$ and user 2 can decode W_0 from $(X_{(-\mathrm{i},0)},Z_2)$; (C.31) follows from iteratively applying Shannon-type inequality for the first summation terms and the inequality $H(X_{(-\mathrm{i},0)},Z_2,W_0)\geq H(Z_2,W_0)$; (C.18) follows from the inequality of conditional entropy. Combining the inequality (C.32), the remaining term $\sum_{\mathrm{i}=1}^{N_1-1}\sum_{\mathrm{j}=1}^{N_2-1}R_{(-\mathrm{i},\mathrm{j})}$ in (C.26), and Lemma 10, we show (C.27).

Inequality (C.28) can be derived from

$$
\begin{aligned}
&(N_2-1)H(Z_1,[W_{-k}]_{k=0}^{N_1-1})+N_1H(Z_2,W_0)\\
&+\sum_{\mathrm{i}=0}^{N_1-1}H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1,[W_{-k}]_{k=0}^{N_1-1}) &\text{(C.33)}\\
=&\sum_{\mathrm{i}=0}^{N_1-1}\Big(H(Z_2,W_0)+H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1},Z_1,[W_{-k}]_{k=0}^{N_1-1})\Big)\\
&+(N_2-N_1-1)H(Z_1,[W_{-k}]_{k=0}^{N_1-1}) &\text{(C.34)}\\
=&\sum_{\mathrm{i}=0}^{N_1-1}\Big(H(W_0)+H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1},Z_1,Z_2,[W_{-k}]_{k=0}^{N_1-1})\Big)\\
&+(N_2-N_1-1)H(Z_1,[W_{-k}]_{k=0}^{N_1-1}) &\text{(C.35)}\\
=&N_1H(W_0)+N_1H([W_{\mathrm{j}}]_{\mathrm{j}=1}^{N_2-1},[W_{-k}]_{k=0}^{N_1-1})\\
&+(N_2-N_1-1)H([W_{-k}]_{k=0}^{N_1-1}) &\text{(C.36)}
\end{aligned}
$$

where (C.34) follows from the property of conditional entropy and the condition $N_2 \geq N_1+1$; (C.35) follows from the Shannon-type inequality; and (C.36) follows from that user 2 can decode $[W_{\mathrm{j}}]_{\mathrm{j}=1}^{N_2-1}$ from $[X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}$ and Z_2.

Finally (C.29) follows from the equality $N_1F + N_1(N_1+N_2-1)F + (N_2-N_1-1)N_1F = (2N_1N_2-N_1)F$. Thus we complete the proof of inequality (C.24). □

Lemma 13. *Consider $K=2$ users and $N = N_1+N_2-1$ files with FDS $\Theta_1 = \{-N_1+1,\dots,0\}$, $\Theta_2 = \{0,\dots,N_2-1\}$, where $N_2 \geq N_1$, the uniform-average-rate $\tilde{R}$ satisfied (P5), or equivalently*

$$N_2M_1 + N_1M_2 + N_1N_2\tilde{R} \geq (2N_1N_2-1)F \tag{C.37}$$

Proof. We derive inequality (C.37) from

$$\begin{aligned}
&N_2M_1 + N_1M_2 + N_1N_2\tilde{R} \\
=&N_2M_1 + (N_1-1)M_2 + \sum_{\mathrm{i}=0}^{N_1-1}\sum_{\mathrm{j}=0}^{N_2-1} R_{(-\mathrm{i},\mathrm{j})} + H(Z_2) && \text{(C.38)}\\
\geq&(N_1-2)H(Z_2,W_0) + (N_2-1)H(Z_1,[W_{-k}]_{k=0}^{N_1-1}) \\
&+ \sum_{\mathrm{i}=1}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1,[W_{-k}]_{k=0}^{N_1-1}) \\
&+ (N_1+N_2)F + H(Z_2) && \text{(C.39)}\\
\geq&(N_1-2)H(W_0) + (N_1-1)H([W_{\mathrm{j}}]_{\mathrm{j}=1}^{N_2-1},[W_{-k}]_{k=0}^{N_1-1}) \\
&+ (N_2-N_1)H([W_{-k}]_{k=0}^{N_1-1}) + (N_1+N_2)F && \text{(C.40)}\\
=&2N_1N_2-1 && \text{(C.41)}
\end{aligned}$$

where equality (C.38) follows from the definition of uniform-average-rate in (4.5); (C.39) follows from (C.13) in Lemma 11.

Inequality (C.40) can be derived from

$$
\begin{aligned}
&(N_1-2)H(Z_2,W_0)+(N_2-1)H(Z_1,[W_{-k}]_{k=0}^{N_1-1}) \\
&+\sum_{\mathrm{i}=1}^{N_1-1} H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}|Z_1,[W_{-k}]_{k=0}^{N_1-1})+H(Z_2) && \text{(C.42)}\\
\geq &\sum_{\mathrm{i}=1}^{N_1-2}\Big(H(Z_2,W_0)+H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1},Z_1,[W_{-k}]_{k=0}^{N_1-1})\Big) \\
&+H([X_{(-N_1+1,\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1},Z_1,[W_{-k}]_{k=0}^{N_1-1})+H(Z_2) \\
&+(N_2-N_1)H(Z_1,[W_{-k}]_{k=0}^{N_1-1}) && \text{(C.43)}\\
\geq &\sum_{\mathrm{i}=1}^{N_1-2}\Big(H(W_0)+H([X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1},Z_1,Z_2,[W_{-k}]_{k=0}^{N_1-1})\Big) \\
&+H([X_{(-N_1+1,\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1},Z_1,Z_2,[W_{-k}]_{k=0}^{N_1-1}) \\
&+(N_2-N_1)H(Z_1,[W_{-k}]_{k=0}^{N_1-1}) && \text{(C.44)}\\
=&(N_1-2)H(W_0)+(N_1-1)H([W_{\mathrm{j}}]_{\mathrm{j}=1}^{N_2-1},[W_{-k}]_{k=0}^{N_1-1}) \\
&+(N_2-N_1)H([W_{-k}]_{k=0}^{N_1-1}) && \text{(C.45)}
\end{aligned}
$$

where (C.43) follows from the property of conditional entropy and the condition $N_2 \geq N_1$; (C.44) follows from the Shannon-type inequality and the property of joint entropy; and (C.45) follows from that user 2 can decode $[W_{\mathrm{j}}]_{\mathrm{j}=1}^{N_2-1}$ from $[X_{(-\mathrm{i},\mathrm{j})}]_{\mathrm{j}=1}^{N_2-1}$ and Z_2.

Finally (C.41) follows from the equality $(N_1-2)F+(N_1-1)(N_1+N_2-1)F+(N_2-N_1)N_1F+(N_1+N_2)F=(2N_1N_2-1)F$. Thus we complete the proof of inequality (C.37). □

Now we show that the rate region characterized by (P0) to (P5) is achievable. Since the rate can be achieved by space-sharing, we only describe the achievable schemes for the corner points in Fig 4.2. The achievable rate $\tilde{R}(M_1,M_2)$ at corner point (M_1,M_2) can be denoted by the triple $v_{\mathrm{i}}:(M_1,M_2,\tilde{R})$. where the rate $\tilde{R}$ can be computed by (P0) to (P5).

The achievable schemes for vertices $v_1:(0,0,2-\frac{1}{N_1N_2})$, $v_7:(N,0,1)$, $v_8:(0,N,1)$, $v_9:(N_1,N_2,0)$, $v_{10}:(N,N_2,0)$, $v_{11}:(N_1,N,0)$, and $v_{12}:(N,N,0)$ are trivial, and we list on the achievable schemes for vertices v_2 to v_6 as follows.

Vertex 2: $(N_1-1,0,1+\frac{N_2-1}{N_1N_2})$. User 1 caches $Z_1=(W_{-N+1},\ldots,W_{-1})$ and user 2 has no cache. For the demand $d_1=d_2=0$, the server transmits W_0 with rate $R_{(0,0)}=1$; for

the demand $d_1 = 0$, $d_2 \neq 0$, the server transmits (W_0, W_{d_2}) with rate $R_{(0,d_2)} = 2$; and for the demand $d_1 \neq 0$, the server transmits W_{d_2} with rate $R_{(d_1,d_2)} = 1$. The average rate is thus $\tilde{R} = \frac{1+2(N_2-1)+(N_1-1)N_2}{N_1N_2} = 1 + \frac{N_2-1}{N_1N_2}$.

Vertex 3: $(0, N_2 - 1, 1 + \frac{N_1-1}{N_1N_2})$. The achievable scheme is symmetric with respect to the scheme of Vertex 2 by exchanging the role of user 1 and 2.

Vertex 4: $(N_1, 0, 1)$. User 1 caches $Z_1 = (W_{-N+1}, \ldots, W_{-1}, W_0)$ and user 2 has no cache. For any demand (d_1, d_2), the server only transmits W_2 such that $R_{(d_1,d_2)} = 1$ and hence the average rate is $\tilde{R} = 1$.

Vertex 5: $(0, N_2, 1)$. The achievable scheme is symmetric with respect to the scheme of Vertex 2 by exchanging the role of user 1 and 2.

Vertex 6: $(N_1 - 1, N_2 - 1, \frac{N}{N_1N_2})$. User 1 caches $Z_1 = (W_{-N+1}, \ldots, W_{-1})$ and user 2 caches $Z_2 = (W_1, \ldots, W_{N_2-1})$. For the demand $d_1 = d_2 = 0$, the server transmits W_0 with rate $R_{(0,0)} = 1$; for the demand $d_1 = 0$, $d_2 \neq 0$, the server transmits W_0 with rate $R_{(0,d_2)} = 1$; for the demand $d_2 = 0$, $d_1 \neq 0$, the server transmits W_0 with rate $R_{(d_1,0)} = 1$; and for the demand $d_1 \neq 0$, $d_2 \neq 0$, the server transmit nothing with rate $R_{(d_1,d_2)} = 0$. The average rate is $\tilde{R} = \frac{N_1+N_2-1}{N_1N_2} = \frac{N}{N_1N_2}$.

C.3 Proof of Proposition 4.4.3

The bounds (Q1) and (Q2) are the cut set bounds following the close argument in [31]. To derive bound (Q3), we first obtain following inequality for $\{3\} \subseteq \{\mathrm{i},\mathrm{j}\} \subseteq \{1,2,3\}$ and $k,l \in \{1,2\}$.

$$
\begin{aligned}
&M_1 + M_2 + R_{(1,\mathrm{i})} + R_{(2,\mathrm{j})} + R_{(k,1)} + R_{(l,2)} && \text{(C.46)}\\
&\geq H(X_{(1,\mathrm{i})}, X_{(2,\mathrm{j})}, Z_1) + H(X_{(k,1)}, X_{(l,2)}, Z_2) && \text{(C.47)}\\
&= H(X_{(1,\mathrm{i})}, X_{(2,\mathrm{j})}, Z_1, W_1, W_2)\\
&\quad + H(X_{(k,1)}, X_{(l,2)}, Z_2, W_1, W_2) && \text{(C.48)}\\
&\geq H(X_{(1,\mathrm{i})}, X_{(2,\mathrm{j})}, X_{(k,1)}, X_{(l,2)}, Z_2, W_1, W_2)\\
&\quad + H(W_1, W_2) && \text{(C.49)}\\
&= H(X_{(1,\mathrm{i})}, X_{(2,\mathrm{j})}, X_{(k,1)}, X_{(l,2)}, Z_2, W_1, W_2, W_3)\\
&\quad + H(W_1, W_2) && \text{(C.50)}\\
&\geq H(W_1, W_2, W_3) + H(W_1, W_2) = 5F && \text{(C.51)}
\end{aligned}
$$

where (C.48) follows from that user 1 can decode W_1 and W_2 based on $X_{(1,\mathrm{i})}$, $X_{(2,\mathrm{j})}$, and Z_1; and user 2 can decode W_1 and W_2 based on $X_{(k,1)}$, $X_{(l,2)}$, and Z_2. (C.49) follows from using the matroidal Shannon inequality. (C.50) follows from the assumption $\mathrm{i} = 3$ or $\mathrm{j} = 3$ such that user 2 can decode W_3 based on $X_{(1,\mathrm{i})}$, $X_{(2,\mathrm{j})}$, and Z_2.

We then obtain three inequalities for $(\mathrm{i},\mathrm{j},k,l) = (3,1,1,2)$, $(2,3,1,2)$, and $(3,3,2,1)$, respectively as follows.

$$
\begin{aligned}
&M_1 + M_2 + R_{(1,3)} + R_{(2,1)} + R_{(1,1)} + R_{(2,2)} \geq 5F && \text{(C.52)}\\
&M_1 + M_2 + R_{(1,2)} + R_{(2,3)} + R_{(1,1)} + R_{(2,2)} \geq 5F && \text{(C.53)}\\
&M_1 + M_2 + R_{(1,3)} + R_{(2,3)} + R_{(2,1)} + R_{(1,2)} \geq 5F. && \text{(C.54)}
\end{aligned}
$$

The summation of (C.52), (C.53), and (C.54) yields $3M_1 + 3M_2 + 12\tilde{R} \geq 15F$ or equivalently (Q3).

To derive the bound (Q4), we first obtain the following inequality for $(\mathrm{i},\mathrm{j}) = (1,2)$ or $(2,1)$.

$$M_1 + M_2 + R_{(\mathrm{j},3)} + R_{(\mathrm{i},\mathrm{j})} \tag{C.55}$$
$$\geq H(X_{(\mathrm{j},3)}, Z_1) + H(X_{(\mathrm{i},\mathrm{j})}, Z_2) \tag{C.56}$$
$$= H(X_{(\mathrm{j},3)}, Z_1, W_{\mathrm{j}}) + H(X_{(\mathrm{i},\mathrm{j})}, Z_2, W_{\mathrm{j}}) \tag{C.57}$$
$$\geq H(X_{(\mathrm{j},3)}, X_{(\mathrm{i},\mathrm{j})}, Z_1, Z_2, W_{\mathrm{j}}) + H(W_{\mathrm{j}}) \tag{C.58}$$
$$= H(X_{(\mathrm{j},3)}, X_{(\mathrm{i},\mathrm{j})}, Z_1, Z_2, W_{\mathrm{i}}, W_{\mathrm{j}}, W_3) + H(W_{\mathrm{j}}) \tag{C.59}$$
$$\geq H(W_1, W_2, W_3) + H(W_{\mathrm{j}}) = 4F. \tag{C.60}$$

where (C.57) follows from that user 1 can decode W_{j} based on $X_{(\mathrm{j},3)}$ and Z_1; and user 2 can decode W_{j} based on $X_{(\mathrm{i},\mathrm{j})}$ and Z_2. (C.58) follows from using the matroidal Shannon inequality. (C.59) follows from the assumption $(\mathrm{i},\mathrm{j}) = (1,2)$ or $(2,1)$ such that user 1 can decode W_{i} based on $X_{(\mathrm{i},\mathrm{j})}$ and Z_1; and user 2 can decode W_3 based on $X_{(\mathrm{j},3)}$ and Z_2.

We then obtain two inequalities for $(\mathrm{i},\mathrm{j}) = (1,2)$ and $(2,1)$, respectively as follows.

$$M_1 + M_2 + R_{(2,3)} + R_{(1,2)} \geq 4F \tag{C.61}$$
$$M_1 + M_2 + R_{(1,3)} + R_{(2,1)} \geq 4F. \tag{C.62}$$

The summation of (C.52) and (C.61) yields $2M_1 + 2M_2 + 6\tilde{R} \geq 9F$ or equivalently (Q4).

To derive the bounds (Q5) and (Q6), we first obtain the following cut-set bounds

$$M_1 + R_{(1,1)} + R_{(2,2)} \geq 2F \tag{C.63}$$
$$M_2 + R_{(1,1)} + R_{(2,2)} \geq 2F. \tag{C.64}$$

The summation of (C.61), (C.62), and (C.63) yields (Q5) and the summation of (C.61), (C.62), and (C.64) yields (Q6).

Now we show that corner points v_1 to v_{11} of bounds (Q1) to (Q6) are zero-error achievable. The achievable scheme for point v_1 is trivial. The points v_2 and v_3 can be zero-error achieved by user 1 caches $Z_1 = (W_1, W_2)$ of size $2F$ in the placement phase and in the delivery phase,

for any demand $(d_1, d_2) = (\mathrm{i}, \mathrm{j})$, the server transmits $X_{(\mathrm{i},\mathrm{j})} = W_{\mathrm{j}}$ corresponding to rate $R_{(\mathrm{i},\mathrm{j})} = F$. The points v_4 and v_5 can be zero-error achieved by user 1 caches $Z_1 = (W_1, W_2)$ of size $2F$ and user 2 caches $Z_2 = (W_1, W_2, W_3)$ of size $3F$ in the placement phase and for any demand patterns, no need for extra transmission in the delivery phase. The points v_6 and v_7 can be zero-error achieved by user 2 caches $Z_2 = (W_1 \oplus W_2, W_1 \oplus W_3)$ of size $2F$ in the placement phase and in the delivery phase, for the demand $(d_1, d_2) = (\mathrm{i}, \mathrm{j})$, the server transmits $X_{(\mathrm{i},\mathrm{j})} = W_{\mathrm{i}}$ corresponding to rate $R_{(\mathrm{i},\mathrm{j})} = F$.

For describing the achievable schemes of corner points v_8 to v_{11}, we divide each files into two disjoint subfiles of equal size $\frac{F}{2}$, i.e., $W_1 = (A_1, A_2)$, $W_2 = (B_1, B_2)$, and $W_3 = (C_1, C_2)$. The zero-error achievable scheme of v_8 is that in the placement phase, user 1 caches $Z_1 = (A_1, B_1)$ of size F, and user 2 caches $Z_2 = (A_2, B_2, W_3)$ of size $2F$ and in the delivery phase, for the 6 possible demands the server transmits $X_{(1,1)} = A_1 \oplus A_2$, $X_{(1,2)} = A_2 \oplus B_1$, $X_{(1,3)} = A_2$, $X_{(2,1)} = A_1 \oplus B_2$, $X_{(2,2)} = B_1 \oplus B_2$, and $X_{(2,3)} = B_2$ corresponding to the same rate $\frac{F}{2}$ and hence achieves the average rate $\tilde{R} = \frac{F}{2}$. The zero-error achievable scheme of v_9 is that in the placement phase, user 1 caches $Z_1 = (A_1, B_1)$ of size F, and user 2 caches $Z_2 = (A_2, B_2)$ of size F and in the delivery phase, for the 6 possible demands the server transmits $X_{(1,1)} = A_1 \oplus A_2$, $X_{(1,2)} = A_2 \oplus B_1$, $X_{(1,3)} = (A_2, W_3)$, $X_{(2,1)} = A_1 \oplus B_2$, $X_{(2,2)} = B_1 \oplus B_2$, and $X_{(2,3)} = (B_2, W_3)$ corresponding to the rates $R_{(1,1)} = R_{(1,2)} = R_{(2,1)} = R_{(2,2)} = \frac{F}{2}$, and $R_{(1,3)} = R_{(2,3)} = \frac{3F}{2}$. The average rate is therefore $\tilde{R} = \frac{5}{6}F$.

The zero-error achievable scheme of v_{10} is that in the placement phase, user 1 caches $Z_1 = (A_1, B_1, C_2 \oplus A_2 \oplus B_2)$ of size $\frac{3F}{2}$, and user 2 caches $Z_2 = (A_2, B_2)$ of size F and in the delivery phase, for the 6 possible demands the server transmits $X_{(1,1)} = A_1 \oplus A_2$, $X_{(1,2)} = A_2 \oplus B_1$, $X_{(1,3)} = (C_1, C_2 \oplus B_2)$, $X_{(2,1)} = A_1 \oplus B_2$, $X_{(2,2)} = B_1 \oplus B_2$, and $X_{(2,3)} = (C_1, C_2 \oplus A_2)$ corresponding to the rates $R_{(1,1)} = R_{(1,2)} = R_{(2,1)} = R_{(2,2)} = \frac{F}{2}$, and $R_{(1,3)} = R_{(2,3)} = F$. The average rate is therefore $\tilde{R} = \frac{2}{3}F$. The zero-error achievable scheme of v_{11} is that in the placement phase, user 1 caches $Z_1 = (A_1, B_1, C_1)$ of size $\frac{3}{2}F$, and user 2 caches $Z_2 = (A_2, B_2, C_2)$ of size $\frac{3}{2}F$ and in the delivery phase, for the 6 possible demands the server transmits $X_{(1,1)} = A_1 \oplus A_2$, $X_{(1,2)} = A_2 \oplus B_1$, $X_{(1,3)} = A_2 \oplus C_1$, $X_{(2,1)} = A_1 \oplus B_2$, $X_{(2,2)} = B_1 \oplus B_2$, and $X_{(2,3)} = B_2 \oplus C_1$ corresponding to the same rate $\frac{F}{2}$ and hence achieves the average rate $\tilde{R} = \frac{F}{2}$.

C.4 Proof of Proposition 4.4.4

The bound (Q7), we first obtain the following inequality for $\mathrm{i} \in \{1, 2\}$

$$
\begin{aligned}
& M_1 + M_2 + R_{(\mathrm{i},1)} + R_{(\mathrm{i},2)} + R_{(\mathrm{i},3)} && \text{(C.65)} \\
& \geq H(X_{(\mathrm{i},3)}, Z_1) + H(X_{(\mathrm{i},1)}, X_{(\mathrm{i},2)}, Z_2) && \text{(C.66)} \\
& = H(X_{(\mathrm{i},3)}, Z_1, W_{\mathrm{i}}) + H(X_{(\mathrm{i},1)}, X_{(\mathrm{i},2)}, Z_2, W_1, W_2) && \text{(C.67)} \\
& = H(X_{(\mathrm{i},3)}, Z_1, W_{\mathrm{i}}) \\
& \quad + H(X_{(\mathrm{i},1)}, X_{(\mathrm{i},2)}, Z_1, Z_2, W_1, W_2) && \text{(C.68)} \\
& \geq H(X_{(\mathrm{i},3)}, X_{(\mathrm{i},1)}, X_{(\mathrm{i},2)}, Z_1, Z_2, W_1, W_2) \\
& \quad + H(Z_1, W_{\mathrm{i}}) && \text{(C.69)} \\
& = H(X_{(\mathrm{i},3)}, X_{(\mathrm{i},1)}, X_{(\mathrm{i},2)}, Z_1, Z_2, W_1, W_2, W_3) \\
& \quad + H(Z_1, W_{\mathrm{i}}) && \text{(C.70)} \\
& \geq H(W_1, W_2, W_3) + H(Z_1, W_{\mathrm{i}}) = 3F + H(Z_1, W_{\mathrm{i}}) && \text{(C.71)}
\end{aligned}
$$

where (C.67) follows from that user 1 can decode W_{i} based on $X_{(\mathrm{i},3)}$ and Z_1; and user 2 can decode W_1 and W_2 based on $X_{(\mathrm{i},1)}$, $X_{(\mathrm{i},2)}$, and Z_2. (C.68) follows from the definition of selfish coded caching where $Z_1 = \phi_1(W_1, W_2)$. (C.69) follows from the assumption $\mathrm{i} \in \{1, 2\}$ and the matroidal Shannon inequality. (C.70) follows from user 2 can decode W_3 based on $X_{(\mathrm{i},3)}$ and Z_2.

We then obtain two inequalities for $\mathrm{i} = 1$ and $\mathrm{i} = 2$ as follows

$$
\begin{aligned}
M_1 + M_2 + \sum_{\mathrm{j}=1}^{3} R_{(1,\mathrm{j})} &\geq 3F + H(Z_1, W_1) && \text{(C.72)} \\
M_1 + M_2 + \sum_{\mathrm{j}=1}^{3} R_{(2,\mathrm{j})} &\geq 3F + H(Z_1, W_2). && \text{(C.73)}
\end{aligned}
$$

The summation of (C.72) and (C.73) leads to

$$2M_1 + 2M_2 + 6\tilde{R} \tag{C.74}$$
$$\geq 6F + H(Z_1, W_1) + H(Z_1, W_2) \tag{C.75}$$
$$\geq 6F + H(Z_1, W_1, W_2) + H(Z_1) \tag{C.76}$$
$$\geq 6F + H(W_1, W_2) + H(Z_1) = 8F + M_1 \tag{C.77}$$

and hence the bound (Q7), where (C.76) follows from the matroidal Shannon inequality.

It is clear that the aforementioned achievable schemes of corner points v_1 to v_7 are actually selfish coded caching schemes. Therefore joint (Q1), (Q2), and (Q5) to (Q7) is the selfish capacity of uniform average rate.

C.5 Proof of Proposition 4.4.5

In an uncoded prefetching scheme, let subfile $w_{k,\mathrm{i}}$ of size $m_k^{(\mathrm{i})}$ be the fraction of file W_i that stored in user k's cache memory such that $\sum_{\mathrm{i}=1}^{N} m_k^{(\mathrm{i})} \leq M_k$ for $k = 1, 2$, we have the following lemma.

Lemma 14. *Consider an uncoded prefeching scheme for $K = 2$ users with file demand sets Θ_1 and Θ_2, the minimum transmission rates for all $\mathrm{i} \in \Theta_1$ and $\mathrm{j} \in \Theta_2$ are*

$$\begin{cases} R_{(\mathrm{i},\mathrm{j})} = F - \min(m_1^{(\mathrm{i})}, m_2^{(\mathrm{j})}) & \mathrm{i} = \mathrm{j} \\ R_{(\mathrm{i},\mathrm{j})} = 2F - \min(m_1^{(\mathrm{i})}, m_2^{(\mathrm{j})}) & \\ \qquad\qquad - \min(F, m_1^{(\mathrm{i})} + m_2^{(\mathrm{i})}, m_1^{(\mathrm{j})} + m_2^{(\mathrm{j})}) & \mathrm{i} \neq \mathrm{j} \end{cases}$$

Proof. For any $\mathrm{i} \in \Theta_1$ and $\mathrm{j} \in \Theta_2$, we have the cut-set bound

$$R_{(\mathrm{i},\mathrm{j})} + m_1^{(\mathrm{i})} \geq F \tag{C.78}$$
$$R_{(\mathrm{i},\mathrm{j})} + m_2^{(\mathrm{j})} \geq F \tag{C.79}$$

and for any $\mathrm{i} \in \Theta_1$, $\mathrm{j} \in \Theta_2$, $\mathrm{i} \neq \mathrm{j}$, we show the additional bounds

$$R_{(\mathrm{i},\mathrm{j})} + m_1^{(\mathrm{i})} + m_1^{(\mathrm{j})} + m_2^{(\mathrm{j})} \geq 2F \tag{C.80}$$

$$R_{(\mathrm{i},\mathrm{j})} + m_1^{(\mathrm{i})} + m_2^{(\mathrm{i})} + m_2^{(\mathrm{j})} \geq 2F. \tag{C.81}$$

Inequality (C.80) can be proved by

$$R_{(\mathrm{i},\mathrm{j})} + m_1^{(\mathrm{i})} + m_1^{(\mathrm{j})} + m_2^{(\mathrm{j})} \tag{C.82}$$

$$= H(X_{(\mathrm{i},\mathrm{j})}, w_{1,\mathrm{i}}, w_{1,\mathrm{j}}, w_{2,\mathrm{j}}) \tag{C.83}$$

$$= H(X_{(\mathrm{i},\mathrm{j})}, w_{1,\mathrm{i}}, w_{1,\mathrm{j}}, w_{2,\mathrm{j}}, W_{\mathrm{i}}) \tag{C.84}$$

$$= H(X_{(\mathrm{i},\mathrm{j})}, w_{1,\mathrm{i}}, w_{1,\mathrm{j}}, w_{2,\mathrm{i}}, w_{2,\mathrm{j}}, W_{\mathrm{i}}) \tag{C.85}$$

$$= H(X_{(\mathrm{i},\mathrm{j})}, w_{1,\mathrm{i}}, w_{1,\mathrm{j}}, w_{2,\mathrm{i}}, w_{2,\mathrm{j}}, W_{\mathrm{i}}, W_{\mathrm{j}}) \geq 2F \tag{C.86}$$

where (C.84) follows from that user 1 can decode W_{i} from $(X_{(\mathrm{i},\mathrm{j})}, w_{1,\mathrm{i}}, w_{1,\mathrm{j}})$; (C.85) follows from that $w_{2,\mathrm{i}}$ is a fraction of W_{i}; and (C.86) follows from that user 2 can decode W_{j} from $(X_{(\mathrm{i},\mathrm{j})}, w_{2,\mathrm{i}}, w_{2,\mathrm{j}})$. Similarly, we can derive (C.81). Combining (C.78) to (C.81), we have for $\mathrm{i} = \mathrm{j}$

$$R_{(\mathrm{i},\mathrm{j})} \geq F - \min(m_1^{(\mathrm{i})}, m_2^{(\mathrm{j})}) \tag{C.87}$$

where the equality of (C.87) can be achieved by transmitting the (coded) of remain fractions of $(W_{\mathrm{i}}, W_{\mathrm{i}})$; and for $\mathrm{i} \neq \mathrm{j}$

$$\begin{aligned} R_{(\mathrm{i},\mathrm{j})} \geq & 2F - \min(m_1^{(\mathrm{i})}, m_2^{(\mathrm{j})}) \\ & - \min(F, m_1^{(\mathrm{i})} + m_2^{(\mathrm{i})}, m_1^{(\mathrm{j})} + m_2^{(\mathrm{j})}) \end{aligned} \tag{C.88}$$

where the equality (C.88) can be achieved by transmitting the (coded) of remain fractions of $(W_{\mathrm{i}}, W_{\mathrm{j}})$.

Thus we complete the proof of Lemma 14. □

Lemma 15. *Without loss of generality, the optimal unselfish and uncoded prefetching schemes for $K = 2$ users with file demand sets Θ_1 and Θ_2 satisfy $m_1^{(\mathrm{i})} \geq m_2^{(\mathrm{i})}$ for all $\mathrm{i} \in \Theta_1 \backslash \Theta_2$ and $m_2^{(\mathrm{j})} \geq m_1^{(\mathrm{j})}$ for all $\mathrm{j} \in \Theta_2 \backslash \Theta_1$.*

We first proof for all $\mathrm{i} \in \{1,2\}$, $\mathrm{j} \in \{1,2,3\}$, $\mathrm{i} \neq \mathrm{j}$

Therefore we have

$$R_{(1,1)} + m_1^{(1)} \geq F \tag{C.89}$$

$$R_{(1,2)} + m_1^{(1)} + m_1^{(2)} + m_2^{(2)} \geq 2F \tag{C.90}$$

$$R_{(1,3)} + m_1^{(1)} + m_1^{(3)} + m_2^{(3)} \geq 2F \tag{C.91}$$

$$R_{(2,1)} + m_1^{(1)} + m_1^{(2)} + m_2^{(1)} \geq 2F \tag{C.92}$$

$$R_{(2,2)} + m_1^{(2)} \geq F \tag{C.93}$$

$$R_{(2,3)} + m_1^{(2)} + m_1^{(3)} \geq F \tag{C.94}$$

where $m_1^{(1)} + m_1^{(2)} + m_1^{(3)} \leq M_1$ and $m_2^{(1)} + m_2^{(2)} + m_2^{(3)} \leq M_2$. Finally we prove (Q8) as follows.

$$4M_1 + M_2 + 6\tilde{R}_{\mathrm{gu}} \tag{C.95}$$

$$\geq 4(m_1^{(1)} + m_1^{(2)} + m_1^{(3)}) + (m_2^{(1)} + m_2^{(2)} + m_2^{(3)})$$

$$+ R_{(1,1)} + R_{(1,2)} + R_{(1,3)} + R_{(2,1)} + R_{(2,2)} + R_{(2,3)} \tag{C.96}$$

$$\geq 9F \tag{C.97}$$

We then prove (Q9). Consider the following inequalities:

$$R_{(1,1)} + m_2^{(1)} \geq F \tag{C.98}$$

$$R_{(2,2)} + m_2^{(2)} \geq F \tag{C.99}$$

$$R_{(1,3)} + m_1^{(1)} + m_1^{(3)} + m_2^{(3)} \geq 2F \tag{C.100}$$

$$R_{(2,3)} + m_1^{(2)} + m_1^{(3)} + m_2^{(3)} \geq 2F \tag{C.101}$$

$$R_{(1,2)} + m_1^{(1)} + m_2^{(1)} + m_2^{(2)} \geq 2F \tag{C.102}$$

$$R_{(2,1)} + m_1^{(2)} + m_2^{(2)} + m_2^{(1)} \geq 2F \tag{C.103}$$

$$R_{(1,2)} + R_{(2,1)} + m_2^{(1)} + m_2^{(2)} \geq 2F \tag{C.104}$$

such that 2[(C.98) + (C.99) + (C.100) + (C.101)] + (C.102) + (C.103) + (C.104) yields

$$\begin{aligned} &2 \sum_{\mathrm{i}\in\Theta_1,\mathrm{j}\in\Theta_2} R_{(\mathrm{i},\mathrm{j})} + 3\left(m_1^{(1)} + m_1^{(2)} + m_1^{(3)}\right) \\ &+ 5\left(m_2^{(1)} + m_2^{(2)} + m_2^{(3)}\right) + m_1^{(3)} - m_2^{(3)} \geq 18F \end{aligned} \tag{C.105}$$

Therefore we prove (Q9) from

$$\begin{aligned} &3M_1 + 5M_2 + 12\tilde{R}_{\mathrm{gu}} \\ \geq &2 \sum_{\mathrm{i}\in\Theta_1,\mathrm{j}\in\Theta_2} R_{(\mathrm{i},\mathrm{j})} + 3\left(m_1^{(1)} + m_1^{(2)} + m_1^{(3)}\right) \\ &+ 5\left(m_2^{(1)} + m_2^{(2)} + m_2^{(3)}\right) + m_1^{(3)} - m_2^{(3)} \\ \geq &18F \end{aligned} \tag{C.106}$$

where $m_1^{(3)} \leq m_2^{(3)}$.

C.6 Proof of Proposition 4.4.7 and Corollary 6

We first prove the following lemmas.

Lemma 16. *Without loss of generality, we can assume the optimal selfish and uncoded prefetching scheme has the following form: There exist non-negative variables* $\{M_1^u, M_1^c, M_2^u, M_2^c\}$

such that for $\mathrm{i} \in \{1,2\}$, $M_{\mathrm{i}}^u + M_{\mathrm{i}}^c = M_{\mathrm{i}}$ *and user* i *stores* $M_{\mathrm{i}}^u/(N_{\mathrm{i}} - \alpha)$ *size for each file in* $\mathcal{U}_{\mathrm{i}}$ *and stores* M_{i}^c/α *size for each file in* $\mathcal{C}$.

Proof. Consider the coded caching problem of N files $\{W_1, \cdots, W_N\}$ of same size F, and $K = 2$ users with cache memory sizes (M_1, M_2) and FDS (Θ_1, Θ_2). We show by contradiction that there always exists an optimal selfish and uncoded prefeching scheme where all the stored subfiles in $\mathcal{U}_1$ (or $\mathcal{C}$) has the same size.

Suppose not, that is, for any optimal selfish and uncoded prefeching scheme that user 1 caches subfile w_k from file W_k of size a_k, $k \in \mathcal{U}_1$ (or $k \in \mathcal{U}_1$), there exist $\mathrm{i} \neq \mathrm{j} \in \mathcal{U}_1$ (or $\mathrm{i} \neq \mathrm{j} \in \mathcal{C}$) such that $a_{\mathrm{i}} \neq a_{\mathrm{j}}$. Then we exchange the index of i and j to form a new coded caching problem of N files of same size and 2 users. Since $\mathrm{i} \neq \mathrm{j} \in \mathcal{U}_1$ (or $\mathrm{i} \neq \mathrm{j} \in \mathcal{C}$), the users has the same FDS (Θ_1, Θ_2) after exchanging the indexes. Therefore the new problem is identical to the original coded caching problem and we have another optimal selfish and uncoded prefeching scheme that user 1 caches subfile w_k from file W_k of size b_k, where $b_{\mathrm{i}} = a_{\mathrm{j}}$, $b_{\mathrm{j}} = a_{\mathrm{i}}$, and $b_k = a_k$ for $k \neq \mathrm{i}, \mathrm{j}$. By using space sharing, we create an optimal selfish and uncoded prefeching scheme that user 1 caches subfile w_k from file W_k of size c_k, where $c_{\mathrm{i}} = c_{\mathrm{j}} = (a_{\mathrm{i}} + a_{\mathrm{j}})/2$ are the same, and $c_k = a_k$ for $k \neq \mathrm{i}, \mathrm{j}$. By iteratively making the size of subfiles even, we finally have an optimal selfish and uncoded scheme where all the stored subfiles in $\mathcal{U}_1$ (or $\mathcal{C}$) has the same size.

Given the placement scheme of user 1, we can apply similar procedure for user 2 such that there always exists an optimal selfish and uncoded prefeching scheme: For $\mathrm{i} \in \{1,2\}$, user i stores size of m_{i}^u for each file in $\mathcal{U}_{\mathrm{i}}$ and stores size of m_{i}^c for each file in $\mathcal{C}$. Thus we have completed the proof of Lemma 16. □

Lemma 17. *Without loss of generality, consider an optimal selfish and uncoded prefeching scheme satisfying Lemma 16 that user 1 stores subfile* w_{i} *of size* m_1^u *for each file* $\mathrm{i} \in \mathcal{U}_1$ *and user 2 stores subfile* w_{j} *of size* m_2^u *for each file* $\mathrm{j} \in \mathcal{U}_2$*. The minimum transmission rate for any demand* $\vec{d} = (d_1, d_2)$, $d_1 \in \mathcal{U}_1$ *and* $d_2 \in \mathcal{U}_2$, *is* $R_{\vec{d}} = 2F - m_1^u - m_2^u$.

Proof. Since the the scheme is selfish and uncoded prefetching, we have the cut-set bound for $d_1 \in \mathcal{U}_1$ and $d_2 \in \mathcal{U}_2$

$$R_{\vec{d}} + m_1^u + m_2^u \geq H(X_{\vec{d}}, w_{d_1}, w_{d_2}) \geq H(W_{d_1}, W_{d_2}) = 2F$$

or equivalently $R_{\vec{d}} \geq 2F - m_1^u - m_2^u$. The equality can be achieved by simply transmitting the renaming part of W_{d_1} to user 1 and renaming part of W_{d_2} to user 2, i.e., $R_{\vec{d}} = F - m_1^u + F - m_2^u$. Thus we have proved Lemma 17. □

Lemma 18. *Without loss of generality, consider an optimal selfish and uncoded prefeching scheme satisfying Lemma 16 that user 1 stores subfile w_{i} of size m_1^u for each file $\mathrm{i} \in \mathcal{U}_1$ and user 2 stores subfile w_{j} of size m_2^c for each file $\mathrm{j} \in \mathcal{C}$. The minimum transmission rate for any demand $\vec{d} = (d_1, d_2)$, $d_1 \in \mathcal{U}_1$ and $d_2 \in \mathcal{C}$, is $R_{\vec{d}} = 2F - m_1^u - m_2^c$.*

Proof. Since the the scheme is selfish and uncoded prefetching, we have the cut-set bound for $d_1 \in \mathcal{U}_1$ and $d_2 \in \mathcal{C}$

$$R_{\vec{d}} + m_1^u + m_2^c \geq H(X_{\vec{d}}, w_{d_1}, w_{d_2}) \geq H(W_{d_1}, W_{d_2}) = 2F$$

or equivalently $R_{\vec{d}} \geq 2F - m_1^u - m_2^c$. The equality can be achieved by simply transmitting the renaming part of W_{d_1} to user 1 and renaming part of W_{d_2} to user 2, i.e., $R_{\vec{d}} = F - m_1^u + F - m_2^c$. Thus we have proved Lemma 18. □

Lemma 19. *Without loss of generality, consider an optimal selfish and uncoded prefeching scheme satisfying Lemma 16 that user 1 and 2 store subfile u_{i} of size m_1^c and v_{i} of size m_2^c for each file $\mathrm{i} \in \mathcal{C}$, respectively. The minimum transmission rate for any demand $\vec{d} = (d_1, d_2)$, $d_1, d_2 \in \mathcal{C}$ is*

$$\begin{cases} R_{\vec{d}} = F - \min(m_1^c, m_2^c) & d_1 = d_2 \\ R_{\vec{d}} = 2F - \min(m_1^c, m_2^c) - \min(F, m_1^c + m_2^c) & d_1 \neq d_2 \end{cases}$$

Proof. Since the the scheme is selfish and uncoded prefetching, we have following cut-set bounds for $d_1, d_2 \in \mathcal{C}$

$$R_{\vec{d}} + m_{\mathrm{i}}^c \geq H(X_{\vec{d}}, u_{d_{\mathrm{i}}}) \geq H(W_{d_{\mathrm{i}}}) = F, \quad \mathrm{i} \in \{1, 2\} \tag{C.107}$$

On the other hand, we have additional cut-set bounds for $d_1 \neq d_2 \in \mathcal{C}$

$$\begin{aligned} R_{\vec{d}} + 2m_1^c + m_2^c &\geq H(X_{\vec{d}}, u_{d_1}, u_{d_2}, v_{d_2}) \\ &= H(X_{\vec{d}}, u_{d_1}, u_{d_2}, v_{d_2}, W_{d_1}) && \text{(C.108)} \\ &= H(X_{\vec{d}}, u_{d_1}, u_{d_2}, v_{d_1}, v_{d_2}, W_{d_1}) && \text{(C.109)} \\ &= H(X_{\vec{d}}, u_{d_1}, u_{d_2}, v_{d_1}, v_{d_2}, W_{d_1}, W_{d_2}) \geq 2F && \text{(C.110)} \end{aligned}$$

where (C.108) follows from that user 1 can decode W_{d_1} from $X_{\vec{d}}$ and (u_{d_1}, u_{d_2}); (C.109) follows from that the uncoded v_{d_1} is a subfile of W_{d_1}; (C.109) follows from that user 2 can decode W_{d_2} from $X_{\vec{d}}$ and (v_{d_2}, v_{d_2}). By symmetry, we have the following equivalent cut-set bounds for $d_1 \neq d_2 \in \mathcal{C}$

$$R_{\vec{d}} + m_1^c + m_2^c + m_{\mathrm{i}}^c \geq 2F, \quad \mathrm{i} \in \{1, 2\}. \tag{C.111}$$

Consider user 1 and 2 store subfile u_{i} of size m_1^c and v_{i} of size m_2^c for each file $\mathrm{i} \in \mathcal{C}$, where v_{i} and u_{i} overlap as little as possible. When $d_1 = d_2 \in \mathcal{C}$, the minimum rate is derived by only (C.107) as $R_{\vec{d}} \geq F - \min(m_1^c, m_2^c)$. The equality can be achieved by dividing file W_{d_1} into subfiles (u_{d_1}, s_{d_1}) of sizes $(m_1^c, F - m_1^c)$ and file W_{d_2} into subfiles (t_{d_2}, v_{d_2}) of sizes $(m_2^c, F - m_2^c)$. The transmitted signal is $X_{\vec{d}} = s_{d_1} \oplus t_{d_2}$ of size $R_{\vec{d}} = \max(F - m_1^c, F - m_2^c)$. When $d_1 \neq d_2 \in \mathcal{C}$, the minimum rate is derived by jointly (C.107) and (C.111) as $R_{\vec{d}} \geq \max(F - m_1^c, F - m_2^c, 2F - 2m_1^c - m_1^c, 2F - m_1^c - 2m_1^c) = 2F - \min(m_1^c, m_2^c) - \min(F, m_1^c + m_2^c)$. If $F < m_1^c + m_2^c$, the achievable scheme is same as the scheme for $d_1 = d_2 \in \mathcal{C}$. If $F \geq m_1^c + m_2^c$, the equality can be achieved by dividing file W_{d_1} into subfiles $(u_{d_1}, q_{d_1}, s_{d_1})$ of sizes $(m_1^c, F - m_1^c - m_2^c, m_2^c)$ and file W_{d_2} into subfiles $(t_{d_2}, r_{d_2}, v_{d_2})$ of sizes $(m_1^c, F - m_1^c - m_2^c, m_2^c)$. The transmitted signal is $X_{\vec{d}} = (q_{d_1}, r_{d_2}, s_{d_1} \oplus t_{d_2})$ of size $R_{\vec{d}} = \max(F - 2m_1^c - m_2^c, F - m_1^c - 2m_2^c)$.

Thus we have completed the proof of Lemma 19. □

From Lemma 17, 18, and 19, we have the minimum average rate with selfish and uncoded prefetching as follows

$$\begin{aligned}\tilde{R}_{\text{su}} = &\frac{(N_1-\alpha)(N_2-\alpha)}{N_1N_2}(2F - m_1^u - m_2^u) \\ &+ \frac{\alpha(N_1-\alpha)}{N_1N_2}(2F - m_1^u - m_2^c) \\ &+ \frac{\alpha(N_2-\alpha)}{N_1N_2}(2F - m_1^c - m_2^u) \\ &+ \frac{\alpha(\alpha-1)}{N_1N_2}(2F - \min(m_1^c, m_2^c) - \min(F, m_1^c + m_2^c)) \\ &+ \frac{\alpha}{N_1N_2}(F - \min(m_1^c, m_2^c))\end{aligned} \tag{C.112}$$

with the constraints

$$\alpha m_{\text{i}}^c + (N_{\text{i}} - \alpha)m_{\text{i}}^u = M_{\text{i}}, \quad \text{i} \in \{1,2\} \tag{C.113}$$

$$0 \le m_{\text{i}}^c \le F, \quad \text{i} \in \{1,2\} \tag{C.114}$$

$$0 \le m_{\text{i}}^u \le F, \quad \text{i} \in \{1,2\}. \tag{C.115}$$

Equivalently, we first transform the rate (C.112) to four inequalities as follows

$$\begin{aligned}\tilde{R}_{\text{su}} \geq & \left(2-\frac{\alpha^2}{N_1N_2}\right)F-\left(1-\frac{\alpha}{N_2}\right)m_1^u-\left(1-\frac{\alpha}{N_1}\right)m_2^u \\ & -\frac{\alpha}{N_2}m_1^c-\frac{\alpha(N_2-\alpha)}{N_1N_2}m_2^c \end{aligned} \tag{C.116}$$

$$\begin{aligned}\tilde{R}_{\text{su}} \geq & \left(2-\frac{\alpha}{N_1N_2}\right)F-\left(1-\frac{\alpha}{N_2}\right)m_1^u-\left(1-\frac{\alpha}{N_1}\right)m_2^u \\ & -\frac{\alpha(N_1+\alpha-1)}{N_1N_2}m_1^c-\frac{\alpha(N_2-1)}{N_1N_2}m_2^c \end{aligned} \tag{C.117}$$

$$\begin{aligned}\tilde{R}_{\text{su}} \geq & \left(2-\frac{\alpha^2}{N_1N_2}\right)F-\left(1-\frac{\alpha}{N_2}\right)m_1^u-\left(1-\frac{\alpha}{N_1}\right)m_2^u \\ & -\frac{\alpha(N_1-\alpha)}{N_1N_2}m_1^c-\frac{\alpha}{N_1}m_2^c \end{aligned} \tag{C.118}$$

$$\begin{aligned}\tilde{R}_{\text{su}} \geq & \left(2-\frac{\alpha}{N_1N_2}\right)F-\left(1-\frac{\alpha}{N_2}\right)m_1^u-\left(1-\frac{\alpha}{N_1}\right)m_2^u \\ & -\frac{\alpha(N_1-1)}{N_1N_2}m_1^c-\frac{\alpha(N_2+\alpha-1)}{N_1N_2}m_2^c \end{aligned} \tag{C.119}$$

and then we substitute the equalities (C.113) as $m_{\text{i}}^c=(M_{\text{i}}-(N_{\text{i}}-\alpha)m_{\text{i}}^u)/\alpha$, $\text{i}\in\{1,2\}$ into (C.116) to (C.119) and eliminate m_1^c and m_2^c as follows.

$$\begin{aligned}\tilde{R}_{\text{su}} \geq & \left(2-\frac{\alpha^2}{N_1N_2}\right)F-\frac{1}{N_2}M_1-\frac{(N_2-\alpha)}{N_1N_2}M_2 \\ & -\frac{N_2-N_1}{N_2}m_1^u-\left(\frac{N_1-N_2}{N_1}-\frac{\alpha(N_2-\alpha)}{N_1N_2}\right)m_2^u \end{aligned} \tag{C.120}$$

$$\begin{aligned}\tilde{R}_{\text{su}} \geq & \left(2-\frac{\alpha}{N_1N_2}\right)F-\frac{N_1+\alpha-1}{N_1N_2}M_1-\frac{N_2-1}{N_1N_2}M_2 \\ & -\left(\frac{N_2-N_1}{N_2}-\frac{(N_1-\alpha)(\alpha-1)}{N_1N_2}\right)m_1^u+\frac{N_2-\alpha}{N_1N_2}m_2^u \end{aligned} \tag{C.121}$$

$$\begin{aligned}\tilde{R}_{\text{su}} \geq & \left(2-\frac{\alpha^2}{N_1N_2}\right)F-\frac{(N_1-\alpha)}{N_1N_2}M_1-\frac{1}{N_1}M_2 \\ & -\left(\frac{N_2-N_1}{N_2}+\frac{\alpha(N_1-\alpha)}{N_1N_2}\right)m_1^u-\frac{N_1-N_2}{N_1}m_2^u \end{aligned} \tag{C.122}$$

$$\begin{aligned}\tilde{R}_{\text{su}} \geq & \left(2-\frac{\alpha}{N_1N_2}\right)F-\frac{N_1-1}{N_1N_2}M_1-\frac{N_2+\alpha-1}{N_1N_2}M_2 \\ & -\frac{N_1-\alpha}{N_1N_2}m_1^u-\left(\frac{N_1-N_2}{N_1}-\frac{(N_2-\alpha)(\alpha-1)}{N_1N_2}\right)m_2^u \end{aligned} \tag{C.123}$$

Finally we apply Fourier-Motzkin Elimination to further eliminate m_1^u and m_2^u with constraints (C.114) and (C.115). The remain inequalities are (P8) to (P13).

To prove the achievability of (P8) to (P13), we show the corresponding achievable schemes of the 10 vertices u_1, u_2, u_4, u_6, u_8, u_{10}, u_{11}, u_{12}, u_{14}, u_{15}, and the rest vertices can be achieved by swapping the role of two users 1,2 and parameter N_1, N_2. More specifically, from Lemma 16 to 19, the achievable schemes can be determined with the following $(M_1^c, M_1^u, M_2^c, M_2^u)$ values: $u_1 : (0,0,0,0)$, $u_2 : (\frac{\alpha}{2}F, 0, \frac{\alpha}{2}F, 0)$, $u_4 : (0,0,0,(N_2-\alpha)F)$, $u_6 : (\frac{\alpha}{2}F, 0, \frac{\alpha}{2}F, (N_2-\alpha)F)$, $u_8 : (0,0,\alpha F, (N_2-\alpha)F)$, $u_{10} : (0,0,\alpha F,(N_2-\alpha)F)$, $u_{11} : (\frac{\alpha}{2}F, (N_1-\alpha)F, \frac{\alpha}{2}F, (N_2-\alpha)F)$, $u_{12} : (\alpha F, (N_1-\alpha)F, \alpha F, (N_2-\alpha)F)$, $u_{14} : (\alpha F, (N_1-\alpha)F, \alpha F, (N_2-\alpha)F)$.

To prove Corollary 6, without loss of generality, we first show that the optimal placement schemes (or optimal $(M_1^c, M_1^u, M_2^c, M_2^u)$ values) described in Algorithm 2 can be summarized in Table C.1 with the corresponding conditions.

For the region $M_1 \le \frac{\alpha}{2}F$ and $M_1 \le M_2 \le M_1 + (N_2-\alpha)F$, we substitute $(M_1, 0, M_1, M_2 - M_1)$ into (C.112) and obtain (P5) with equality. For the region $M_1 \le \frac{\alpha}{2}F$, $M_1 + (N_2-\alpha)F < M_2$, and $M_1 + M_2 \le N_2 F$, we substitute $(M_1, 0, M_2 - (N_2-\alpha)F, (N_2-\alpha)F)$ into (C.112) and obtain (P13) with equality. For the region $M_1 \le \frac{\alpha}{2}F$, $M_1 + (N_2-\alpha)F < M_2$, and $M_1 + M_2 > N_2 F$, we substitute $(M_1, 0, M_2 - (N_2-\alpha)F, (N_2-\alpha)F)$ into (C.112) and obtain (P8) with equality. For the region $\frac{\alpha}{2}F < M_1 \le N_1 - \frac{\alpha}{2}F$ and $\frac{\alpha}{2}F < M_2 \le N_2 - \frac{\alpha}{2}F$, we substitute $(\frac{\alpha}{2}F, M_1 - \frac{\alpha}{2}F, \frac{\alpha}{2}F, M_2 - \frac{\alpha}{2}F)$ into (C.112) and obtain (P10) with equality. For the region $\frac{\alpha}{2}F < M_1 \le N_1 - \frac{\alpha}{2}F$ and $N_2 - \frac{\alpha}{2}F < M_2$, we substitute $(\frac{\alpha}{2}F, M_1 - \frac{\alpha}{2}F, M_2 - (N_2 - \alpha)F, (N_2-\alpha)F)$ into (C.112) and obtain (P8) with equality. For the region $N_1 - \frac{\alpha}{2}F < M_1$, $N_2 - \frac{\alpha}{2}F < M_2$, and $M_1 + (N_2 - N_1)F \le M_2$, we substitute $(M_1 - (N_1-\alpha)F, (N_1-\alpha)F, M_2 - (N_2-\alpha)F, (N_2-\alpha)F)$ into (C.112) and obtain (P11) with equality.

Since the placement scheme in Table C.1 can achieve the capacity (P2) to (P5), we thus prove Corollary 6.

Table C.1. The optimal $(M_1^c, M_1^u, M_2^c, M_2^u)$ values of the selfish and uncoded prefeching schemes for $\alpha \geq 2$.

Optimal $(M_1^c, M_1^u, M_2^c, M_2^u)$ values	Conditions
$(M_1, 0, M_1, M_2 - M_1)$	$M_1 \leq \frac{\alpha}{2}F,\ M_1 \leq M_2 \leq M_1 + (N_2 - \alpha)F$
$(M_1, 0, M_2 - (N_2 - \alpha)F, (N_2 - \alpha)F)$	$M_1 \leq \frac{\alpha}{2}F,\ M_1 + (N_2 - \alpha)F < M_2$
$(\frac{\alpha}{2}F, M_1 - \frac{\alpha}{2}F, \frac{\alpha}{2}F, M_2 - \frac{\alpha}{2}F)$	$\frac{\alpha}{2}F < M_1 \leq N_1 - \frac{\alpha}{2}F,\ \frac{\alpha}{2}F < M_2 \leq N_2 - \frac{\alpha}{2}F$
$(\frac{\alpha}{2}F, M_1 - \frac{\alpha}{2}F, M_2 - (N_2 - \alpha)F, (N_2 - \alpha)F)$	$\frac{\alpha}{2}F < M_1 \leq N_1 - \frac{\alpha}{2}F,\ N_2 - \frac{\alpha}{2}F < M_2$
$(M_1 - (N_1 - \alpha)F, (N_1 - \alpha)F, M_2 - (N_2 - \alpha)F, (N_2 - \alpha)F)$	$N_1 - \frac{\alpha}{2}F < M_1,\ N_2 - \frac{\alpha}{2}F < M_2,\ M_1 + (N_2 - N_1)F \leq M_2$

VITA

Chih-Hua Chang received the B.S. degree in Electrical Engineering and the M.S. degree in Communication Engineering from National Taiwan University, Taipei, Taiwan, in 2010 and 2012, respectively. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. From 2014 to 2015, he was a Research Assistant with the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taipei, Taiwan. His current research interests include Internet of Things, network coding, and coded caching.